

# SD-WAN

The following topics provide information about SD-WAN:

- [SD-WAN overview on page 782](#)
- [SD-WAN quick start on page 786](#)
- [SD-WAN members and zones on page 796](#)
- [Performance SLA on page 808](#)
- [SD-WAN rules on page 851](#)
- [Advanced routing on page 946](#)
- [VPN overlay on page 977](#)
- [Advanced configuration on page 1076](#)
- [SD-WAN cloud on-ramp on page 1122](#)
- [SD-WAN Network Monitor service on page 1145](#)
- [Troubleshooting SD-WAN on page 1174](#)

## SD-WAN overview

SD-WAN is a software-defined approach to managing Wide-Area Networks (WAN). It consolidates the physical transport connections, or underlays, and monitors and load-balances traffic across the links. VPN overlay networks can be built on top of the underlays to control traffic across different sites.

Health checks and SD-WAN rules define the expected performance and business priorities, allowing the FortiGate to automatically and intelligently route traffic based on the application, internet service, or health of a particular connection.

WAN security and intelligence can be extended into the LAN by incorporating wired and wireless networks under the same domain. FortiSwitch and FortiAP devices integrate seamlessly with the FortiGate to form the foundation of an SD-Branch.

Some of the key benefits of SD-WAN include:






- Reduced cost with transport independence across MPLS, 4G/5G LTE, and others.
- Reduced complexity with a single vendor and single-pane-of-glass management.
- Improve business application performance thanks to increased availability and agility.
- Optimized user experience and efficiency with SaaS and public cloud applications.

## SD-WAN components and design principles

SD-WAN can be broken down into three layers:

- Management and orchestration
- Control, data plane, and security
- Network access

The control, data plane, and security layer can only be deployed on a FortiGate. The other two layers can help to scale and enhance the solution. For large deployments, FortiManager and FortiAnalyzer provide the management and orchestration capabilities FortiSwitch and FortiAP provide the components to deploy an SD-Branch.

| Layer                             | Functions  | Devices  |  |
|-----------------------------------|--|--|--|
| Management and orchestration      | <ul style="list-style-type: none"> <li>• Unified management</li> <li>• Template based solution</li> <li>• Zero touch provisioning</li> <li>• Logging, monitoring, and analysis</li> <li>• Automated orchestration using the REST API</li> </ul>  | FortiManager<br>  | FortiAnalyzer<br> |
| Control, data plane, and security | <ul style="list-style-type: none"> <li>• Consolidation of underlays and overlays into SD-WAN zones <ul style="list-style-type: none"> <li>• <a href="#">Underlay</a> and <a href="#">Overlay</a></li> </ul> </li> <li>• Scalable VPN solutions using ADVPN <ul style="list-style-type: none"> <li>• <a href="#">Overlay</a></li> </ul> </li> <li>• Static and dynamic routing definition <ul style="list-style-type: none"> <li>• <a href="#">Routing</a></li> </ul> </li> <li>• NGFW firewalling <ul style="list-style-type: none"> <li>• <a href="#">Security</a></li> </ul> </li> <li>• SD-WAN health-checks and monitoring <ul style="list-style-type: none"> <li>• <a href="#">SD-WAN</a></li> </ul> </li> <li>• Application-aware steering and intelligence <ul style="list-style-type: none"> <li>• <a href="#">SD-WAN</a></li> </ul> </li> </ul> | FortiGate<br>     |  |
| Network access                    | <ul style="list-style-type: none"> <li>• Wired and wireless network segmentation</li> <li>• Built-in network access control</li> </ul>   | FortiSwitch<br> | FortiAP<br>     |

## Design principles

The [Five-pillar approach](#), described in the SD-WAN / SD-Branch Architecture for MSSPs guide, is recommended when designing a secure SD-WAN solution.

### Underlay

Determine the WAN links that will be used for the underlay network, such as your broadband link, MPLS, 4G/5G LTE connection, and others.

For each link, determine the bandwidth, quality and reliability (packet loss, latency, and jitter), and cost. Use this information to determine which link to prefer, what type of traffic to send across the each link, and to help you the baselines for health-checks.

## Overlay

VPN overlays are needed when traffic must travel across multiple sites. These are usually site-to-site IPsec tunnels that interconnect branches, datacenters, and the cloud, forming a hub-and-spoke topology.

The management and maintenance of the tunnels should be considered when determining the overlay network requirements. Manual tunnel configuration might be sufficient in a small environment, but could become unmanageable as the environment size increases. ADVPN can be used to help scale the solution; see [ADVPN on page 2217](#) for more information.

## Routing

Traditional routing designs manipulate routes to steer traffic to different links. SD-WAN uses traditional routing to build the basic routing table to reach different destinations, but uses SD-WAN rules to steer traffic. This allows the steering to be based on criteria such as destination, internet service, application, route tag, and the health of the link. Routing in an SD-WAN solution is used to identify all possible routes across the underlays and overlays, which the FortiGate balances using ECMP.

In the most basic configuration, static gateways that are configured on an SD-WAN member interface automatically provide the basic routing needed for the FortiGate to balance traffic across the links. As the number of sites and destinations increases, manually maintaining routes to each destination becomes difficult. Using dynamic routing to advertise routes across overlay tunnels should be considered when you have many sites to interconnect.

## Security

Security involves defining policies for access control and applying the appropriate protection using the FortiGate's NGFW features. Efficiently grouping SD-WAN members into SD-WAN zones must also be considered. Typically, underlays provide direct internet access and overlays provide remote internet or network access. Grouping the underlays together into one zone, and the overlays into one or more zones could be an effective method.

## SD-WAN

The SD-WAN pillar is the intelligence that is applied to traffic steering decisions. It is comprised of four primary elements:

- **SD-WAN zones**

SD-WAN is divided into zones. SD-WAN member interfaces are assigned to zones, and zones are used in policies as source and destination interfaces. You can define multiple zones to group SD-WAN interfaces together, allowing logical groupings for overlay and underlay interfaces. Routing can be configured per zone.

See [SD-WAN members and zones on page 796](#).

- **SD-WAN members**

Also called interfaces, SD-WAN members are the ports and interfaces that are used to run traffic. At least one interface must be configured for SD-WAN to function.

See [Configuring the SD-WAN interface on page 786](#).

- **Performance SLAs**

Also called health-checks, performance SLAs are used to monitor member interface link quality, and to detect link failures. When the SLA falls below a configured threshold, the route can be removed, and traffic can be steered to different links in the SD-WAN rule.

SLA health-checks use active or passive probing:

- Active probing requires manually defining the server to be probed, and generates consistent probing traffic.
- Passive probing uses active sessions that are passing through firewall policies used by the related SD-WAN interfaces to derive health measurements. It reduces the amount of configuration, and eliminates probing traffic. See [Passive WAN health measurement on page 821](#) for details.

See [Performance SLA on page 808](#).

- **SD-WAN rules**

Also called services, SD-WAN rules control path selection. Specific traffic can be dynamically sent to the best link, or use a specific route.

Rules control the strategy that the FortiGate uses when selecting the outbound traffic interface, the SLAs that are monitored when selecting the outgoing interface, and the criteria for selecting the traffic that adheres to the rule. When no SD-WAN rules match the traffic, the implicit rule applies.

See [SD-WAN rules on page 851](#).

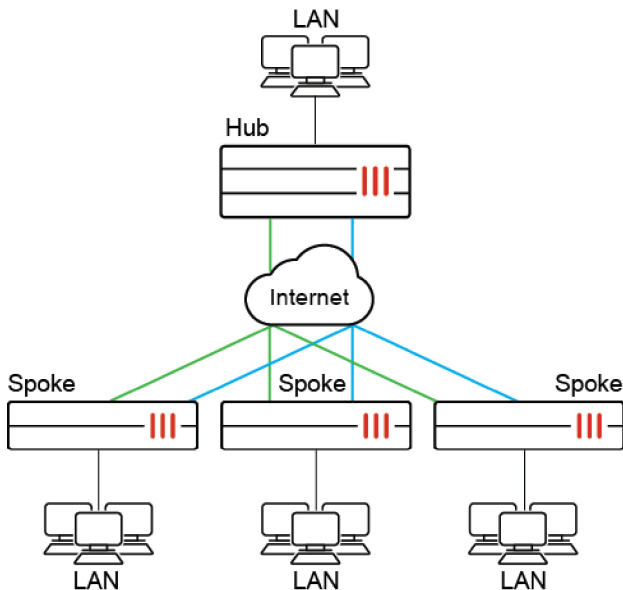
## SD-WAN designs and architectures

The core functionalities of Fortinet's SD-WAN solution are built into the FortiGate. Whether the environment contains one FortiGate, or one hundred, you can use SD-WAN by enabling it on the individual FortiGates.

At a basic level, SD-WAN can be deployed on a single device in a single site environment:

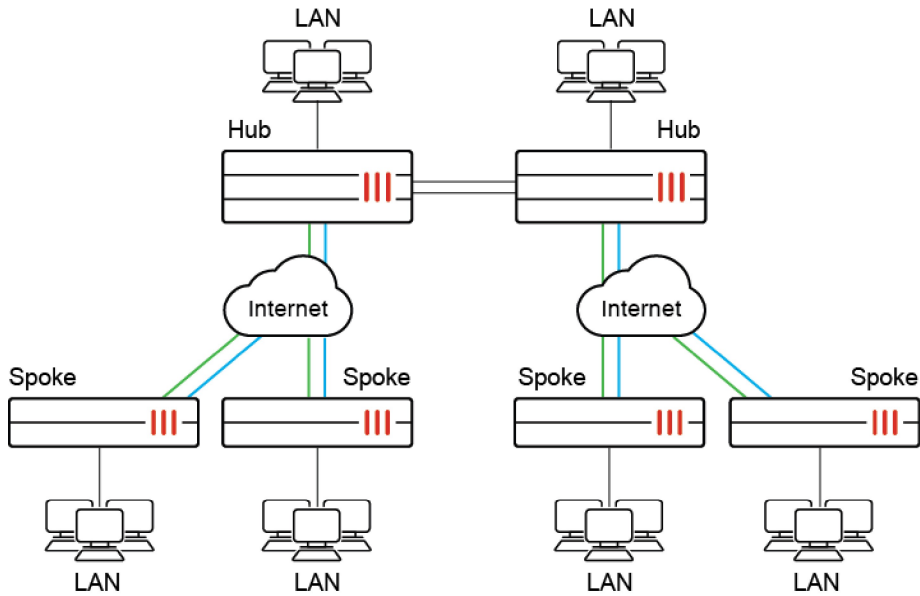


At a more advanced level, SD-WAN can be deployed in a multi-site, hub and spoke environment:



At an enterprise or MSSP level, the network can include multiple hubs, possibly across multiple regions:



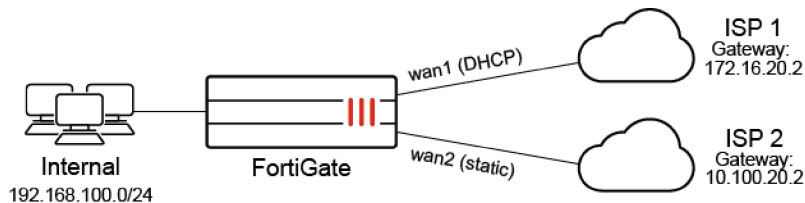


For more details, see the [SD-WAN / SD-Branch Architecture for MSSPs guide](#).

## SD-WAN quick start

This section provides an example of how to start using SD-WAN for load balancing and redundancy.

In this example, two ISP internet connections, wan1 (DHCP) and wan2 (static), use SD-WAN to balance traffic between them at 50% each.



1. [Configuring the SD-WAN interface on page 786](#)
2. [Adding a static route on page 788](#)
3. [Selecting the implicit SD-WAN algorithm on page 789](#)
4. [Configuring firewall policies for SD-WAN on page 789](#)
5. [Link monitoring and failover on page 790](#)
6. [Results on page 791](#)
7. [Configuring SD-WAN in the CLI on page 794](#)

## Configuring the SD-WAN interface

First, SD-WAN must be enabled and member interfaces must be selected and added to a zone. The selected FortiGate interfaces can be of any type (physical, aggregate, VLAN, IPsec, and others), but must be removed from any other configurations on the FortiGate.

In this step, two interfaces are configured and added to the default SD-WAN zone (virtual-wan-link) as SD-WAN member interfaces. This example uses a mix of static and dynamic IP addresses; your deployment could also use only one or the other.

Once the SD-WAN members are created and added to a zone, the zone can be used in firewall policies, and the whole SD-WAN can be used in static routes.

### To configure SD-WAN members:

1. Configure the wan1 and wan2 interfaces. See [Interface settings on page 164](#) for details.
  - a. Set the wan1 interface *Addressing mode* to *DHCP* and *Distance* to *10*.



By default, *Retrieve default gateway from server* (`defaultgw` in the CLI) is enabled for DHCP interfaces. This enables using the default gateway information that is retrieved from the DHCP server to create a default route through the DHCP interface with the default administrative distance.

The default administrative distance for DHCP interfaces is 5, and for static routes it is 10. It is important to account for this when configuring your SD-WAN for 50/50 load balancing by setting the DHCP interface's distance to 10.

- b. Set the wan2 interface *IP/Netmask* to *10.100.20.1 255.255.255.0*.
2. Go to *Network > SD-WAN*, select the *SD-WAN Zones* tab, and click *Create New > SD-WAN Member*.
3. Set the *Interface* to *wan1*.
4. Leave *SD-WAN Zone* as *virtual-wan-link*.
5. As wan1 uses DHCP, leave *Gateway* set to *0.0.0.0*.

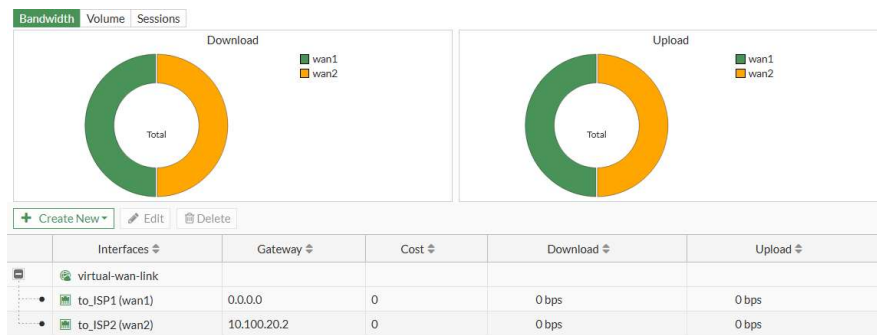
If IPv6 visibility is enabled in the GUI, an IPv6 gateway can also be added for each member. See [Feature visibility on page 3084](#) for details.

6. Leave *Cost* as *0*.

The *Cost* field is used by the Lowest Cost (SLA) strategy. The link with the lowest cost is chosen to pass traffic. The lowest possible *Cost* is 0.

7. Set *Status* to *Enable*, and click *OK*.

8. Repeat the above steps for wan2, setting *Gateway* to the ISP's gateway: *10.100.20.2*.



## Adding a static route

You must configure a default route for the SD-WAN. The default gateways for each SD-WAN member interface do not need to be defined in the static routes table. FortiGate will decide what route or routes are preferred using Equal Cost Multi-Path (ECMP) based on distance and priority.

### To create a static route for SD-WAN:

1. Go to *Network > Static Routes*.
2. Click *Create New*. The *New Static Route* page opens.
3. Set *Destination* to *Subnet*, and leave the IP address and subnet mask as *0.0.0.0/0.0.0.0*.
4. In the *Interface* field select an SD-WAN zone.

5. Ensure that *Status* is *Enabled*.
6. Click *OK*.

By default, a static route or the default route outgoing through an SD-WAN zone have an administrative distance of 1.



### To change the default distance in the CLI:

```
config router static
  edit <static-route-entry>
    set distance <AD>
  next
end
```

## Selecting the implicit SD-WAN algorithm

SD-WAN rules define specific routing options to route traffic to an SD-WAN member.

If no routing rules are defined, the default *Implicit* rule is used. It can be configured to use one of five different load balancing algorithms. See [Implicit rule on page 859](#) for more details and examples.

This example shows four methods to equally balance traffic between the two WAN connections. Go to *Network > SD-WAN*, select the *SD-WAN Rules* tab, and edit the *sd-wan* rule to select the method that is appropriate for your requirements.

- **Source IP** (CLI command: `source-ip-based`):  
Select this option to balance traffic equally between the SD-WAN members according to a hash algorithm based on the source IP addresses.
- **Session** (`weight-based`):  
Select this option to balance traffic equally between the SD-WAN members by the session numbers ratio among its members. Use weight 50 for each of the 2 members.
- **Source-Destination IP** (`source-dest-ip-based`):  
Select this option to balance traffic equally between the SD-WAN members according to a hash algorithm based on the source and destination IP addresses.
- **Volume** (`measured-volume-based`):  
Select this option to balance traffic equally between the SD-WAN members according to the bandwidth ratio among its members.

## Configuring firewall policies for SD-WAN

SD-WAN zones can be used in policies as source and destination interfaces. Individual SD-WAN members cannot be used in policies.

You must configure a policy that allows traffic from your organization's internal network to the SD-WAN zone. Policies configured with the SD-WAN zone apply to all SD-WAN interface members in that zone.

### To create a firewall policy for SD-WAN:

1. Go to *Policy & Objects > Firewall Policy*.
2. Click *Create New*. The *New Policy* page opens.

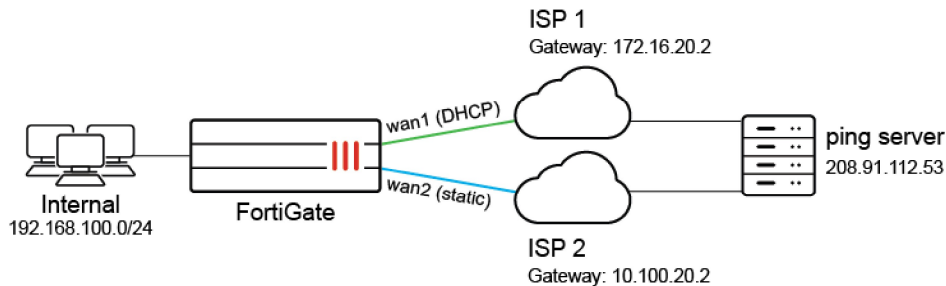
## 3. Configure the following:

|                                   |   |
|-----------------------------------|---|
| <b>Name</b>                       | Enter a name for the policy.  |
| <b>Incoming Interface</b>         | <i>internal</i>   |
| <b>Outgoing Interface</b>         | <i>virtual-wan-link</i>   |
| <b>Source</b>                     | <i>all</i>  |
| <b>Destination</b>                | <i>all</i>  |
| <b>Schedule</b>                   | <i>always</i>   |
| <b>Service</b>                    | <i>ALL</i>  |
| <b>Action</b>                     | <i>ACCEPT</i>   |
| <b>Firewall / Network Options</b> | Enable <i>NAT</i> and set <i>IP Pool Configuration</i> to <i>Use Outgoing Interface Address</i> .           |
| <b>Security Profiles</b>          | Apply profiles as required.   |
| <b>Logging Options</b>            | Enable <i>Log Allowed Traffic</i> and select <i>All Sessions</i> . This allows you to verify results later. |

## 4. Enable the policy, then click OK.

## Link monitoring and failover

Performance SLA link monitoring measures the health of links that are connected to SD-WAN member interfaces by sending probing signals through each link to a server, and then measuring the link quality based on latency, jitter, and packet loss. If a link is broken, the routes on that link are removed and traffic is routed through other links. When the link is working again, the routes are re-enabled. This prevents traffic being sent to a broken link and lost.



In this example, the detection server IP address is 208.91.112.53. A performance SLA is created so that, if ping fails per the metrics defined, the routes to that interface are removed and traffic is detoured to the other interface. The ping protocol is used, but other protocols could also be selected as required.

### To configure a performance SLA:

1. Go to *Network > SD-WAN*, select the *Performance SLAs* tab, and click *Create New*.
2. Enter a name for the SLA and set *Protocol* to *Ping*.
3. In the *Server* field, enter the detection server IP address (208.91.112.53 in this example).
4. In the *Participants* field, select *Specify* and add wan1 and wan2.

SLA targets are not required for link monitoring.

5. Configure the required metrics in *Link Status*.
6. Ensure that *Update static route* is enabled. This disables static routes for the inactive interface and restores routes on recovery.
7. Click *OK*.

## Results

The following GUI pages show the function of the SD-WAN and can be used to confirm that it is setup and running correctly:

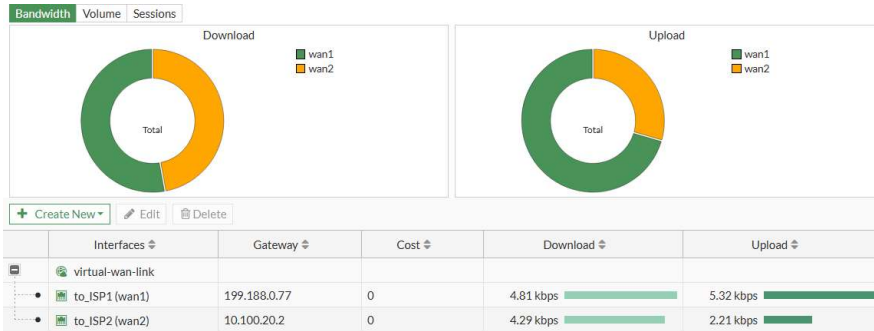
- [Interface usage on page 791](#)
- [Performance SLA on page 792](#)
- [Routing table on page 794](#)
- [Firewall policy on page 794](#)

## Interface usage

Go to *Network > SD-WAN* and select the *SD-WAN Zones* tab to review the SD-WAN interfaces' usage.

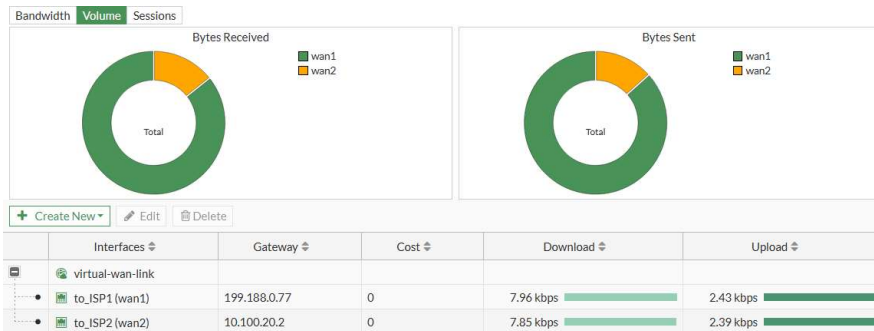
## Bandwidth

Select *Bandwidth* to view the amount of downloaded and uploaded data for each interface.



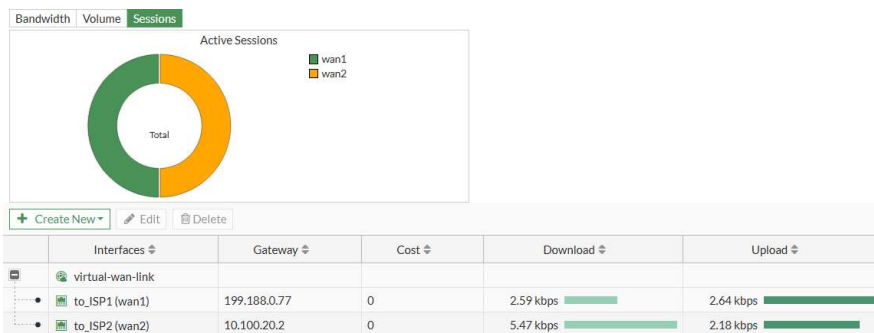
## Volume

Select *Volume* to see donut charts of the received and sent bytes on the interfaces.



## Sessions

Select *Sessions* to see a donut chart of the number of active sessions on each interface.

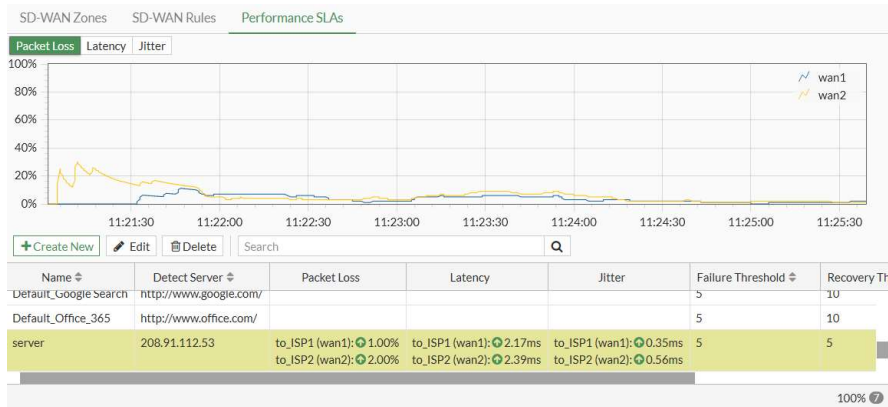


## Performance SLA

Go to *Network > SD-WAN*, select the *Performance SLAs* tab, and select the SLA from the table (*server* in this example) to view the packet loss, latency, and jitter on each SD-WAN member in the health check server.

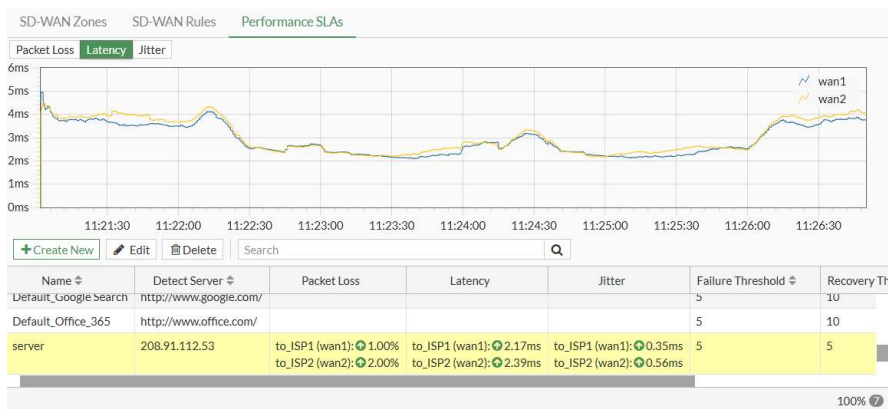
## Packet loss

Select *Packet Loss* to see the percentage of packets lost for each member.



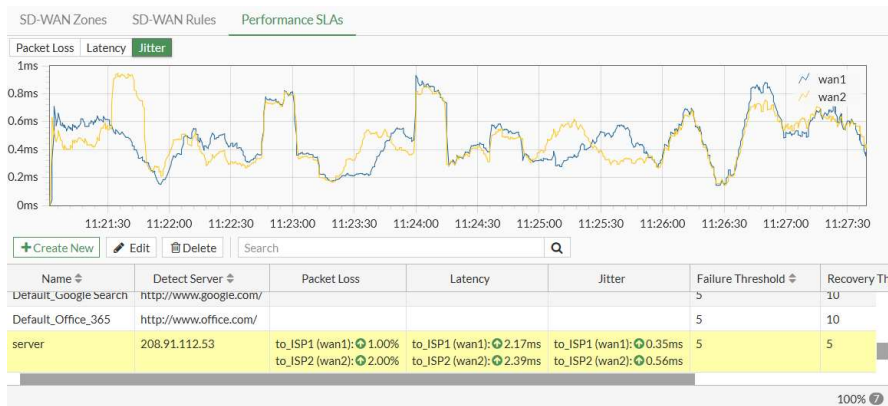
## Latency

Select *Latency* to see the current latency, in milliseconds, for each member.



## Jitter

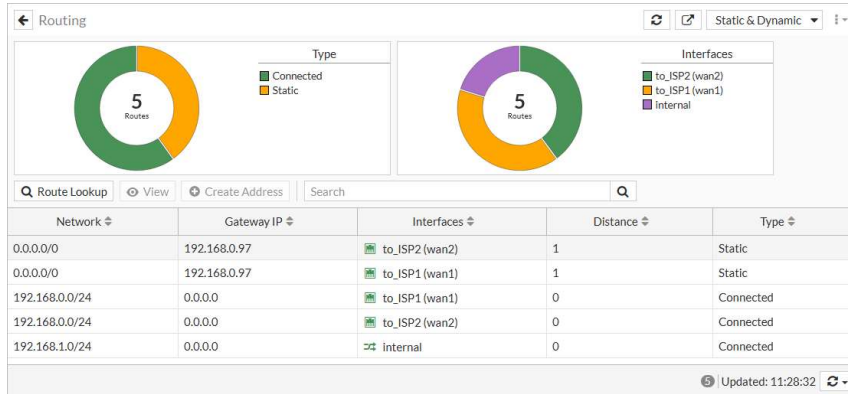
Select *Jitter* to see the jitter, in milliseconds, for each member.





## Routing table

Go to *Dashboard > Network*, expand the *Routing* widget, and select *Static & Dynamic* to review all static and dynamic routes. For more information about the widget, see [Static & Dynamic Routing monitor on page 117](#).



## Firewall policy

Go to *Policy & Objects > Firewall Policy* to review the SD-WAN policy.

| Name          | Source | Destination | Schedule | Service | Action | NAT     | Security Profiles | Log      | Bytes    |
|---------------|--------|-------------|----------|---------|--------|---------|-------------------|----------|----------|
| sd-wan        | all    | all         | always   | ALL     | ACCEPT | Enabled | no-inspection     | All      | 59.19 MB |
| Implicit Deny | all    | all         | always   | ALL     | DENY   |         |                   | Disabled | 1.27 kB  |

## Configuring SD-WAN in the CLI

This example can be entirely configured using the CLI.

### To configure SD-WAN in the CLI:

1. Configure the wan1 and wan2 interfaces:

```
config system interface
  edit "wan1"
    set alias to_ISP1
    set mode dhcp
    set distance 10
  next
  edit "wan2"
    set alias to_ISP2
    set ip 10.100.20.1 255.255.255.0
  next
end
```

**2. Enable SD-WAN and add the interfaces as members:**

```
config system sdwan
  set status enable
  config members
    edit 1
      set interface "wan1"
    next
    edit 2
      set interface "wan2"
      set gateway 10.100.20.2
    next
  end
end
```



If no SD-WAN zone is specified, members are added to the default *virtual-wan-link* zone.

---

**3. Create a static route for SD-WAN:**

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

**4. Select the implicit SD-WAN algorithm:**

```
config system sdwan
  set load-balance-mode {source-ip-based | weight-based | source-dest-ip-based |
  measured-volume-based}
end
```

**5. Create a firewall policy for SD-WAN:**

```
config firewall policy
  edit <policy_id>
    set name <policy_name>
    set srcintf "internal"
    set dstintf "virtual-wan-link"
    set srcaddr all
    set dstaddr all
    set action accept
    set schedule always
    set service ALL
    set utm-status enable
    set ssl-ssh-profile <profile_name>
    set av-profile <profile_name>
    set webfilter-profile <profile_name>
    set dnsfilter-profile <profile_name>
    set emailfilter-profile <profile_name>
    set ips_sensor <sensor_name>
    set application-list <app_list>
    set voip-profile <profile_name>
    set logtraffic all
    set nat enable
```

```

        set status enable
    next
end

```

## 6. Configure a performance SLA:

```

config system sdwan
    config health-check
        edit "server"
            set server "208.91.112.53"
            set update-static-route enable
            set members 1 2
        next
    end
end

```

## Results

### To view the routing table:

```

# get router info routing-table all

Routing table for VRF=0
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

S*    0.0.0.0/0 [1/0] via 172.16.20.2, wan1
      [1/0] via 10.100.20.2, wan2
C     10.100.20.0/24 is directly connected, wan2
C     172.16.20.2/24 is directly connected, wan1
C     192.168.0.0/24 is directly connected, internal

```

### To diagnose the Performance SLA status:

```

FGT # diagnose sys sdwan health-check
Health Check(server):
Seq(1): state(alive), packet-loss(0.000%) latency(15.247), jitter(5.231) sla_map=0x0
Seq(2): state(alive), packet-loss(0.000%) latency(13.621), jitter(6.905) sla_map=0x0

```

## SD-WAN members and zones

SD-WAN bundles interfaces together into zones. Interfaces are first configured as SD-WAN members. This does not change the interface, it just allows SD-WAN to reference the interface as a member. SD-WAN member interfaces can be any interface supported by FortiGates, such as physical ports, VLAN interfaces, LAGs, IPsec tunnels, GRE tunnels, IPIP tunnels, and FortiExtender interfaces. Once SD-WAN members are configured, they can be assigned to a zone. Zones are used in policies as source and destination interfaces, in static routes, and in SD-WAN rules.

Multiple zones can be used to group SD-WAN interfaces for logical scenarios, such as overlay and underlay interfaces. Using multiple zones in policies allows for more granular control over functions like resource access and UTM access. Individual SD-WAN member interfaces cannot be used directly in policies, but they can be moved between SD-WAN zones at any time. If a member interface requires a special SD-WAN consideration, it can be put into an SD-WAN zone by itself.

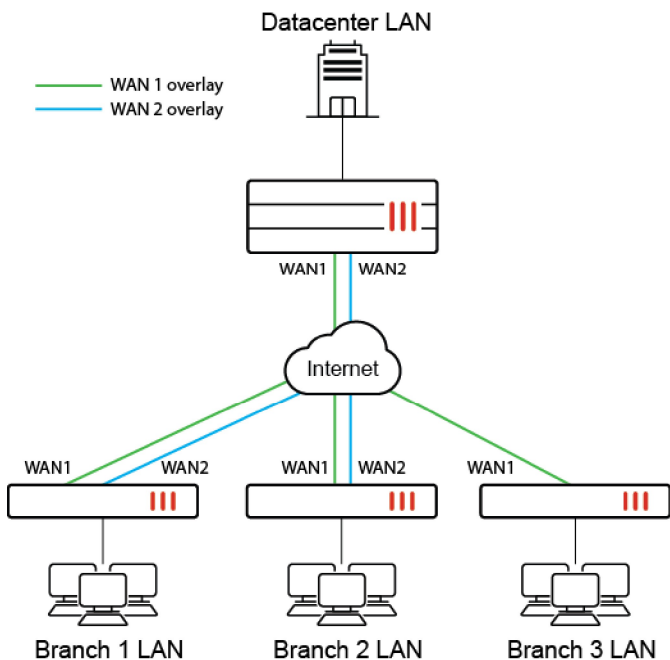
SD-WAN zones and members can be used in IPv4 and IPv6 static routes to make route configurations more flexible. SD-WAN zones and members can be used in SD-WAN rules to simplify the rule configuration. See [Specify an SD-WAN zone in static routes and SD-WAN rules on page 802](#) for more information.

When the Security Fabric is configured, SD-WAN zones are included in the Security Fabric topology views.

## Topology

This topology is used in the following procedures:

- [Configuring SD-WAN member interfaces](#)
- [Configuring SD-WAN zones](#)
- [Using SD-WAN zones](#)



## Configuring SD-WAN member interfaces

When configuring SD-WAN zones and members, it does not matter what order they are defined. In this example, the members are defined first, and they will be placed temporarily in the default zone called virtual-wan-link. A zone must be defined when creating a member, and the overlay and underlay zones will be created in the next procedure. It is standard practice to create SD-WAN members for each underlay and overlay interface, as most SD-WAN implementations apply SD-WAN intelligence to both underlay and overlay networks.

The following options can be configured for SD-WAN members:

| GUI option                  | CLI option                    | Description   |
|-----------------------------|-------------------------------|---|
| <i>Interface</i>            | <code>interface</code>        | Select the interface to use as an SD-WAN member. Optionally, select <i>None</i> in the GUI to not use an interface yet.   |
| <i>SD-WAN Zone</i>          | <code>zone</code>             | Select the destination zone if it exists at the time of member creation. Otherwise, the default virtual-wan-link zone is applied.<br>A new zone can be created within the GUI dropdown field.   |
| <i>Gateway/IPv6 Gateway</i> | <code>gateway/gateway6</code> | Enter the default gateway for the interface. For interfaces that already have a default gateway, such as those configured using DHCP, this field is pre-populated in the GUI.   |
| <i>Cost</i>                 | <code>cost</code>             | Enter the cost of the interface for services in SLA mode (0 - 4294967295, default = 0). A lower cost has a higher preference.   |
| <i>Priority</i>             | <code>priority</code>         | Enter the priority of the interface for IPv4 (1 - 65535, default = 1). The priority is used in the static route created for the SD-WAN member interface and in SD-WAN rules (including the implicit rule). When priority is used to determine the best route, the lower value takes precedence. |
| <i>Status</i>               | <code>status</code>           | Enable or disable the interface in SD-WAN.  |
| <i>n/a</i>                  | <code>source/source6</code>   | Set the source IP address used in the health check packet to the server.  |

### To configure the SD-WAN members and add them to the default zone in the GUI:

1. Go to *Network > SD-WAN*, select the *SD-WAN Zones* tab, and click *Create New > SD-WAN Member*.
2. Set the *Interface* to *WAN1*.
3. Leave the *SD-WAN Zone* as *virtual-wan-link*.

4. Click *OK*.
5. Repeat these steps to create SD-WAN members for the *WAN2*, *VPN1*, and *VPN2* interfaces.

### To configure the SD-WAN members and add them to the default zone in the CLI:

```

config system sdwan
  config members
    edit 1
      set interface "WAN1"
      set zone "virtual-wan-link"
    next
    edit 2
      set interface "WAN2"
      set zone "virtual-wan-link"
    next
    edit 3
      set interface "VPN1"
      set zone "virtual-wan-link"
    next
    edit 4
      set interface "VPN2"
      set zone "virtual-wan-link"
    next
  end
end

```

## Configuring SD-WAN zones

While SD-WAN zones are primarily used to logically group interfaces that are often used for the same purpose (such as WAN1 and WAN2), sometimes an SD-WAN zone can have a single member. This is due to the constraint that SD-WAN members may not be referenced directly in policies; however, SD-WAN members can be referenced directly in SD-WAN rules.

In this example, two zones named Overlay and Underlay are configured, and the member interfaces are added to their respective zones.

### To configure the SD-WAN zones in the GUI:

1. Go to *Network > SD-WAN* and select the *SD-WAN Zones* tab.
2. Click *Create New > SD-WAN Zone*.
3. Enter the *Name, Underlay*.
4. Set the *Interface members* to *WAN1* and *WAN2*.

The screenshot shows a 'New SD-WAN Zone' dialog box. The 'Name' field is filled with 'Underlay'. Below it, the 'Interface members' section shows a list with 'WAN1' and 'WAN2', each with a small 'x' icon to its right. To the right of the list is a plus sign. On the right side of the dialog, under 'Additional Information', there are several links: 'API Preview' (with an eye icon), 'Edit in CLI' (with a right-pointing arrow), 'Online Guides' (with a question mark icon), 'Relevant Documentation' (with a document icon and an external link icon), 'Video Tutorials' (with a video camera icon and an external link icon), 'FortiAnswers' (with a speech bubble icon), and 'Join the Discussion' (with a speech bubble icon and an external link icon). At the bottom of the dialog are 'OK' and 'Cancel' buttons.

5. Click *OK*.
6. Repeat these steps to configure the *Overlay* zone with members *VPN1* and *VPN2*.

## To configure the SD-WAN zones in the CLI:

### 1. Configure the SD-WAN zones:

```
config system sdwan
  config zone
    edit "Overlay"
    next
    edit "Underlay"
    next
  end
end
```

### 2. Add the member interfaces to their respective zones:

```
config system sdwan
  config members
    edit 1
      set interface WAN1
      set zone "Underlay"
    next
    edit 2
      set interface WAN2
      set zone "Underlay"
    next
    edit 3
      set interface VPN1
      set zone "Overlay"
    next
    edit 4
      set interface VPN2
      set zone "Overlay"
    next
  end
end
```



In the `config zone` settings, there is a `service-sla-tie-break` parameter that includes three options for the tie-break method used when multiple interfaces in a zone are eligible for traffic:

- `cfg-order`: members that meet the SLA are selected in the order they are configured (default).
- `fib-best-match`: members that meet the SLA are selected that match the longest prefix in the routing table.
- `input-device`: members that meet the SLA are selected by matching the input device.

See [Overlay stickiness on page 1093](#) for more information.

## Using SD-WAN zones

Once SD-WAN zones are defined, they can be used in firewall policies. This section covers three policy scenarios:

- [Datacenter resource access](#)
- [Direct internet access](#)

- [Remote internet access](#)



SD-WAN zones are a critical component of SD-WAN rules. See [Fields for configuring WAN intelligence on page 856](#) for more information.

## Datacenter resource access

Datacenter resources are made available through the VPN branches or overlay. In this example, there are two SD-WAN members in the overlay zone that the branch FortiGate can use to route traffic to and from the datacenter resource. The overlay zone is used as the destination in the firewall policy.

### To configure the firewall policy:

1. Go to *Policy & Objects > Firewall Policy* and click *Create New*.
2. Configure the following settings:

|                           |                   |
|---------------------------|-------------------|
| <b>Name</b>               | <i>DC_Access</i>  |
| <b>Incoming Interface</b> | <i>LAN</i>        |
| <b>Outgoing Interface</b> | <i>Overlay</i>    |
| <b>Source</b>             | <i>Branch_LAN</i> |
| <b>Destination</b>        | <i>DC_LAN</i>     |
| <b>Action</b>             | <i>ACCEPT</i>     |

3. Configure the other settings as needed.
4. Click *OK*.



This firewall policy allows traffic to any interfaces included in the zone. The SD-WAN rules contain the intelligence used to select which members in the zone to use.

## Direct internet access

Direct internet access (DIA) is how a branch may access resources contained on the public internet. This can be non-business resources (such as video streaming sites), or publically available business resources (such as vendor portals).

### To configure the firewall policy:

1. Go to *Policy & Objects > Firewall Policy* and click *Create New*.
2. Configure the following settings:

|                           |            |
|---------------------------|------------|
| <b>Name</b>               | <i>DIA</i> |
| <b>Incoming Interface</b> | <i>LAN</i> |



|                           |                   |
|---------------------------|-------------------|
| <b>Outgoing Interface</b> | <i>Underlay</i>   |
| <b>Source</b>             | <i>Branch_LAN</i> |
| <b>Destination</b>        | <i>all</i>        |
| <b>Action</b>             | <i>ACCEPT</i>     |

3. Configure the other settings as needed.
4. Click *OK*.

## Remote internet access

Remote internet access (RIA) is the ability for a branch location to route public internet access requests across the overlay and out one of the hub's (or datacenter's) WAN interfaces. This option is effective when a branch has a WAN circuit with a local ISP and a second circuit that is private, such as MPLS. When the WAN circuit goes down, it is possible to send traffic through the hub using the MPLS overlay.

### To configure the firewall policy:

1. Go to *Policy & Objects > Firewall Policy* and click *Create New*.
2. Configure the following settings:

|                           |                   |
|---------------------------|-------------------|
| <b>Name</b>               | <i>RIA</i>        |
| <b>Incoming Interface</b> | <i>LAN</i>        |
| <b>Outgoing Interface</b> | <i>Overlay</i>    |
| <b>Source</b>             | <i>Branch_LAN</i> |
| <b>Destination</b>        | <i>all</i>        |
| <b>Action</b>             | <i>ACCEPT</i>     |

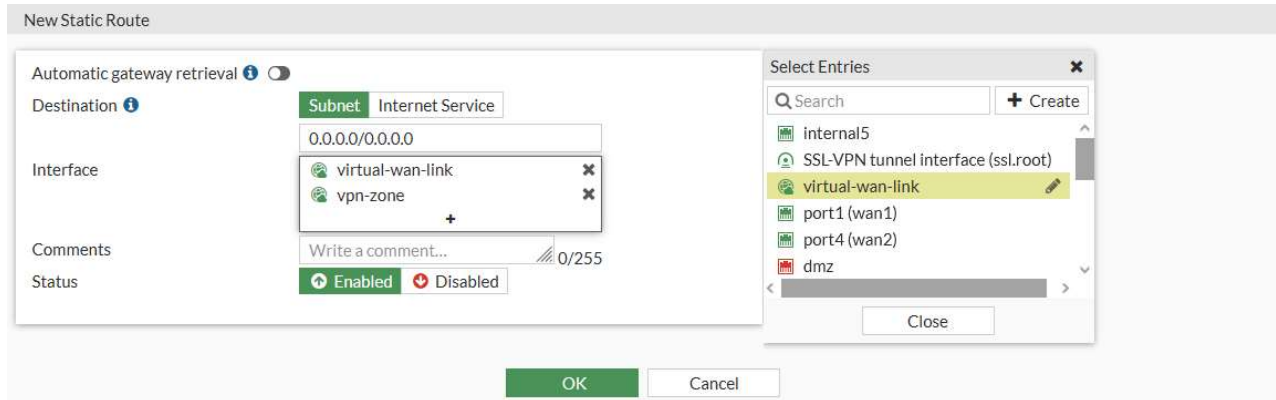
3. Configure the other settings as needed.
4. Click *OK*.

## Specify an SD-WAN zone in static routes and SD-WAN rules

SD-WAN zones can be used in IPv4 and IPv6 static routes, and in SD-WAN service rules. This makes route configuration more flexible, and simplifies SD-WAN rule configuration.

### To configure an SD-WAN zone in a static route in the GUI:

1. Go to *Network > Static Routes*
2. Edit an existing static route, or click *Create New* to create a new route.
3. Set *Interface* to one or more SD-WAN zones.



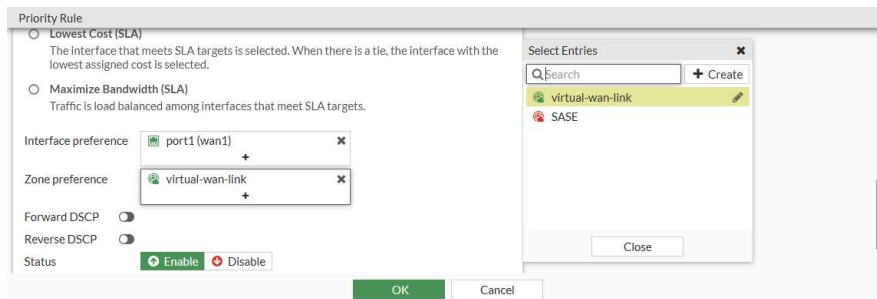
4. Configure the remaining settings are required.
5. Click **OK**.

### To configure an SD-WAN zone in a static route in the CLI:

```
config router {static | static6}
  edit 1
    set sdwan-zone <zone> <zone> ...
  next
end
```

### To configure an SD-WAN zone in an SD-WAN rule in the GUI:

1. Go to *Network > SD-WAN* and select the *SD-WAN Rules* tab
2. Edit an existing rule, or click *Create New* to create a new rule.
3. In the *Zone preference* field add one or more SD-WAN zones.



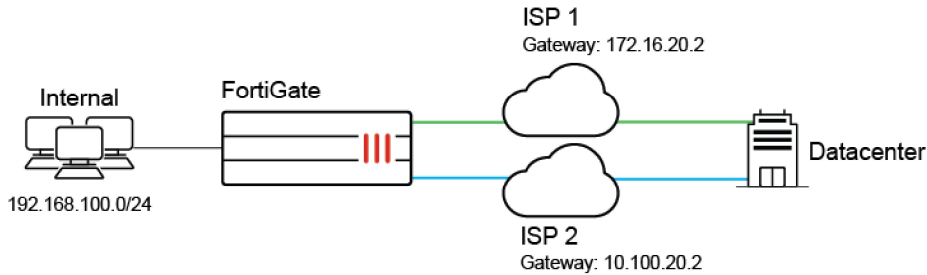
4. Configure the remaining settings are needed.
5. Click **OK**.

### To configure an SD-WAN zone in an SD-WAN rule in the CLI:

```
config system sdwan
  config service
    edit 1
      set priority-zone <zone>
    next
  end
end
```

## Examples

In these two examples, three SD-WAN members are created. Two members, port13 and port15, are in the default zone (*virtual-wan-link*), and the third member, to\_FG\_B\_root, is in the SASE zone.



### Example 1

In this example:

- Two service rules are created. Rule 1 uses the *virtual-wan-link* zone, and rule 2 uses the SASE zone.
- Two IPv4 static routes are created. The first route uses the *virtual-wan-link* zone, and the second route uses the SASE zone.

#### To configure the SD-WAN:

1. Assign port13 and port15 to the *virtual-wan-link* zone and to\_FG\_B\_root to the SASE zone:

```
config system sdwan
  set status enable
  config members
    edit 1
      set interface "port13"
      set zone "virtual-wan-link"
      set gateway 10.100.1.1
    next
    edit 2
      set interface "port15"
      set zone "virtual-wan-link"
      set gateway 10.100.1.5
    next
    edit 3
      set interface "to_FG_B_root"
      set zone "SASE"
    next
  end
end
```

2. Create two service rules, one for each SD-WAN zone:

```
config system sdwan
  config service
    edit 1
      set dst "10.100.20.0"
      set priority-zone "virtual-wan-link"
    next
```

```

        edit 2
            set internet-service enable
            set internet-service-name "Fortinet-FortiGuard"
            set priority-zone "SASE"
        next
    end
end

```

### 3. Configure static routes for each of the SD-WAN zones:

```

config router static
    edit 1
        set distance 1
        set sdwan-zone "virtual-wan-link"
    next
    edit 2
        set dst 172.16.109.0 255.255.255.0
        set distance 1
        set sdwan-zone "SASE"
    next
end

```

## To verify the results:

### 1. Check the service rule 1 diagnostics:

```

# diagnose sys sdwan service4 1

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
Members(2):
    1: Seq_num(1 port13), alive, selected
    2: Seq_num(2 port15), alive, selected
Dst address(1):
    10.100.20.0-10.100.20.255

```

Both members of the *virtual-wan-link* zone are selected. In manual mode, the interface members are selected based on the member configuration order. In SLA and priority mode, the order depends on the link status. If all of the link statuses pass, then the members are selected based on the member configuration order.

### 2. Check the service rule 2 diagnostics:

```

# diagnose sys sdwan service4 2

Service(2): Address Mode(IPV4) flags=0x200 use-shortcut-sla
Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
Members(1):
    1: Seq_num(3 to_FG_B_root), alive, selected
Internet Service(1): Fortinet-FortiGuard(1245324,0,0,0)

```

The member of the *SASE* zone is selected.

### 3. Review the routing table:

```

# get router info routing-table static
Routing table for VRF=0
S*    0.0.0.0/0 [1/0] via 10.100.1.1, port13
      [1/0] via 10.100.1.5, port15
S     172.16.109.0/24 [1/0] via 172.16.206.2, to_FG_B_root

```

The default gateway has the members from the *virtual-wan-link* zone, and the route to 172.16.10.9/24 has the single member from the *SASE* zone.

## Example 2

In this example, two IPv6 static routes are created. The first route uses the *virtual-wan-link* zone, and the second route uses the *SASE* zone.

### To configure the SD-WAN:

1. Configure port13 and port15 with IPv6 addresses and assign them to the *virtual-wan-link* zone, and assign to \_FG\_B\_root to the *SASE* zone:

```
config system sdwan
  set status enable
  config members
    edit 1
      set interface "port13"
      set zone "virtual-wan-link"
      set gateway6 2004:10:100:1::1
      set source6 2004:10:100:1::2
    next
    edit 2
      set interface "port15"
      set zone "virtual-wan-link"
      set gateway6 2004:10:100:1::5
      set source6 2004:10:100:1::6
    next
    edit 3
      set interface "to_FG_B_root"
      set zone "SASE"
    next
  end
end
```

2. Configure IPv6 static routes for each of the SD-WAN zones:

```
config router static6
  edit 1
    set distance 1
    set sdwan-zone "virtual-wan-link"
  next
  edit 2
    set dst 2003:172:16:109::/64
    set distance 1
    set sdwan-zone "SASE"
  next
end
```

### To verify the results:

1. Review the routing table:

```
# get router info6 routing-table static
Routing table for VRF=0
S*      ::/0 [1/0] via 2004:10:100:1::1, port13, 00:20:51, [1024/0]
```

```

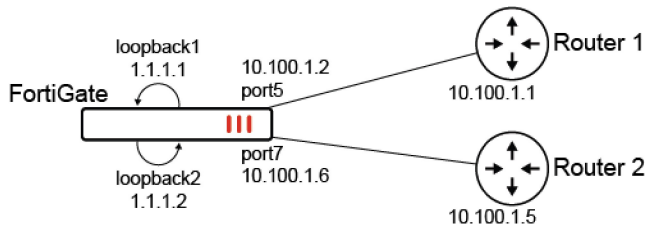
[1/0] via 2004:10:100:1::5, port15, 00:20:51, [1024/0]
S    2003:172:16:109::/64 [1/0] via ::ac10:ce02, to_FG_B_root, 00:20:51, [1024/0]
S    2003:172:16:209::/64 [5/0] via ::ac10:ce02, to_FG_B_root, 14:40:14, [1024/0]

```

The IPv6 default route includes the members from the *virtual-wan-link* zone, and the route to 2003:172:16:109::/64 has the single member from the *SASE* zone.

## Defining a preferred source IP for local-out egress interfaces on SD-WAN members

The preferred source IP can be configured on SD-WAN members so that local-out traffic is sourced from that IP. In the following example, two SD-WAN members (port5 and port6) will use loopback1 and loopback2 as sources instead of their physical interface address. A static route is created for destination 200.0.0.0/24 to use the virtual-wan-link. In turn, the FortiGate will create two ECMP routes to the member gateways and source the traffic from the loopback IPs.



### To configure preferred source IPs for SD-WAN members:

#### 1. Configure the SD-WAN members and other settings:

```

config system sdwan
  set status enable
  config zone
    edit "virtual-wan-link"
    next
  end
  config members
    edit 1
      set interface "port5"
      set gateway 10.100.1.1
      set preferred-source 1.1.1.1
      set source 1.1.1.1
    next
    edit 2
      set interface "port7"
      set gateway 10.100.1.5
      set preferred-source 1.1.1.2
      set source 1.1.1.2
    next
  end
end

```



In the SD-WAN `config members` settings, configuring the `source` for the health check probes is still required. SD-WAN adds dedicated kernel routes (`proto=17`) for the health checks using the interface IP or source IP when specified. To view the kernel routes, use `diagnose ip route list`.

**2. Configure the static route:**

```
config router static
  edit 2000
    set dst 200.0.0.0 255.255.255.0
    set distance 1
    set sdwan-zone "virtual-wan-link"
  next
end
```

**To verify the configuration:****1. Verify the kernel routing table for 200.0.0.0/24:**

```
# get router info kernel | grep -A 2 200.0.0.0/24
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0->200.0.0.0/24 pref=0.0.0.0
  gwy=10.100.1.1 flag=14 hops=255 oif=13(port5) pref=1.1.1.1
  gwy=10.100.1.5 flag=14 hops=254 oif=15(port7) pref=1.1.1.2
```

**2. Verify the routing table for 200.0.0.0/24:**

```
# get router info routing-table details 200.0.0.0/24
Routing table for VRF=0
Routing entry for 200.0.0.0/24
  Known via "static", distance 1, metric 0, best
  * vrf 0 10.100.1.1, via port5, prefersrc 1.1.1.1
  * vrf 0 10.100.1.5, via port7, prefersrc 1.1.1.2
```

**3. Run a sniffer trace after some traffic passes.****a. When traffic leaves port5:**

```
# diagnose sniffer packet any "host 200.0.0.1" 4
interfaces=[any]
filters=[host 200.0.0.1]
6.592488 port5 out 1.1.1.1 -> 200.0.0.1: icmp: echo request
7.592516 port5 out 1.1.1.1 -> 200.0.0.1: icmp: echo request
8.592532 port5 out 1.1.1.1 -> 200.0.0.1: icmp: echo request
```

**b. When traffic leaves port7:**

```
# diagnose sniffer packet any "host 200.0.0.1" 4
interfaces=[any]
filters=[host 200.0.0.1]
75.664173 port7 out 1.1.1.2 -> 200.0.0.1: icmp: echo request
76.664194 port7 out 1.1.1.2 -> 200.0.0.1: icmp: echo request
```

Traffic exiting each interface is sourced from the corresponding loopback IP.

## Performance SLA

Performance SLAs are used to measure the health of links that are connected to SD-WAN member interfaces by either sending probing signals through each link to a server, or using session information that is captured by firewall policies (see [Passive WAN health measurement on page 821](#) for information), and measuring the link quality based on latency, jitter, and packet loss. If a link fails all of the health checks, the routes on that link are removed from the SD-WAN link

load balancing group, and traffic is routed through other links. When the link passes SLA, the routes are reestablished. This prevents traffic from being sent to a broken link and getting lost.

The following topics provide instructions on configuring performance SLA:

- [Performance SLA overview on page 809](#)
- [Link health monitor on page 814](#)
- [Monitoring performance SLA on page 816](#)
- [Passive WAN health measurement on page 821](#)
- [Passive health-check measurement by internet service and application on page 826](#)
- [Mean opinion score calculation and logging in performance SLA health checks on page 831](#)
- [Embedded SD-WAN SLA information in ICMP probes on page 833](#)
- [SD-WAN application monitor using FortiMonitor on page 842](#)
- [Classifying SLA probes for traffic prioritization on page 846](#)

## Performance SLA overview

Performance SLAs consist of three parts:

- [Health checks](#)
- [SLA targets](#)
- [Link status](#)

### Health checks

A health check is defined by a [probe mode](#), [protocol](#), and [server](#). These three options specify what resource is being evaluated and how the evaluation is done. Each health check should be configured specifically for that resource, so the probe mode, protocol and server should be tailored for the particular service. For example, the health check for a VoIP service will differ than one for a database replication service.

Performance SLA participants are the interfaces that will be evaluated for a given health check. They must be SD-WAN member interfaces, but do not have to belong to the same zone. When selecting participants, only select participants that you expect the service communications to use. For example, a health check for a corporate resource might only use the overlay to access the service. Therefore, you would only add the VPN interfaces as participants.

There are six predefined performance SLA profiles for newly created VDOMs or factory reset FortiGate devices: AWS, DNS, FortiGuard, Gmail, Google Search, and Office 365. These performance SLA profiles provide Fortinet recommended settings for common services. To complete the performance SLA configuration, add the participants for the service. You can adjust the default settings to suit your needs.



Bandwidth limits and traffic prioritization can be enabled using SLA probe classification. The `class-id` command can be used to assign a class ID to SLA probes. Class IDs then guarantee that assigned bandwidth is honored when traffic congestion occurs. See [Classifying SLA probes for traffic prioritization on page 846](#).

---

### Probe mode

The probe mode can be set to active, passive, or prefer passive.



In active mode, the FortiGate sends a packet of the type specified by the protocol setting towards the defined server. This allows you to evaluate the path to the destination server using the protocol that matches the service provided by the server. Active probing does add some overhead in the form of health check probes (and additional configurations to define the probe type and server), but it has the benefit of constantly measuring the performance of the path to the server. This can be beneficial when reviewing historical data.

In passive mode, session information captured by firewall policies is used to determine latency, jitter, and packet loss. This has the added benefit of not generating additional traffic, and does not require the performance SLA to define a specific server for measurement. Instead, the SD-WAN rule must define the traffic to evaluate, and the firewall policy permitting the traffic must have a setting enabled. See [Passive WAN health measurement on page 821](#) and [Passive health-check measurement by internet service and application on page 826](#) for more information.

Prefer passive mode is a combination of active and passive modes. Health is measured using traffic when there is traffic, and using probes when there is no traffic. A protocol and server must be configured.

## Protocol

Health checks support a variety of protocols and protocol specific options. The most commonly used protocols (ping, HTTP, and DNS) can be configured in the GUI when creating a new performance SLA on the *Network > SD-WAN > Performance SLAs* page. The following protocols and options can be configured in the CLI using the `set protocol <option>` parameter:

|             |  |
|-------------|--|
| ping        | Use PING to test the link with the server.   |
| tcp-echo    | Use TCP echo to test the link with the server.   |
| udp-echo    | Use UDP echo to test the link with the server.   |
| http        | Use HTTP-GET to test the link with the server.   |
| https       | Use HTTP-GET to test the link with the server.   |
| twamp       | Use TWAMP to test the link with the server.  |
| dns         | Use DNS query to test the link with the server.<br>The FortiGate sends a DNS query for an A Record and the response matches the expected IP address.   |
| tcp-connect | Use a full TCP connection to test the link with the server.<br>The method to measure the quality of the TCP connection can be: <ul style="list-style-type: none"> <li>• <code>half-open</code>: FortiGate sends SYN and gets SYN-ACK. The latency is based on the round trip between SYN and SYN-ACK (default).</li> <li>• <code>half-close</code>: FortiGate sends FIN and gets FIN-ACK. The latency is based on the round trip between FIN and FIN-ACK.</li> </ul> |
| ftp         | Use FTP to test the link with the server.<br>The FTP mode can be: <ul style="list-style-type: none"> <li>• <code>passive</code>: The FTP health-check initiates and establishes the data connection (default).</li> <li>• <code>port</code>: The FTP server initiates and establishes the data connection.</li> </ul>  |



SD-WAN health checks can generate traffic that becomes quite high as deployments grow. Take this into consideration when setting DoS policy thresholds. For details on setting DoS policy thresholds, refer to [DoS policy on page 1365](#).



The default AWS, FortiGuard, Google Search, and Office 365 performance SLA profiles use HTTPS.

### To use UDP-echo and TCP-echo as health checks:

```
config system sdwan
  set status enable
  config health-check
    edit "h4_udpl"
      set protocol udp-echo
      set port 7
      set server <server>
    next
    edit "h4_tcp1"
      set protocol tcp-echo
      set port 7
      set server <server>
    next
    edit "h6_udpl"
      set addr-mode ipv6
      set server "2032::12"
      set protocol udp-echo
      set port 7
    next
  end
end
```

### To use DNS as a health check, and define the IP address that the response must match:

```
config system sdwan
  set status enable
  config health-check
    edit "h4_dns1"
      set protocol dns
      set dns-request-domain "ip41.forti2.com"
      set dns-match-ip 1.1.1.1
    next
    edit "h6_dns1"
      set addr-mode ipv6
      set server "2000::15.1.1.4"
      set protocol dns
      set port 53
      set dns-request-domain "ip61.xxx.com"
    next
  end
end
```

### To use TCP Open (SYN/SYN-ACK) and TCP Close (FIN/FIN-ACK) to verify connections:

```
config system sdwan
  set status enable
  config health-check
    edit "h4_tcpconnect1"
```

```

        set protocol tcp-connect
        set port 443
        set quality-measured-method {half-open | half-close}
        set server <server>
    next
    edit "h6_tcpconnect1"
        set addr-mode ipv6
        set server "2032::13"
        set protocol tcp-connect
        set port 444
        set quality-measured-method {half-open | half-close}
    next
end
end

```

### To use active or passive mode FTP to verify connections:

```

config system sdwan
    set status enable
    config health-check
        edit "h4_ftpl"
            set protocol ftp
            set port 21
            set user "root"
            set password *****
            set ftp-mode {passive | port}
            set ftp-file "1.txt"
            set server <server>
        next
        edit "h6_ftpl"
            set addr-mode ipv6
            set server "2032::11"
            set protocol ftp
            set port 21
            set user "root"
            set password *****
            set ftp-mode {passive | port}
            set ftp-file "2.txt"
        next
    end
end
end

```

Health check probe packets support DSCP markers for accurate link performance evaluation for high priority applications. This allows the probe packet to match the real traffic it is providing measurements for, including how that traffic is shaped by upstream devices based on the DSCP markers.

### To mark health check packets with DSCP:

```

config system sdwan
    config health-check
        edit <name>
            set diffservcode <6-bits_binary, 000000-111111>
            set protocol <option>
        next
    end
end
end

```

## Server

An IP address or FQDN can be defined as the server that the probe packets will be sent to. Up to two servers can be defined this way. When two servers are provided, both must fail in order for the health check to fail. This is to avoid a scenario where one remote server is down and causes a false positive that the link is down. The FortiGate can still use the interface associated with this health check to reach the remaining healthy server.

The purpose of the server is not simply to measure the health of the link, but rather the health of the path to a resource. It is highly recommended to use an IP address or FQDN that reflects the resource so the traffic path is considered.



A server can only be used in one performance SLA at any given time.

---

## SLA targets

SLA targets are a set of constraints that are used in SD-WAN rules to control the paths that traffic takes. The constraints are:

- Latency threshold: latency for SLA to make a decision, in milliseconds (0 - 10000000, default = 5).
- Jitter threshold: jitter for SLA to make a decision, in milliseconds (0 - 10000000, default = 5).
- Packet loss threshold: packet loss for SLA to make a decision, in percentage (0 - 100, default = 0).

These settings should be specific to the service whose performance is being considered. You should attempt to configure the constraints to be just under the maximum values for the application or service to function well. For example, if your application requires less than 100 ms latency, then you should configure the SLA target to be 90 ms. Misconfiguring these settings will cause the performance SLA to lose value. If the values are too tight, then you may have traffic flipping between links before necessary. If the values are too loose, then performance may be impacted and the FortiGate will do nothing about it.

In the GUI, one SLA target can be configured, but additional targets can be configured in the CLI. Once a second target is configured in the CLI, additional targets can be configured from the GUI. Multiple SLA targets can be configured where a server provides multiple services that have different values for acceptable performance. For example, Google provides a DNS service and entertainment services (YouTube), so it is necessary to configure multiple SLA targets in this case since you can only configure a server in one performance SLA.

## Link status

The *Link Status* section of the performance SLA configuration consists of three settings that determine the frequency that the link is evaluated, and the requirements to be considered valid or invalid:

- *Check interval*: the interval in which the FortiGate checks the interface, in milliseconds (500 - 3600000, default = 500).
- *Failures before inactive*: the number of failed status checks before the interface shows as inactive (1 - 3600, default = 5). This setting helps prevent flapping, where the system continuously transfers traffic back and forth between links.
- *Restore link after*: the number of successful status checks before the interface shows as active (1 - 3600, default = 5). This setting also helps prevent flapping.

When a participant becomes inactive, the performance SLA causes the FortiGate to withdraw all static routes associated with that interface. If there are multiple static routes using the same interface, they will all be withdrawn when the link monitor is failing.

## Link health monitor

Performance SLA link health monitoring measures the health of links that are connected to SD-WAN member interfaces by either sending probing signals through each link to a server, or using session information that is captured on firewall policies (see [Passive WAN health measurement on page 821](#) for information), and measuring the link quality based on latency, jitter, and packet loss. If a link fails all of the health checks, the routes on that link are removed from the SD-WAN link load balancing group, and traffic is routed through other links. When the link is working again the routes are reestablished. This prevents traffic being sent to a broken link and lost.

When an SD-WAN member has multiple health checks configured, all of the checks must fail for the routes on that link to be removed from the SD-WAN link load balancing group.

Two health check servers can be configured to ensure that, if there is a connectivity issue, the interface is at fault and not the server. A server can only be used in one health check.

The FortiGate uses the first server configured in the health check server list to perform the health check. If the first server is unavailable, then the second server is used. The second server continues to be used until it becomes unavailable, and then the FortiGate returns to the first server, if it is available. If both servers are unavailable, then the health check fails.

You can configure the protocol that is used for status checks, including: Ping, HTTP, HTTPS, DNS, TCP echo, UDP echo, two-way active measurement protocol (TWAMP), TCP connect, and FTP. In the GUI, only Ping, HTTP, and DNS are available.

You can view link quality measurements by going to *Network > SD-WAN* and selecting the *Performance SLAs* tab. The table shows the default health checks, the health checks that you configured, and information about each health check. The values shown in the *Packet Loss*, *Latency*, and *Jitter* columns are for the health check server that the FortiGate is currently using. The green up arrows indicate that the server is responding, and does not indicate if the health checks are being met. See [Results on page 791](#) for more information.

### To configure a link health monitor in the GUI:

1. Go to *Network > SD-WAN*, select the *Performance SLAs* tab, and click *Create New*.
2. Set a *Name* for the SLA.
3. If enabled in Feature Visibility, set the *IP Version*. *IPv6* does not support all of the protocols.
4. Set the *Probe mode*:
  - *Active*: Send probes to determine link quality.
  - *Passive*: Use traffic to determine link quality. Enable passive health checks in policies to allow measurement.
  - *Prefer Passive*: Same as passive mode, but send probes when there is no traffic.
5. Set the *Protocol* that you need to use for status checks: *Ping*, *HTTP*, or *DNS*.
6. Set *Server* to the IP addresses of up to two servers that all of the SD-WAN members in the performance SLA can reach. If the *Protocol* is *DNS*, set the *DNS Server* to either the same as the system DNS, or specify the primary and secondary DNS servers.
7. Set *Participants* to *All SD-WAN Members*, or select *Specify* to choose specific SD-WAN members.
8. Set *Enable probe packets* to enable or disable sending probe packets.
9. Configure *SLA Target*:

If the health check is used in an SD-WAN rule that uses *Manual* or *Best Quality* strategies, enabling *SLA Target* is optional. If the health check is used in an SD-WAN rule that uses *Lowest Cost (SLA)* or *Maximum Bandwidth (SLA)* strategies, then *SLA Target* is enabled.

When *SLA Target* is enabled, configure the following:

- *Latency threshold*: Calculated based on last 30 probes (default = 5ms).
- *Jitter threshold*: Calculated based on last 30 probes (default = 5ms).
- *Packet Loss threshold*: Calculated based on last 100 probes (default = 0%).

10. In the *Link Status* section configure the following:

- *Check interval*: The interval in which the FortiGate checks the interface, in milliseconds (500 - 3600000, default = 500).
- *Failures before inactive*: The number of failed status checks before the interface shows as inactive (1 - 3600, default = 5). This setting helps prevent flapping, where the system continuously transfers traffic back and forth between links
- *Restore link after*: The number of successful status checks before the interface shows as active (1 - 3600, default = 5). This setting helps prevent flapping, where the system continuously transfers traffic back and forth between links

11. In the *Actions when Inactive* section, enable *Update static route* to disable static routes for inactive interfaces and restore routes when interfaces recover.

12. Click **OK**.

### To configure a link health monitor in the CLI:

```
config system sdwan
  config health-check
    edit "PingSLA"
      set addr-mode {ipv4 | ipv6}
      set server <server1_IP_address> <server2_IP_address>
      set detect-mode {active | passive | prefer-passive}
      set protocol {ping | tcp-echo | udp-echo | http | https | twamp | dns | tcp-
connect | ftp}
      set ha-priority <integer>
      set probe-timeout <integer>
      set probe-count <integer>
```

```

set probe-packets {enable | disable}
set interval <integer>
set failtime <integer>
set recoverytime <integer>
set diffservcode <binary>
set update-static-route {enable | disable}
set update-cascade-interface {enable | disable}
set sla-fail-log-period <integer>
set sla-pass-log-period <integer>
set threshold-warning-packetloss <integer>
set threshold-alert-packetloss <integer>
set threshold-warning-latency <integer>
set threshold-alert-latency <integer>
set threshold-warning-jitter <integer>
set threshold-alert-jitter <integer>
set vrf <integer>
set source <ip address>
set members <member_number> ... <member_number>
config sla
  edit 1
    set link-cost-factor {latency jitter packet-loss}
    set latency-threshold <integer>
    set jitter-threshold <integer>
    set packetloss-threshold <integer>
  next
end
next
end
end

```

Additional settings are available for some of the protocols:

| Protocol    | Additional options   |
|-------------|--|
| http, https | port <port_number><br>http-get <url><br>http-agent <string><br>http-match <response_string>              |
| twamp       | port <port_number><br>security mode {none   authentication}<br>password <password><br>packet-size <size> |
| ftp         | ftp-mode {passive   port}<br>ftp-file <path>   |

For more examples see [Protocol](#).

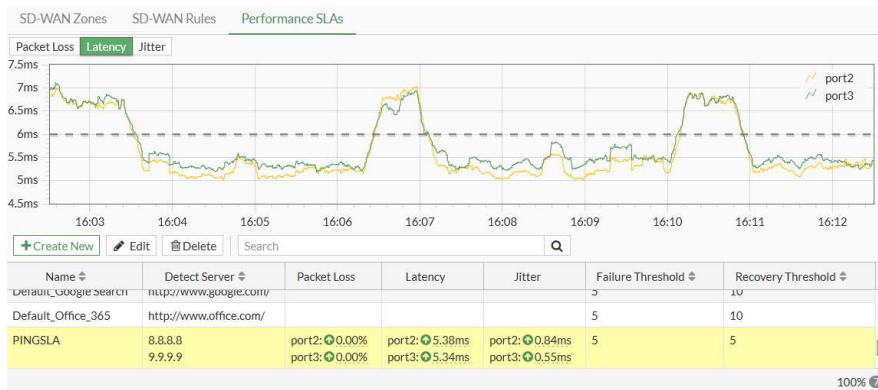
## Monitoring performance SLA

SD-WAN diagnostics can be used to help maintain your SD-WAN solution.

## Monitoring SD-WAN link quality status

Link quality plays a significant role in link selection for SD-WAN. Investigate any prolonged issues with packet loss, latency, or jitter to ensure that your network does not experience degraded performance or an outage.

You can monitor the link quality status of SD-WAN interface members by going to *Network > SD-WAN* and selecting the *Performance SLAs* tab.



The live charts show the packet loss, latency, or jitter for the selected health check. Hover the cursor over a line in the chart to see the specific value for that interface at that specific time.

The table shows information about each health check, including the configured servers, link quality data, and thresholds. The colored arrow indicates the status of the interface when the last status check was performed: green means that the interface was active, and red means that the interface was inactive. Hover the cursor over the arrow for additional information.

## Monitoring system event logs

The features adds an SD-WAN daemon function to keep a short, 10 minute history of SLA that can be viewed in the CLI.

Performance SLA results related to interface selection, session failover, and other information, can be logged. These logs can then be used for long-term monitoring of traffic issues at remote sites, and for reports and views in FortiAnalyzer.

The time intervals that Performance SLA fail and pass logs are generated in can be configured.

### To configure the fail and pass logs' generation time interval:

```
config system sdwan
  config health-check
    edit "PingSLA"
      set sla-fail-log-period 30
      set sla-pass-log-period 60
    next
  end
end
```



**To view the 10 minute Performance SLA link status history:**

```
FGDocs # diagnose sys sdwan sla-log PingSLA 1
Timestamp: Fri Sep  4 10:32:37 2020, vdom root, health-check PingSLA, interface: wan2,
status: up, latency: 4.455, jitter: 0.430, packet loss: 0.000%.
Timestamp: Fri Sep  4 10:32:37 2020, vdom root, health-check PingSLA, interface: wan2,
status: up, latency: 4.461, jitter: 0.436, packet loss: 0.000%.
Timestamp: Fri Sep  4 10:32:38 2020, vdom root, health-check PingSLA, interface: wan2,
status: up, latency: 4.488, jitter: 0.415, packet loss: 0.000%.
...
Timestamp: Fri Sep  4 10:42:36 2020, vdom root, health-check PingSLA, interface: wan2,
status: up, latency: 6.280, jitter: 0.302, packet loss: 0.000%.
Timestamp: Fri Sep  4 10:42:37 2020, vdom root, health-check PingSLA, interface: wan2,
status: up, latency: 6.261, jitter: 0.257, packet loss: 0.000%.
Timestamp: Fri Sep  4 10:42:37 2020, vdom root, health-check PingSLA, interface: wan2,
status: up, latency: 6.229, jitter: 0.245, packet loss: 0.000%.
```

**SLA pass logs**

The FortiGate generates Performance SLA logs at the specified pass log interval (`sla-pass-log-period`) when SLA passes.

```
date="2021-04-15" time="10:04:56" id=6951431609690095758 bid=52507 dvid=1047
itime=1618506296 euid=3 epid=3 dsteuid=3 dstepid=3 logver=700000066 logid="0113022925"
type="event" subtype="sdwan" level="information" msg="Health Check SLA status."
logdesc="SDWAN SLA information" status="up" interface="port1" eventtime=1618506296222639301
tz="-0700" eventtype="SLA" jitter="0.277" inbandwidthavailable="10.00Gbps"
outbandwidthavailable="10.00Gbps" bibandwidthavailable="20.00Gbps" packetloss="1.000%"
latency="186.071" slamap="0x1" healthcheck="BusinessCritical_CloudApps" slatargetid=1
outbandwidthused="40kbps" inbandwidthused="24kbps" bibandwidthused="64kbps"
devid="FGVM02TM20000000" vd="root" devname="Branch_Office_01" csf="fabric"

date="2021-04-15" time="10:04:56" id=6951431609690095759 bid=52507 dvid=1047
itime=1618506296 euid=3 epid=3 dsteuid=3 dstepid=3 logver=700000066 logid="0113022925"
type="event" subtype="sdwan" level="information" msg="Health Check SLA status."
logdesc="SDWAN SLA information" status="up" interface="port2" eventtime=1618506296223163068
tz="-0700" eventtype="SLA" jitter="0.204" inbandwidthavailable="10.00Gbps"
outbandwidthavailable="10.00Gbps" bibandwidthavailable="20.00Gbps" packetloss="0.000%"
latency="185.939" slamap="0x1" healthcheck="BusinessCritical_CloudApps" slatargetid=1
outbandwidthused="142kbps" inbandwidthused="23kbps" bibandwidthused="165kbps"
devid="FGVM02TM20000000" vd="root" devname="Branch_Office_01" csf="fabric"
```

In the FortiAnalyzer GUI:

| #  | ▲ Date/Time | Level       | Device ID      | Interface  | Status | Message                    |
|----|-------------|-------------|----------------|------------|--------|----------------------------|
| 19 | 10:04:38    | information | FGVM02TM200... | port1      | up     | Health Check SLA status.   |
| 20 | 10:04:38    | information | FGVM02TM200... | port2      | up     | Health Check SLA status.   |
| 21 | 10:04:39    | notice      | FGVM02TM200... | To-HQ-MPLS | down   | Health Check SLA status. S |
| 22 | 10:04:42    | notice      | FGVM02TM200... | To-HQ-MPLS | down   | Health Check SLA status. S |
| 23 | 10:04:49    | notice      | FGVM02TM200... | To-HQ-MPLS | down   | Health Check SLA status. S |
| 24 | 10:04:53    | notice      | FGVM02TM200... | To-HQ-MPLS | down   | Health Check SLA status. S |
| 25 | 10:04:56    | information | FGVM02TM200... | port1      | up     | Health Check SLA status.   |
| 26 | 10:04:56    | information | FGVM02TM200... | port2      | up     | Health Check SLA status.   |
| 27 | 10:04:58    | information | FGVM02TM200... | port1      | up     | Health Check SLA status.   |
| 28 | 10:04:58    | information | FGVM02TM200... | port2      | up     | Health Check SLA status.   |
| 29 | 10:04:58    | notice      | FGVM02TM200... | To-HQ-MPLS | down   | Health Check SLA status. S |
| 30 | 10:05:03    | notice      | FGVM02TM200... | To-HQ-MPLS | down   | Health Check SLA status. S |
| 31 | 10:05:09    | notice      | FGVM02TM200... | To-HQ-MPLS | down   | Health Check SLA status. S |
| 32 | 10:05:13    | notice      | FGVM02TM200... | To-HQ-MPLS | down   | Health Check SLA status. S |
| 33 | 10:05:15    | information | FGVM02TM200... | port1      | up     | Health Check SLA status.   |
| 34 | 10:05:15    | information | FGVM02TM200... | port2      | up     | Health Check SLA status.   |
| 35 | 10:05:18    | information | FGVM02TM200... | port1      | up     | Health Check SLA status.   |
| 36 | 10:05:18    | information | FGVM02TM200... | port2      | up     | Health Check SLA status.   |

## SLA fail logs

The FortiGate generates Performance SLA logs at the specified fail log interval (`sla-fail-log-period`) when SLA fails.

```
date="2021-04-15" time="10:04:59" id=6951431618280030243 bid=52507 dvid=1047
itime=1618506298 euid=3 epid=3 dsteuid=3 dstepid=3 logver=700000066 logid="0113022925"
type="event" subtype="sdwan" level="notice" msg="Health Check SLA status. SLA failed due to
being over the performance metric threshold." logdesc="SDWAN SLA information" status="down"
interface="To-HQ-MPLS" eventtime=1618506299718862835 tz="-0700" eventtype="SLA"
jitter="0.000" inbandwidthavailable="10.00Gbps" outbandwidthavailable="10.00Gbps"
bibandwidthavailable="20.00Gbps" packetloss="100.000%" latency="0.000" slamap="0x0"
healthcheck="BusinessCritical_CloudApps" slatargetid=1 metric="packetloss"
outbandwidthused="0kbps" inbandwidthused="0kbps" bibandwidthused="0kbps"
devid="FGVM02TM20000000" vd="root" devname="Branch_Office_01" csf="fabric"
```

```
date="2021-04-15" time="10:05:03" id=6951431639754866704 bid=52514 dvid=1046
itime=1618506303 euid=3 epid=3 dsteuid=3 dstepid=3 logver=700000066 logid="0113022925"
type="event" subtype="sdwan" level="notice" msg="Health Check SLA status. SLA failed due to
being over the performance metric threshold." logdesc="SDWAN SLA information" status="down"
interface="To-HQ-MPLS" eventtime=1618506304085863643 tz="-0700" eventtype="SLA"
jitter="0.000" inbandwidthavailable="10.00Gbps" outbandwidthavailable="10.00Gbps"
bibandwidthavailable="20.00Gbps" packetloss="100.000%" latency="0.000" slamap="0x0"
healthcheck="BusinessCritical_CloudApps" slatargetid=1 metric="packetloss"
outbandwidthused="6kbps" inbandwidthused="3kbps" bibandwidthused="9kbps"
devid="FGVM02TM20000000" vd="root" devname="Branch_Office_02" csf="fabric"
```

In the FortiAnalyzer GUI:

| #  | Date/Time | Level       | Device ID      | Interface  | Status | Message                    |
|----|-----------|-------------|----------------|------------|--------|----------------------------|
| 15 | 10:04:28  | notice      | FGVM02TM200... | To-HQ-MPLS | down   | Health Check SLA status. S |
| 16 | 10:04:32  | notice      | FGVM02TM200... | To-HQ-MPLS | down   | Health Check SLA status. S |
| 17 | 10:04:35  | information | FGVM02TM200... | port1      | up     | Health Check SLA status.   |
| 18 | 10:04:35  | information | FGVM02TM200... | port2      | up     | Health Check SLA status.   |
| 19 | 10:04:38  | information | FGVM02TM200... | port1      | up     | Health Check SLA status.   |
| 20 | 10:04:38  | information | FGVM02TM200... | port2      | up     | Health Check SLA status.   |
| 21 | 10:04:39  | notice      | FGVM02TM200... | To-HQ-MPLS | down   | Health Check SLA status. S |
| 22 | 10:04:42  | notice      | FGVM02TM200... | To-HQ-MPLS | down   | Health Check SLA status. S |
| 23 | 10:04:49  | notice      | FGVM02TM200... | To-HQ-MPLS | down   | Health Check SLA status. S |
| 24 | 10:04:53  | notice      | FGVM02TM200... | To-HQ-MPLS | down   | Health Check SLA status. S |
| 25 | 10:04:56  | information | FGVM02TM200... | port1      | up     | Health Check SLA status.   |
| 26 | 10:04:56  | information | FGVM02TM200... | port2      | up     | Health Check SLA status.   |
| 27 | 10:04:58  | information | FGVM02TM200... | port1      | up     | Health Check SLA status.   |
| 28 | 10:04:58  | information | FGVM02TM200... | port2      | up     | Health Check SLA status.   |
| 29 | 10:04:58  | notice      | FGVM02TM200... | To-HQ-MPLS | down   | Health Check SLA status. S |
| 30 | 10:05:03  | notice      | FGVM02TM200... | To-HQ-MPLS | down   | Health Check SLA status. S |
| 31 | 10:05:09  | notice      | FGVM02TM200... | To-HQ-MPLS | down   | Health Check SLA status. S |
| 32 | 10:05:13  | notice      | FGVM02TM200... | To-HQ-MPLS | down   | Health Check SLA status. S |
| 33 | 10:05:15  | information | FGVM02TM200... | port1      | up     | Health Check SLA status.   |

## Monitoring using the REST API

SLA log and interface information can be monitored using the REST API. This feature is also used by FortiManager as part of its detailed SLA monitoring and drilldown features.

| API call         | URL   |
|------------------|---|
| Interface log    | <a href="https://172.172.172.9/api/v2/monitor/virtual-wan/interface-log">https://172.172.172.9/api/v2/monitor/virtual-wan/interface-log</a> |
| SLA log          | <a href="https://172.172.172.9/api/v2/monitor/virtual-wan/sla-log">https://172.172.172.9/api/v2/monitor/virtual-wan/sla-log</a>             |
| Health check log | <a href="https://172.172.172.9/api/v2/monitor/virtual-wan/health-check">https://172.172.172.9/api/v2/monitor/virtual-wan/health-check</a>   |

A comprehensive list of API calls with sample output is available on the [Fortinet Developer Network](#).

### CLI diagnose commands:

```
# diagnose sys sdwan intf-sla-log port13
Timestamp: Wed Jan 9 18:33:49 2019, used inbandwidth: 3208bps, used outbandwidth:
3453bps, used bibandwidth: 6661bps, tx bytes: 947234bytes, rx bytes: 898622bytes.
Timestamp: Wed Jan 9 18:33:59 2019, used inbandwidth: 3317bps, used outbandwidth:
3450bps, used bibandwidth: 6767bps, tx bytes: 951284bytes, rx bytes: 902937bytes.
Timestamp: Wed Jan 9 18:34:09 2019, used inbandwidth: 3302bps, used outbandwidth:
3389bps, used bibandwidth: 6691bps, tx bytes: 956268bytes, rx bytes: 907114bytes.
Timestamp: Wed Jan 9 18:34:19 2019, used inbandwidth: 3279bps, used outbandwidth:
3352bps, used bibandwidth: 6631bps, tx bytes: 958920bytes, rx bytes: 910793bytes.
Timestamp: Wed Jan 9 18:34:29 2019, used inbandwidth: 3233bps, used outbandwidth:
3371bps, used bibandwidth: 6604bps, tx bytes: 964374bytes, rx bytes: 914854bytes.
Timestamp: Wed Jan 9 18:34:39 2019, used inbandwidth: 3235bps, used outbandwidth:
3362bps, used bibandwidth: 6597bps, tx bytes: 968250bytes, rx bytes: 918846bytes.
Timestamp: Wed Jan 9 18:34:49 2019, used inbandwidth: 3165bps, used outbandwidth:
3362bps, used bibandwidth: 6527bps, tx bytes: 972298bytes, rx bytes: 922724bytes.
Timestamp: Wed Jan 9 18:34:59 2019, used inbandwidth: 3184bps, used outbandwidth:
3362bps, used bibandwidth: 6546bps, tx bytes: 977282bytes, rx bytes: 927019bytes.

# diagnose sys sdwan sla-log ping 1 spoke11-p1_0
Timestamp: Wed Mar 3 15:35:20 2021, vdom root, health-check ping, interface: spoke11-
p1_0, status: up, latency: 0.135, jitter: 0.029, packet loss: 0.000%.
```

```
# diagnose sys sdwan sla-log ping 2 spoke12-p1_0
Timestamp: Wed Mar  3 15:36:08 2021, vdom root, health-check ping, interface: spoke12-
p1_0, status: up, latency: 0.095, jitter: 0.010, packet loss: 0.000%.

# diagnose sys sdwan health-check
Health Check(ping):
Seq(1 spoke11-p1): state(alive), packet-loss(0.000%) latency(0.156), jitter(0.043) sla_
map=0x1
Seq(1 spoke11-p1_0): state(alive), packet-loss(0.000%) latency(0.128), jitter(0.024)
sla_map=0x1
Seq(2 spoke12-p1): state(alive), packet-loss(0.000%) latency(0.125), jitter(0.028) sla_
map=0x1
Seq(2 spoke12-p1_0): state(alive), packet-loss(0.000%) latency(0.093), jitter(0.008)
sla_map=0x1
```

## Passive WAN health measurement

SD-WAN passive WAN health measurement determines the health check measurements (jitter, latency, and packet loss) using session information captured from the firewall policies that have *Passive Health Check* (`passive-wan-health-measurement`) enabled. Passive measurements analyze session information that is gathered from various TCP sessions can be viewed using the command `diagnose sys link-monitor-passive admin list by-interface`.

Using passive WAN health measurement reduces the amount of configuration required and decreases the traffic that is produced by health check monitor probes doing active measurements. Passive WAN health measurement analyzes real-life traffic; active WAN health measurement using a detection server might not reflect the real-life traffic.

By default, active WAN health measurement is enabled when a new health check is created. It can be changed to passive or prefer passive:

|                       |  |
|-----------------------|--|
| <b>passive</b>        | Health is measured using live traffic passing through an SD-WAN link to determine link metrics (jitter, latency, and packet loss) of participating SD-WAN links. No link health monitor needs to be configured.  |
| <b>prefer-passive</b> | Health is measured using live traffic when there is traffic passing through an SD-WAN link to determine link metrics (jitter, latency, and packet loss). If there is no live traffic flowing through an SD-WAN link for three continuous minutes, then the FortiGate sends out active probes to the configured health check server ( <code>set server</code> ) to calculate the link metrics. A link health monitor must be configured, see <a href="#">Link health monitor</a> for details. |

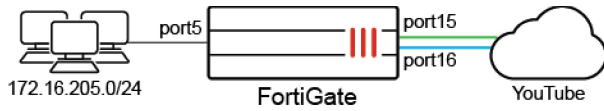


When `passive-wan-health-measurement` is enabled, `auto-asic-offload` will be disabled.

## Example

In this example, the FortiGate is configured to load-balance between two WAN interfaces, port15 and port16. A health check is configured in passive mode, and SLA thresholds are set. Passive WAN health measurement is enabled on the SD-WAN policy.

Measurements are taken from YouTube traffic generated by the PC. When latency is introduced to the traffic on port15, the passive health check trigger threshold is exceeded and traffic is rerouted to port16.



### To configure the SD-WAN in the GUI:

1. Create the SD-WAN zone:
  - a. Go to *Network > SD-WAN* and select the *SD-WAN Zones* tab.
  - b. Click *Create New > SD-WAN Zone*.
  - c. Enter a name for the zone, such as *SD-WAN*.
  - d. Click *OK*.
2. Create the SD-WAN members:
  - a. Go to *Network > SD-WAN* and select the *SD-WAN Zones* tab.
  - b. Click *Create New > SD-WAN Member*.
  - c. Set *Interface* to *port15*, *SD-WAN Zone* to *SD-WAN*, and *Gateway* set to *172.16.209.2*.
  - d. Click *OK*.
  - e. Click *Create New > SD-WAN Member* again.
  - f. Set *Interface* to *port16*, *SD-WAN Zone* to *SD-WAN*, and *Gateway* set to *172.16.210.2*.
  - g. Click *OK*.
3. Create a performance SLA:
  - a. Go to *Network > SD-WAN* and select the *Performance SLAs* tab.
  - b. Edit an existing health check, or create a new one.
  - c. Set *Probe mode* to *Passive*.
  - d. Set *Participants* to *Specify* and add *port15* and *port16*.
  - e. Configure two SLA targets. Note that the second SLA target must be configured in the CLI.

The screenshot shows the 'New Performance SLA' configuration window. The 'Name' field is set to 'Passive\_Check'. The 'Probe mode' is set to 'Passive'. The 'Participants' are set to 'All SD-WAN Members' and 'Specify', with 'port15' and 'port16' listed. The 'SLA Targets' section shows two targets: Target 1 and Target 2. Target 1 has a Latency threshold of 500 ms, Jitter threshold of 500 ms, and Packet Loss threshold of 10%. Target 2 has a Latency threshold of 1000 ms, Jitter threshold of 1000 ms, and Packet Loss threshold of 10%. The 'Actions when Inactive' section has 'Update static route' checked. The 'Additional Information' section on the right includes links for API Preview, Performance SLA Setup Guides, Link Monitoring, SLA Targets, Online Help, and Video Tutorials. The 'OK' button is highlighted in green.

- f. Configure the remaining settings as needed.

- g. Click *OK*.

The SLA list shows the probe mode in the *Detect Server* column, if the probe mode is passive or prefer passive.



Probe packets can only be disabled in the CLI and when the probe mode is not passive.

4. Create SD-WAN rules:

- a. Go to *Network > SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.  
 b. Configure the first rule:

|                             |   |
|-----------------------------|---|
| <b>Name</b>                 | Background_Traffic  |
| <b>Source address</b>       | 172.16.205.0  |
| <b>Application</b>          | Click in the field, and in the <i>Select Entries</i> pane search for <i>YouTube</i> and select all of the entries |
| <b>Strategy</b>             | Maximize Bandwidth (SLA)  |
| <b>Interface preference</b> | port15 and port16   |
| <b>Required SLA target</b>  | Passive_Check#2   |

- c. Click *OK*.  
 d. Click *Create New* again and configure the second rule:

|                             |                    |
|-----------------------------|--------------------|
| <b>Name</b>                 | Foreground_Traffic |
| <b>Source address</b>       | 172.16.205.0       |
| <b>Address</b>              | all                |
| <b>Protocol number</b>      | Specify - 1        |
| <b>Strategy</b>             | Lowest Cost (SLA)  |
| <b>Interface preference</b> | port15 and port16  |
| <b>Required SLA target</b>  | Passive_Check#1    |

- e. Click *OK*.

#### To configure the firewall policy in the GUI:

1. Go to *Policy & Objects > Firewall Policy* and click *Create New*.  
 2. Configure the policy:

|                           |                  |
|---------------------------|------------------|
| <b>Name</b>               | SD-WAN-HC-policy |
| <b>Incoming Interface</b> | port5            |
| <b>Outgoing Interface</b> | SD-WAN           |
| <b>Source</b>             | all              |
| <b>Destination</b>        | all              |

|                             |  |
|-----------------------------|--|
| <b>Schedule</b>             | always   |
| <b>Service</b>              | ALL  |
| <b>Action</b>               | ACCEPT   |
| <b>Passive Health Check</b> | Enabled<br>Passive health check can only be enabled in a policy when the outgoing interface is an SD-WAN zone. |

3. Click *OK*.

### To configure the SD-WAN in the CLI:

```

config system sdwan
  set status enable
  config zone
    edit "SD-WAN"
    next
  end
  config members
    edit 1
      set zone "SD-WAN"
      set interface "port15"
      set gateway 172.16.209.2
    next
    edit 2
      set zone "SD-WAN"
      set interface "port16"
      set gateway 172.16.210.2
    next
  end
  config health-check
    edit "Passive_Check"
      set detect-mode passive
      set members 1 2
      config sla
        edit 1
          set latency-threshold 500
          set jitter-threshold 500
          set packetloss-threshold 10
        next
        edit 2
          set latency-threshold 1000
          set jitter-threshold 1000
          set packetloss-threshold 10
        next
      end
    next
  end
  config service
    edit 1
      set name "Background_Traffic"
      set mode sla
      set load-balance enable
      set src "172.16.205.0"

```

```

        set internet-service enable
        set internet-service-app-ctrl 31077 33321 41598 31076 33104 23397 30201 16420
17396 38569 25564
        config sla
            edit "Passive_Check"
                set id 2
            next
        end
        set priority-member 1 2
    next
    edit 2
        set name "Foreground_Traffic"
        set mode sla
        set src "172.16.205.0"
        set protocol 1
        set dst "all"
        config sla
            edit "Passive_Check"
                set id 1
            next
        end
        set priority-member 1 2
    next
end
end

```

### To configure the firewall policy in the CLI:

```

config firewall policy
    edit 1
        set name "SD-WAN-HC-policy"
        set srcintf "port5"
        set dstintf "SD-WAN"
        set nat enable
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set passive-wan-health-measurement enable
        set auto-asic-offload disable
    next
end

```

## Results

### When both links pass the SLA:

```

# diagnose sys link-monitor-passive admin list by-interface
Interface port16 (28):
    Default(0x00000000): latency=10.0    15:46:36, jitter=5.0    15:46:37, pktloss=0.0 %
10:09:21

```

```

Interface port15 (27):

```



```

Default(0x00000000): latency=60.0    15:46:36, jitter=0.0    15:46:37, pktloss=0.0  %
10:39:24

# diagnose sys sdwan health-check
Health Check(Passive_Check):
Seq(1 port15): state(alive), packet-loss(0.000%) latency(60.000), jitter(0.750) sla_map=0x3
Seq(2 port16): state(alive), packet-loss(0.000%) latency(10.000), jitter(5.000) sla_map=0x3

# diagnose sys sdwan service4 2

Service(2): Address Mode(IPV4) flags=0x200
Gen(1), TOS(0x0/0x0), Protocol(1: 1->65535), Mode(sla), sla-compare-order
Members(2):
  1: Seq_num(1 port15), alive, sla(0x1), gid(0), cfg_order(0), cost(0), selected
  2: Seq_num(2 port16), alive, sla(0x1), gid(0), cfg_order(1), cost(0), selected
Src address(1):
  172.16.205.0-172.16.205.255

Dst address(1):
  8.8.8.8-8.8.8.8

```

### When the latency is increased to 610ms on port15, the SLA is broken and pings are sent on port16:

```

# diagnose sys sdwan health-check
Health Check(Passive_Check):
Seq(1 port15): state(alive), packet-loss(0.000%) latency(610.000), jitter(2.500) sla_map=0x3
Seq(2 port16): state(alive), packet-loss(0.000%) latency(50.000), jitter(21.000) sla_map=0x3

# diagnose sys sdwan service4 2

Service(2): Address Mode(IPV4) flags=0x200
Gen(6), TOS(0x0/0x0), Protocol(1: 1->65535), Mode(sla), sla-compare-order
Members(2):
  1: Seq_num(2 port16), alive, sla(0x1), gid(1), cfg_order(1), cost(0), selected
  2: Seq_num(1 port15), alive, sla(0x0), gid(2), cfg_order(0), cost(0), selected
Src address(1):
  172.16.205.0-172.16.205.255

Dst address(1):
  8.8.8.8-8.8.8.8

```

## Passive measurement

Passive measurement allows SLA information per internet service/application to be differentiated and collected when internet services/applications are defined in an SD-WAN rule that uses passive or prefer passive SLA. The SLA metrics (jitter, latency, and packet loss) on each SD-WAN member in the rule are calculated based on the relevant internet services/applications SLA information. These metrics help analyze the performance of different applications using the same WAN link. See [Passive health-check measurement by internet service and application on page 826](#) for more information.

## Passive health-check measurement by internet service and application

Active probing relies on checking the performance metrics of underlying infrastructure using layer 3 probes (ping) and layer 4 probes (tcp-echo, http, dns, and others) to provide limited information about an application's true performance.

Passive WAN health measurement uses passive probing to provide more realistic application performance information by collecting the performance metrics (jitter, latency, and packet loss) of live traffic that is passing through the firewall policies. See [Passive WAN health measurement on page 821](#).

Different applications can have different performance on the same WAN link, depending on the application's implementation. Passive measurement can be used to measure the performance of different internet services/applications that use the same WAN link.

The following is required:

**1. Firewall policy configuration:**

- Enable passive WAN health measurement (`set passive-wan-health-measurement enable`).
- Disable hardware offloading (`set auto-asic-offload disable`).
- Use an application control security profile to identify applications.

**2. SD-WAN rule configuration:**

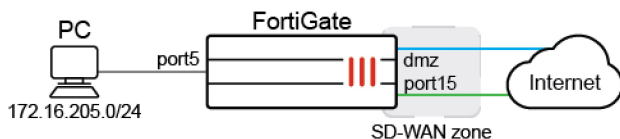
- Use passive or prefer passive performance SLA.
- Use ISDB/application signatures or ISDB/application signature groups to identify applications.
- Enable passive measurement (`set passive-measurement enable`).

If internet services or applications are defined in an SD-WAN rule with passive or prefer passive performance SLA, SLA information for each service or application will be differentiated and collected. SLA metrics (jitter, latency, and packet loss) on each SD-WAN member in the rule are then calculated based on the relevant internet service's or application's SLA information.

In this example, three SD-WAN rules are created:

- Rule 1: Best quality (latency) using passive SLA for the internet services Alibaba and Amazon.
- Rule 2: Best quality (latency) using passive SLA for the applications Netflix and YouTube.
- Rule 3: Best quality (latency) using passive SLA for all other traffic.

After passive application measurement is enabled for rules one and two, the SLA metric of rule one is the average latency of the internet services Alibaba and Amazon, and the SLA metric of rule two is the average latency of the applications Netflix and YouTube.



**To configure the SD-WAN:**

**1. Configure the SD-WAN members:**

```
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
end

```

## 2. Configure the passive mode health check:

```

config health-check
    edit "Passive_HC"
        set detect-mode passive
        set members 1 2
    next
end

```

## 3. Configure SD-WAN service rules:

```

config service
    edit 1
        set name "1"
        set mode priority
        set src "172.16.205.0"
        set internet-service enable
        set internet-service-name "Alibaba-Web" "Amazon-Web"
        set health-check "Passive_HC"
        set priority-members 1 2
        set passive-measurement enable //Enable "passive application measurement", it
is a new command which is introduced in this project.
    next
    edit 2
        set name "2"
        set mode priority
        set src "172.16.205.0"
        set internet-service enable
        set internet-service-app-ctrl 18155 31077
        set health-check "Passive_HC"
        set priority-members 1 2
        set passive-measurement enable ////Enable "passive application measurement"
    next
    edit 3
        set name "3"
        set mode priority
        set dst "all"
        set src "172.16.205.0"
        set health-check "Passive_HC"
        set priority-members 1 2
    next
end

```

## 4. Configure SD-WAN routes:

```

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

```

## 5. Configure the firewall policy with passive WAN health measurement enabled:

```
config firewall policy
  edit 1
    set uuid 972345c6-1595-51ec-66c5-d705d266f712
    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 passive-wan-health-measurement enable
    set utm-status enable
    set ssl-ssh-profile "certificate-inspection"
    set application-list "g-default"
    set auto-asic-offload disable
  next
end
```

### To verify the results:

1. On the PC, open the browser and visit the internet services and applications.
2. On the FortiGate, check the collected SLA information to confirm that each server or application on the SD-WAN members was measured individually:

```
# diagnose sys link-monitor-passive admin list by-interface

Interface dmz (5):
  Default(0x00000000): latency=3080.0  11:57:54, jitter=5.0      11:58:08,
pktloss=0.0 % NA
  Alibaba-Web(0x00690001): latency=30.0  11:30:06, jitter=25.0  11:29:13,
pktloss=0.0 % NA
  YouTube(0x00007965): latency=100.0  12:00:35, jitter=2.5  12:00:30,
pktloss=0.0 % NA
  Netflix(0x000046eb): latency=10.0  11:31:24, jitter=10.0  11:30:30,
pktloss=0.0 % NA
  Amazon-Web(0x00060001): latency=80.0  11:31:52, jitter=35.0  11:32:07,
pktloss=0.0 % NA

Interface port15 (27):
  Default(0x00000000): latency=100.0  12:00:42, jitter=0.0  12:00:42,
pktloss=0.0 % NA
  Amazon-Web(0x00060001): latency=30.0  11:56:05, jitter=0.0  11:55:21,
pktloss=0.0 % NA
  Alibaba-Web(0x00690001): latency=0.0  11:26:08, jitter=35.0  11:27:08,
pktloss=0.0 % NA
  YouTube(0x00007965): latency=100.0  11:33:34, jitter=0.0  11:33:50,
pktloss=0.0 % NA
  Netflix(0x000046eb): latency=0.0  11:26:29, jitter=0.0  11:29:03,
pktloss=0.0 % NA
```



The Default (0x00000000) applications are other, unidentified applications that do not have ISDB or application signatures configured in SD-WAN rules. The latency of default/application is taken into account in per SD-WAN rule calculations only if passive-measurement is disabled in any one of the SD-WAN rules.

### 3. Verify that the SLA metrics on the members are calculated as expected:

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x600 use-shortcut-sla
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(priority), link-cost-factor
(latency), link-cost-threshold(10), health-check(Passive_HC)
  Members(2):
    1: Seq_num(2 port15), alive, latency: 15.000, selected           // Average latency
of "Alibaba-Web" and "Amazon-Web" on port15:      15.000 = (0.0+30.0)/2
    2: Seq_num(1 dmz), alive, latency: 55.000, selected           // Average latency
of "Alibaba-Web" and "Amazon-Web" on dmz:         55.000 = (30.0+80.0)/2
  Internet Service(2): Alibaba-Web(6881281,0,0,0) Amazon-Web(393217,0,0,0)
  Src address(1):
    172.16.205.0-172.16.205.255

Service(2): Address Mode(IPV4) flags=0x600 use-shortcut-sla
  Gen(2), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(priority), link-cost-factor
(latency), link-cost-threshold(10), health-check(Passive_HC)
  Members(2):
    1: Seq_num(1 dmz), alive, latency: 55.000, selected           // Average latency
of "Netflix" and "YouTube" on dmz:                55.000 = (10.0+100.0)/2
    2: Seq_num(2 port15), alive, latency: 50.000, selected       // Average latency
of "Netflix" and "YouTube" on port15:             50.000 = (0.0+100.0)/2
  Internet Service(2): Netflix(4294837427,0,0,0 18155) YouTube(4294838283,0,0,0 31077)
  Src address(1):
    172.16.205.0-172.16.205.255

Service(3): Address Mode(IPV4) flags=0x200 use-shortcut-sla
  Gen(9), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(priority), link-cost-factor
(latency), link-cost-threshold(10), health-check(Passive_HC)
  Members(2):
    1: Seq_num(2 port15), alive, latency: 46.000, selected       // Average latency
of all TCP traffic on port15:                      46 = (100.0+30.0+0.0+100.0+0.0)/5
    2: Seq_num(1 dmz), alive, latency: 660.000, selected        // Average latency of
all TCP traffic on dmz:                             660 = (3080.0+30.0+100.0+10.0+80.0)/5
  Src address(1):
    172.16.205.0-172.16.205.255

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



The latency on each member interface per SD-WAN rule is the average of the latency of the application identified by respective SD-WAN rules.

The SLA metrics listed for each member interface per SD-WAN rule shown by the `diagnose sys sdwan service4` and `diagnose sys sdwan service6` commands are derived from the output of the SLA information for the applications shown in the output of the `diagnose sys link-monitor-passive admin list by-interface` command.

Until the applications are identified, their SLA metrics are not used to calculate SLA metrics for each member per SD-WAN rule. Applications are identified only when there is (or was) any application traffic passing through a member interface.

## 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). |

### 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
    end
  end
end
```

```

        config sla
            edit 1
                set link-cost-factor mos
                set mos-threshold "4.0"
            next
        end
    next
end
end

```

### To use an MOS SLA to steer traffic in an SD-WAN rule:

```

config system sdwan
    config service
        edit 1
            set name "MOS_traffic_steering"
            set mode sla
            set dst "HQ_LAN"
            set src "Branch_LAN"
            config sla
                edit "Test_MOS"
                    set id 1
                next
            end
            set priority-members 0
        next
    end
end

```



The MOS currently cannot be used to steer traffic when the mode is set to priority.

### 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."
```

## Embedded SD-WAN SLA information in ICMP probes

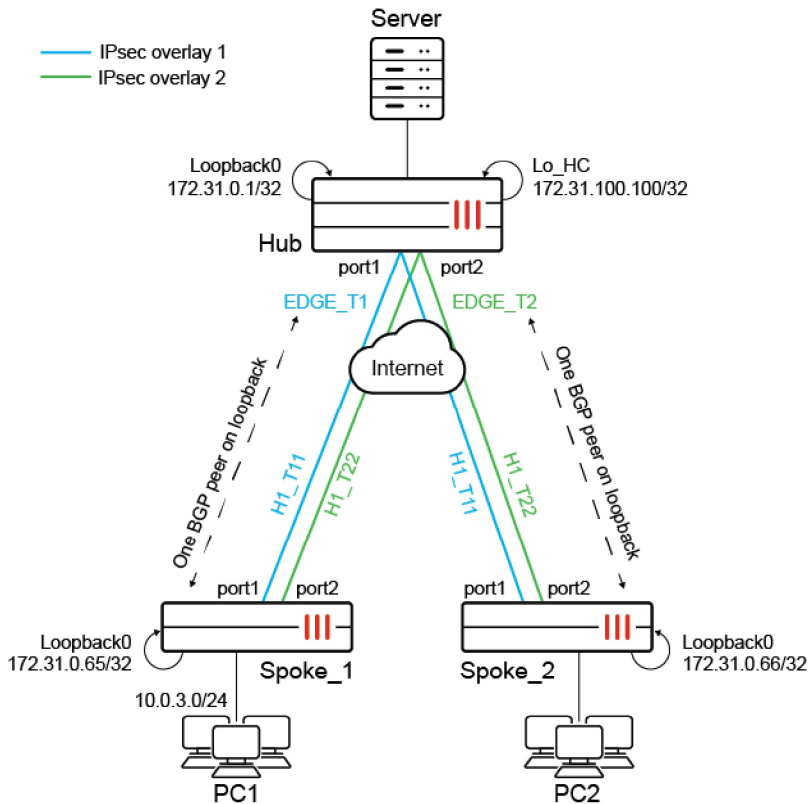
In the hub and spoke SD-WAN design, in order for traffic to pass symmetrically from spoke to hub and hub to spoke, it is essential for the hub to know which IPsec overlay is in SLA and out of SLA. Prior to introducing embedded SLA information in ICMP probes, it is common practice for spokes to use the 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
    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"> <li>• <b>active:</b> the probes are sent actively (default).</li> <li>• <b>passive:</b> the traffic measures health without probes.</li> <li>• <b>prefer-passive:</b> the probes are sent in case of no new traffic.</li> <li>• <b>remote:</b> the link health is obtained from remote peers.</li> </ul> |
| <pre> embed-measured-health   {enable   disable} </pre>                    | <p>Enable/disable embedding SLA information in ICMP probes (default = disable).</p>   |
| <pre> set priority-in-sla   &lt;integer&gt; </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   &lt;integer&gt; </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 &lt;id&gt; </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
        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
    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

```

```
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 service4
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                      |

3. Verify that the spoke has sent its health check result to hub.

a. On the hub, display the status of the health checks for EDGE\_T1 and EDGE\_T2:

```
# diagnose sys sdwan health-check remote
Remote Health Check: 2(2)
  Passive remote statistics of EDGE_T2(22):
EDGE_T2_0(172.31.3.5): timestamp=02-09 16:19:11, latency=1.056, jitter=0.582,
pktloss=0.000%
Remote Health Check: 1(1)
  Passive remote statistics of EDGE_T1(21):
EDGE_T1_0(172.31.3.1): timestamp=02-09 16:19:11, latency=1.269, jitter=0.675,
pktloss=0.000%
```

4. When there are multiple spokes, additional options can be used to filter a spoke by health check name, or health check name and the member's sequence number (`diagnose system sdwan health-check remote <hc_name> <seq_num>`).

a. To filter the health check by health check name:

```
# diagnose sys sdwan health-check remote 1
Remote Health Check: 1(1)
  Passive remote statistics of EDGE_T1(21):
EDGE_T1_0(172.31.3.1): timestamp=02-09 16:43:37, latency=1.114, jitter=0.473,
pktloss=0.000%
```

When this method is used, the output displays all the members of the specified health check name.

b. To filter the health check by health check name and the member's sequence number:

```
# diagnose sys sdwan health-check remote 1 1
Remote Health Check: 1(1)
  Passive remote statistics of EDGE_T1(21):
EDGE_T1_0(172.31.3.1): timestamp=02-09 16:43:41, latency=1.178, jitter=0.497,
pktloss=0.000%
```

When this method is used, the output displays the specified member of the specified health check name.



If the `detect-mode` is set to `remote`, use `diagnose sys sdwan health-check remote` in lieu of `diagnose sys sdwan health-check`.

5. 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 the spoke will be routed to EDGE\_T1.

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 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. These statistics can be used to gain a deeper insight into the SD-WAN traffic performance.

FortiGate can log statistics when using FortiMonitor to detect advanced SD-WAN application performance metrics. These logs may also be sent to FortiAnalyzer and FortiManager for review and reporting.

```
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
  set app-perf-log-period <time in seconds>
end
```

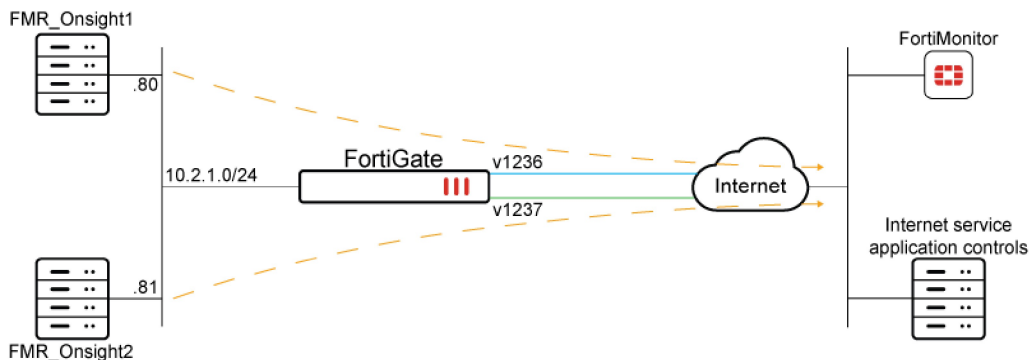
The following diagnostic commands can be used to view agent related metrics:

```
# 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 on page 3239](#) 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. Set the logging frequency:

```
config system sdwan
  set status enable
  set app-perf-log-period 60
end
```

#### 3. Configure the SD-WAN zone and members:

```
config system sdwan
  config zone
    edit "virtual-wan-link"
    next
  end
  config members
    edit 1
      set interface "v1236"
      set gateway 10.12.36.2
    next
    edit 2
      set interface "v1237"
      set gateway 10.12.37.20
    next
  end
end
```

#### 4. 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

```

#### 5. 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

```

#### 5. Review the SD-WAN logs:

6. # execute log filter category event  
# execute log filter field subtype sdwan  
# execute log display

```
1: date=2023-01-27 time=16:32:15 eventtime=1674865935918381398 tz="-0800"  
logid="0113022937" type="event" subtype="sdwan" level="information" vd="root"  
logdesc="Virtuan WAN Link application performance metrics via FortiMonitor"  
eventtype="Application Performance Metrics" app="fortinet.com" appid=0 interface="v1237"  
latency="200.2" jitter="0.6" packetloss="0.0" serverresponsetime="827.7"  
networktransfertime="107.7" apperror="0.0" timestamp="01-28 00:31:59" msg="Application  
Performance Metrics via FortiMonitor"
```

```
2: date=2023-01-27 time=16:32:15 eventtime=1674865935918367770 tz="-0800"  
logid="0113022937" type="event" subtype="sdwan" level="information" vd="root"  
logdesc="Virtuan WAN Link application performance metrics via FortiMonitor"  
eventtype="Application Performance Metrics" app="fortinet.com" appid=0 interface="v1236"  
latency="200.0" jitter="0.3" packetloss="0.0" serverresponsetime="870.6"  
networktransfertime="130.4" apperror="0.0" timestamp="01-28 00:31:59" msg="Application  
Performance Metrics via FortiMonitor"
```

```
3: date=2023-01-27 time=16:31:15 eventtime=1674865875917685437 tz="-0800"  
logid="0113022937" type="event" subtype="sdwan" level="information" vd="root"  
logdesc="Virtuan WAN Link application performance metrics via FortiMonitor"  
eventtype="Application Performance Metrics" app="fortinet.com" appid=0 interface="v1237"  
latency="200.5" jitter="0.7" packetloss="0.0" serverresponsetime="1008.9"  
networktransfertime="129.8" apperror="0.0" timestamp="01-28 00:31:02" msg="Application  
Performance Metrics via FortiMonitor"
```

```
4: date=2023-01-27 time=16:31:15 eventtime=1674865875917672824 tz="-0800"  
logid="0113022937" type="event" subtype="sdwan" level="information" vd="root"  
logdesc="Virtuan WAN Link application performance metrics via FortiMonitor"  
eventtype="Application Performance Metrics" app="fortinet.com" appid=0 interface="v1236"  
latency="200.3" jitter="0.8" packetloss="0.0" serverresponsetime="825.4"  
networktransfertime="106.4" apperror="0.0" timestamp="01-28 00:31:02" msg="Application  
Performance Metrics via FortiMonitor"
```

```
5: date=2023-01-27 time=16:30:15 eventtime=1674865815912801725 tz="-0800"  
logid="0113022937" type="event" subtype="sdwan" level="information" vd="root"  
logdesc="Virtuan WAN Link application performance metrics via FortiMonitor"  
eventtype="Application Performance Metrics" app="fortinet.com" appid=0 interface="v1237"  
latency="200.1" jitter="0.4" packetloss="0.0" serverresponsetime="845.4"  
networktransfertime="116.0" apperror="0.0" timestamp="01-28 00:30:01" msg="Application  
Performance Metrics via FortiMonitor"
```

```
6: date=2023-01-27 time=16:30:15 eventtime=1674865815912786458 tz="-0800"  
logid="0113022937" type="event" subtype="sdwan" level="information" vd="root"  
logdesc="Virtuan WAN Link application performance metrics via FortiMonitor"  
eventtype="Application Performance Metrics" app="fortinet.com" appid=0 interface="v1236"  
latency="200.0" jitter="0.3" packetloss="0.0" serverresponsetime="1032.0"  
networktransfertime="138.9" apperror="0.0" timestamp="01-28 00:30:01" msg="Application  
Performance Metrics via FortiMonitor"
```

## Classifying SLA probes for traffic prioritization

Traffic classification on SLA probes helps to ensure that they are prioritized in times of congestion. This prevents SD-WAN link flapping and unexpected routing behaviors, and stabilizes SD-WAN from unnecessary failovers.

SLA probes can be classified into a specific class ID so that SLA probes assigned to a class ID with higher priority are prioritized over other traffic. SLA probes are assigned using the `class-id` command:

```
config system sdwan
  config health-check
    edit <health-check name>
      set class-id <class name>
    next
  end
end
```



For more information on traffic shaping, see [Traffic shaping on page 1511](#).

### Example

In this example, SLA probes are assigned into different class ID. The interfaces `dmz` and `vd1-01` both have outbandwidth of 1000000 Kbps (1 Gbps) configured. Three traffic shaping classes are defined:

| Class ID | Name        | Definition  |
|----------|-------------|---|
| 2        | sla_probe   | High priority with a guaranteed 10% of bandwidth (100 Mbps)   |
| 3        | default     | Low priority with a guaranteed 80% of bandwidth (800 Mbps)    |
| 4        | sla_probe_2 | Medium priority with a guaranteed 10% of bandwidth (100 Mbps) |

Under this scheme, when congestion occurs, traffic in each class will have their guaranteed bandwidth honored. If there is remaining bandwidth, higher priority traffic will get the bandwidth. On the SD-WAN health check, probes to server 2.2.2.2 are assigned to class 2 (`sla_probe`). This means it has a guaranteed bandwidth and has the highest priority to use unused bandwidth. This allows SD-WAN health check to function properly even during times of congestion.

#### To classify SLA probes for traffic prioritization:

1. Configure the firewall traffic class:

```
config firewall traffic-class
  edit 2
    set class-name "sla_probe"
  next
  edit 3
    set class-name "default"
  next
```

```
    edit 4
      set class-name "sla_probe_2"
    next
  end
```

## 2. Configure the class ID priority and guaranteed bandwidth:

```
config firewall shaping-profile
  edit "profile-1"
    set default-class-id 3
    config shaping-entries
      edit 2
        set class-id 2
        set priority high
        set guaranteed-bandwidth-percentage 10
        set maximum-bandwidth-percentage 100
      next
      edit 3
        set class-id 3
        set priority low
        set guaranteed-bandwidth-percentage 80
        set maximum-bandwidth-percentage 100
      next
      edit 4
        set class-id 4
        set priority medium
        set guaranteed-bandwidth-percentage 10
        set maximum-bandwidth-percentage 100
      next
    end
  next
end
```

## 3. Configure the interfaces:

```
config system interface
  edit "dmz"
    set outbandwidth 1000000
    set egress-shaping-profile "profile-1"
    ...
  next
  edit "vd1-p1"
    set outbandwidth 1000000
    set egress-shaping-profile "profile-1"
    ...
  next
end
```

## 4. Configure the SD-WAN health check and assign the SLA probes into class 2:

```
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 "vd1-p1"
    next
end
config health-check
    edit "1"
        set server "2.2.2.2"
        set members 1 2
        set class-id 2
        config sla
            edit 1
                next
            end
        next
    end
end
end
end

```

### To verify the SLA probe assignment:

#### 1. Verify the health check diagnostics:

```

# diagnose sys sdwan health-check
Health Check(1):
Seq(1 dmz): state(alive), packet-loss(0.000%) latency(0.247), jitter(0.022), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
Seq(2 vd1-p1): state(alive), packet-loss(0.000%) latency(0.247), jitter(0.018), mos
(4.404), bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_map=0x1

```

#### 2. Verify the SLA probes are assigned into class 2:

```

# diagnose netlink interface list dmz
if=dmz family=00 type=1 index=5 mtu=1500 link=0 master=0
ref=36 state=start present fw_flags=10018000 flags=up broadcast run multicast
Qdisc=mq hw_addr=e0:23:ff:9d:f9:9e broadcast_addr=ff:ff:ff:ff:ff:ff
egress traffic control:
    bandwidth=1000000 (kbps) lock_hit=0 default_class=3 n_active_class=3
    class-id=3      allocated-bandwidth=800000 (kbps)      guaranteed-
bandwidth=800000 (kbps)
                                max-bandwidth=1000000 (kbps)      current-bandwidth=1 (kbps)
                                priority=low      forwarded_bytes=1446
                                dropped_packets=0      dropped_bytes=0
    class-id=4      allocated-bandwidth=100000 (kbps)      guaranteed-
bandwidth=100000 (kbps)
                                max-bandwidth=1000000 (kbps)      current-bandwidth=0 (kbps)
                                priority=medium      forwarded_bytes=0
                                dropped_packets=0      dropped_bytes=0
    class-id=2      allocated-bandwidth=100000 (kbps)      guaranteed-
bandwidth=100000 (kbps)
                                max-bandwidth=1000000 (kbps)      current-bandwidth=1 (kbps)
                                priority=high      forwarded_bytes=1404
                                dropped_packets=0      dropped_bytes=0
stat: rxp=19502 txp=14844 rxb=2233923 txb=802522 rx=0 tx=0 rxd=0 txd=0 mc=0
collision=0 @ time=1675121675
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0

```

```

te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=36

# diagnose netlink interface list vd1-p1
if=vd1-p1 family=00 type=768 index=99 mtu=1420 link=0 master=0
ref=20 state=start present fw_flags=10010000 flags=up p2p run noarp multicast
Qdisc=noqueue
egress traffic control:
    bandwidth=1000000 (kbps) lock_hit=0 default_class=3 n_active_class=3
    class-id=3    allocated-bandwidth=800000 (kbps)    guaranteed-
bandwidth=800000 (kbps)
                                max-bandwidth=1000000 (kbps)    current-bandwidth=0 (kbps)
                                priority=low    forwarded_bytes=0
                                dropped_packets=0    dropped_bytes=0
    class-id=4    allocated-bandwidth=100000 (kbps)    guaranteed-
bandwidth=100000 (kbps)
                                max-bandwidth=1000000 (kbps)    current-bandwidth=0 (kbps)
                                priority=medium    forwarded_bytes=0
                                dropped_packets=0    dropped_bytes=0
    class-id=2    allocated-bandwidth=100000 (kbps)    guaranteed-
bandwidth=100000 (kbps)
                                max-bandwidth=1000000 (kbps)    current-bandwidth=1 (kbps)
                                priority=high    forwarded_bytes=1120
                                dropped_packets=0    dropped_bytes=0
stat: rxp=4097 txp=4586 rxb=540622 txb=221500 rx=0 txe=19 rxd=0 txd=0 mc=0
collision=0 @ time=1675121742
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=20

```



When verifying the class assignment, the counter value should increase.

The example also demonstrates assigning SLA probes to class 4 (sla\_probe\_2), in which case the probes get medium priority.

### To assign the SLA probe to medium priority:

#### 1. Assign SLA probes into class 4:

```

config sys sdwan
    config health-check
        edit 1
            set class-id 4
        next
    end
    set status disable
end
config sys sdwan
    set status enable
end

```

#### 2. Verify the SLA probes are assigned into class 4.



```

# diagnose netlink interface list dmz
if=dmz family=00 type=1 index=5 mtu=1500 link=0 master=0
ref=34 state=start present fw_flags=10018000 flags=up broadcast run multicast
Qdisc=mq hw_addr=e0:23:ff:9d:f9:9e broadcast_addr=ff:ff:ff:ff:ff:ff
egress traffic control:
    bandwidth=1000000 (kbps) lock_hit=0 default_class=3 n_active_class=3
    class-id=3    allocated-bandwidth=800000 (kbps)    guaranteed-
bandwidth=800000 (kbps)
                    max-bandwidth=1000000 (kbps)    current-bandwidth=1 (kbps)
                    priority=low    forwarded_bytes=24K
                    dropped_packets=0    dropped_bytes=0
    class-id=4    allocated-bandwidth=100000 (kbps)    guaranteed-
bandwidth=100000 (kbps)
                    max-bandwidth=1000000 (kbps)    current-bandwidth=1 (kbps)
                    priority=medium    forwarded_bytes=1674
                    dropped_packets=0    dropped_bytes=0
    class-id=2    allocated-bandwidth=100000 (kbps)    guaranteed-
bandwidth=100000 (kbps)
                    max-bandwidth=1000000 (kbps)    current-bandwidth=0 (kbps)
                    priority=high    forwarded_bytes=0
                    dropped_packets=0    dropped_bytes=0
stat: rxp=20818 txp=15874 rxb=2382789 txb=857674 rx=0 tx=0 rxd=0 txd=0 mc=0
collision=0 @ time=1675122057
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=34

# diagnose netlink interface list vd1-p1
if=vd1-p1 family=00 type=768 index=99 mtu=1420 link=0 master=0
ref=20 state=start present fw_flags=10010000 flags=up p2p run noarp multicast
Qdisc=noqueue
egress traffic control:
    bandwidth=1000000 (kbps) lock_hit=0 default_class=3 n_active_class=3
    class-id=3    allocated-bandwidth=800000 (kbps)    guaranteed-
bandwidth=800000 (kbps)
                    max-bandwidth=1000000 (kbps)    current-bandwidth=0 (kbps)
                    priority=low    forwarded_bytes=0
                    dropped_packets=0    dropped_bytes=0
    class-id=4    allocated-bandwidth=100000 (kbps)    guaranteed-
bandwidth=100000 (kbps)
                    max-bandwidth=1000000 (kbps)    current-bandwidth=1 (kbps)
                    priority=medium    forwarded_bytes=1280
                    dropped_packets=0    dropped_bytes=0
    class-id=2    allocated-bandwidth=100000 (kbps)    guaranteed-
bandwidth=100000 (kbps)
                    max-bandwidth=1000000 (kbps)    current-bandwidth=0 (kbps)
                    priority=high    forwarded_bytes=0
                    dropped_packets=0    dropped_bytes=0
stat: rxp=4097 txp=4703 rxb=540622 txb=226180 rx=0 tx=19 rxd=0 txd=0 mc=0 collision=0
@ time=1675122058
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=20

```

## SD-WAN rules

SD-WAN rules, which are sometimes called *service rules*, identify traffic of interest, and then route the traffic based on a strategy and the condition of the route or *link* between two devices. You can use many strategies to select the outgoing interface and many performance service level agreements (SLAs) to evaluate the link conditions.

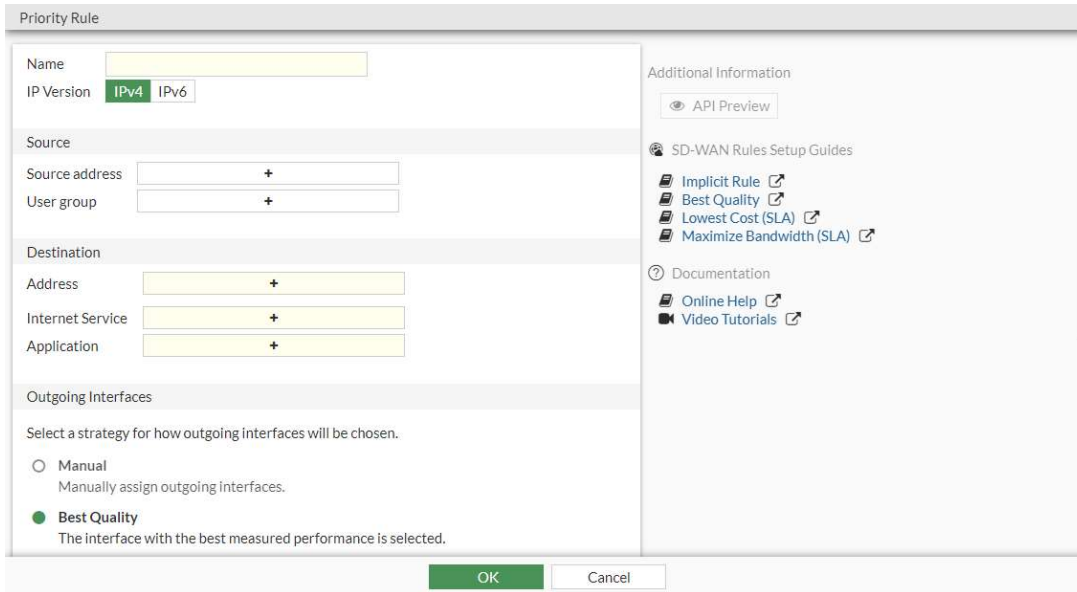
Use the following topics to learn about and create SD-WAN rules for your needs:

- [SD-WAN rules overview on page 851](#)
- [Implicit rule on page 859](#)
- [Automatic strategy on page 863](#)
- [Manual strategy on page 864](#)
- [Best quality strategy on page 867](#)
- [Lowest cost \(SLA\) strategy on page 871](#)
- [Load balancing strategy on page 877](#)
- [SD-WAN traffic shaping and QoS on page 878](#)
- [SDN dynamic connector addresses in SD-WAN rules on page 883](#)
- [Application steering using SD-WAN rules on page 885](#)
- [DSCP tag-based traffic steering in SD-WAN on page 898](#)
- [ECMP support for the longest match in SD-WAN rule matching on page 905](#)
- [Override quality comparisons in SD-WAN longest match rule matching on page 907](#)
- [Internet service and application control steering on page 910](#)
- [Use maximize bandwidth to load balance traffic between ADVPN shortcuts on page 919](#)
- [Use SD-WAN rules to steer multicast traffic on page 926](#)
- [Use SD-WAN rules for WAN link selection with load balancing on page 940](#)

## SD-WAN rules overview

SD-WAN rules control how sessions are distributed to SD-WAN members. You can configure SD-WAN rules from the GUI and CLI.

From the GUI, go to *Network > SD-WAN > SD-WAN Rules*. When creating a new SD-WAN rule, or editing an existing SD-WAN rule, use the *Source* and *Destination* sections to identify traffic, and use the *Outgoing interfaces* section to configure WAN intelligence for routing traffic.



From the CLI, use the following command to configure SD-WAN rules:

```
config system sdwan
    config service
        edit <ID>
            next
        end
    end
end
```

The following topics describe the fields used to configure SD-WAN rules:

- [Fields for identifying traffic on page 852](#)
- [Fields for configuring WAN intelligence on page 856](#)
- [Additional fields for configuring WAN intelligence on page 858](#)

### Fields for identifying traffic

This topic describes the fields in an SD-WAN rule used for defining the traffic to which the rule applies. Some fields are available only in the CLI.

SD-WAN rules can identify traffic by a variety of means:

| Address type                  | Source | Destination |
|-------------------------------|--------|-------------|
| IPv4/6                        | ✓      | ✓           |
| MAC                           | ✓      | ✓           |
| Group                         | ✓      | ✓           |
| FABRIC_DEVICE dynamic address | ✓      | ✓           |
| Users                         | ✓      | ✓           |
| User groups                   | ✓      | ✓           |

| Address type                                    | Source | Destination |
|---|--------|-------------|
| Application control (application aware routing) |        | ✓           |
| Internet service database (ISDB)                |        | ✓           |
| BGP route tags                                  |        | ✓           |
| Differentiated Services Code Point (DSCP) tags  |        | ✓           |

In the GUI, go to *Network > SD-WAN > SD-WAN Rules*. Click *Create New*, or double-click an existing rule to open it for editing. The *Source* and *Destination* sections are used to identify traffic for the rule:

In the CLI, edit the service definition ID number to identify traffic for the rule:

```
config system sdwan
  config service
    edit <ID>
      <CLI commands from the following tables>
      ...
    end
  end
end
```

The following table describes the fields used for the name, ID, and IP version of the SD-WAN rule:

| ID, Name, and IP version |  |   |
|--------------------------|--|---|
| Field                    | CLI  | Description   |
| ID                       | <pre>config system sdwan   config service     edit &lt;ID&gt;       next     end   end end</pre> | ID is generated when the rule is created. You can only specify the ID from the CLI. |

| ID, Name, and IP version |  |  |
|--------------------------|--|--|
| Field                    | CLI  | Description  |
| Name                     | <code>set name &lt;string&gt;</code>           | The name does not need to relate to the traffic being matched, but it is good practice to have intuitive rule names.                       |
| IP version               | <code>set addr-mode &lt;ipv4   ipv6&gt;</code> | The addressing mode can be IPv4 or IPv6.<br>To configure in the GUI, IPv6 must be enabled from <i>System &gt; Feature Visibility</i> page. |

The following table describes the fields used for source section of the SD-WAN rule:

| Source           |  |  |
|------------------|--|--|
| Field            | CLI  | Description  |
| Source address   | <code>set src &lt;object&gt;</code><br><code>set start-src-port &lt;integer&gt;</code><br><code>set end-src-port &lt;integer&gt;</code><br><b>Use <code>set src-negate enable</code> to negate the address object.</b> | One or more address objects.<br>Start source port number. CLI only.<br>End source port number. CLI only. |
| User group       | <code>set users &lt;user object&gt;</code><br><code>set groups &lt;group object&gt;</code>   | Individual users or user groups  |
| Source interface | <code>set input-device &lt;interface name&gt;</code><br><b>Can be negated with <code>set input-device-negate enable</code>.</b>  | Select one or more source interfaces. CLI only.  |

The following table describes the fields used for the destination section of the SD-WAN rule:

| Destination      |   |  |
|------------------|---|--|
| Field            | CLI   | Description  |
| Address          | <code>set dst &lt;object&gt;</code><br><code>set protocol &lt;integer&gt;</code><br><code>set start-port &lt;integer&gt;</code><br><code>set end-port &lt;integer&gt;</code><br><b>Use <code>set dst-negate enable</code> to negate the address object.</b> | One or more address objects.<br>One protocol and one port range can be combined with the address object.<br>If it is necessary for an SD-WAN rule to match multiple protocols or multiple port ranges, you can create a custom Internet Service. |
| Internet Service | <code>set internet-service enable</code><br><code>set internet-service-custom &lt;name_1&gt; &lt;name_2&gt; ... &lt;name_n&gt;</code><br><code>set internet-service-custom-group &lt;name_1&gt; &lt;name_2&gt; ... &lt;name_n&gt;</code>                    | One or more internet services or service groups.   |

| Destination                          |  |   |
|--------------------------------------|--|---|
| Field                                | CLI  | Description   |
|                                      | <pre>set internet-service-name   &lt;name_1&gt; &lt;name_2&gt; ...   &lt;name_n&gt; set internet-service-group   &lt;name_1&gt; &lt;name_2&gt; ...   &lt;name_n&gt;</pre>  |   |
| Application                          | <pre>set internet-service-app-ctrl   &lt;id_1&gt; &lt;id_2&gt; ... &lt;id_n&gt; set internet-service-app-ctrl-   group &lt;name_1&gt; &lt;name_2&gt;   ... &lt;name_n&gt; set internet-service-app-ctrl-   category &lt;id_1&gt; &lt;id_2&gt;   ... &lt;id_n&gt;</pre> | <p>One or more applications or application groups.</p> <p>Can be used with internet services or service group.</p>  |
| Route tag ( <code>route-tag</code> ) | <pre>set route-tag &lt;integer&gt;</pre>   | <p>CLI only.</p> <p>This replaces the <code>dst</code> field (if previously configured) and matches a BGP route tag configured in a route map. See <a href="#">Using BGP tags with SD-WAN rules on page 952</a>.</p>  |
| TOS mask ( <code>tos-mask</code> )   | <pre>set tos-mask &lt;8-bit hex value&gt;</pre>  | <p>CLI only.</p> <p>In order to leverage type of service (TOS) matching or DSCP matching on the IP header, the SD-WAN rule must specify the bit mask of the byte holding the TOS value. For example, a TOS mask of 0xe0 (11100000) matches the upper 3 bits.</p>  |
| TOS ( <code>tos</code> )             | <pre>set tos &lt;8 bit hex value&gt;</pre>   | <p>CLI only.</p> <p>The value specified here is matched after the <code>tos-mask</code> is applied.</p> <p>For example, the FortiGate receives DSCP values 110000 and 111011. (DSCP is the upper 6 bits of the TOS field – 11000000 and 11101100 respectively). Using the TOS value 0xe0 (11100000), only the second DSCP value is matched.</p> |

By default, individual applications and application groups cannot be selected in SD-WAN rules. To enable this functionality in the GUI, go to *System > Feature Visibility* and enable *Application Detection Based SD-WAN*. In the CLI, enter:

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

## Fields for configuring WAN intelligence

This topic describes the fields in an SD-WAN rule used for configuring WAN intelligence, which processes and routes traffic that matches the SD-WAN rule.

In the GUI, go to *Network > SD-WAN > SD-WAN Rules*. Click *Create New*, or double-click an existing rule to open it for editing. The *Outgoing Interfaces* section is used to configure WAN intelligence for the rule:

Priority Rule

Settings Info

Outgoing Interfaces

Interface selection strategy

- Manual  
Manually assign outgoing interfaces.
- Best quality  
The interface with the best measured performance is selected.
- Lowest cost (SLA)**  
The interface that meets SLA targets is selected. When there is a tie, the interface with the lowest assigned cost is selected.

Interface preference  +

Zone preference  +

Measured SLA  ▾

Required SLA target  +

Load balancing

Quality criteria  ▾

Forward DSCP

Reverse DSCP

OK Cancel

WAN intelligence is comprised of the following parts:

- [Interface or zone preference on page 856](#)
- [Strategy on page 857](#)
- [Performance SLA on page 857](#)

### Interface or zone preference

By default, the configured order of interfaces and/or zones in a rule are used. Interfaces and zones that are selected first have precedence over interfaces selected second and so on.

You can specify both interfaces and zones. When a zone is specified in the *Zone preference* field, it is equivalent to selecting each of the contained interface members in the *Interface preference* section. Interface members in a zone have lower priority than interfaces configured in the *Interface preference* section.

For example:

- There are 3 interfaces: port1, port2 and port3.
  - Port2 is in Zone1
  - Port1 and port3 belong to the default *virtual-wan-link* zone.
- An SD-WAN rule is created with *Interface preference* set to *port3* and *port1*, and *Zone preference* set to *Zone1*.

Interface preference

- ×
- ×
- +

Zone preference

- ×
- +

The SD-WAN rule prefers the interfaces in the following order:

1. port3
2. port1
3. port2

You can configure the interface and zone preference in the CLI:

```
config system sdwan
  config service
    edit <ID>
      set priority-members <integer>
      set priority-zone <interface>
    next
  end
end
```

## Strategy

Strategy dictates how the interface and/or zone order changes as link conditions change. You can use the following strategies:

- Automatic (`auto`): interfaces are assigned a priority based on quality. See [Automatic strategy on page 863](#).
- Manual (`manual`): interfaces are manually assigned a priority. See [Manual strategy on page 864](#).
- Best Quality (`priority`): interfaces are assigned a priority based on the `link-cost-factor` of the interface. See [Best quality strategy on page 867](#).
- Lowest cost (SLA) (`sla`): interfaces are assigned a priority based on selected SLA settings. See [Lowest cost \(SLA\) strategy on page 871](#).

## Performance SLA

The best quality, lowest cost, and maximize bandwidth strategies are the most intelligent modes, and they leverage SLA health checks to provide meaningful metrics for a given link. FortiGate uses the metrics to make intelligent decisions to route traffic.

Automatic and manual strategies have pre-configured logic that do not leverage SLA health checks.

The goal of the performance SLA is to measure the quality of each SD-WAN member link. The following methods can be used to measure the quality of a link:

- Active measurement
  - Health-check traffic is sent to a server with a variety of protocols options.
  - The following SLA metrics are measured on this probe traffic:
    - Latency
    - Jitter
    - Packet loss
- Passive measurement
  - SLA metrics are measured on real or live traffic, reducing the amount of probe traffic that is sent and received.
  - There is the option (`prefer passive`) to initiate probe traffic when no live traffic is present.

Performance SLA is utilized by `auto`, *Lowest Cost (SLA)*, *Maximize Bandwidth (SLA)*, and *Best Quality* strategies. *Lowest Cost (SLA)* and *Maximize Bandwidth SLA* use SLA targets in a pass or fail style to evaluate whether a link is considered for traffic. *Best Quality* compares a specific metric of the SLA to pick the best result.



Therefore it is integral to select or create an SLA target(s) that relates to the traffic targeted by the rule. It does not make sense to evaluate a public resource, such as YouTube, when the rule matches Azure traffic.

See [Performance SLA on page 808](#) for more details.

## Additional fields for configuring WAN intelligence

This topic describes the fields in an SD-WAN rule used for configuring WAN intelligence for egress traffic:

- [Forward and/or reverse differentiated services code point \(DSCP\) on page 858](#)
- [Default and gateway options on page 858](#)

For information about accessing fields for configuring WAN intelligence, see [Fields for configuring WAN intelligence on page 856](#).

### Forward and/or reverse differentiated services code point (DSCP)

The FortiGate differentiated services feature can be used to change the DSCP value for all packets accepted by a policy.

The packet's DSCP field for traffic initiating a session (forward) or for reply traffic (reverse) can be changed and enabled in each direction separately by configuring it in the firewall policy using the *Forward DSCP* and *Reverse DSCP* fields.

From the CLI:

```
config system sdwan
  config service
    edit <ID>
      ...
      set dscp-forward enable
      ...
    next
  end
end
```

|                                |                                 |
|--------------------------------|---------------------------------|
| set dscp-forward enable        | Enable use of forward DSCP tag. |
| set dscp-forward-tag<br>000000 | Forward traffic DSCP tag.       |
| set dscp-reverse enable        | Enable use of reverse DSCP tag. |
| set dscp-reverse-tag<br>000000 | Reverse traffic DSCP tag.       |

### Default and gateway options

Following are additional gateway options that can be set only in the CLI:

```
config system sdwan
  config service
    edit <ID>
      ...
      set default enable
      ...
    next
  end
end
```

|   |   |
|---|---|
| <code>set default</code><br><code>[enable disable]</code> | Enable or disable use of SD-WAN as default service. |
| <code>set gateway</code><br><code>[enable disable]</code> | Enable or disable SD-WAN service gateway.           |

By default, these settings are set to `disable`.

These two commands help adjust FortiGate route selection by affecting how the FortiGate consults the Forward Information Base (FIB).

In order to decide whether an SD-WAN policy-route can be matched, FortiGate performs the following FIB lookups:

- FIB best match for the destination must return an SD-WAN member.
- FIB route to the destination must exist over the desired SD-WAN member.

When `set default enable` is used with `set gateway enable`, FortiGate bypasses the FIB checks, and instead routes any matching traffic of the SD-WAN rule to the chosen SD-WAN member using the member's configured gateway. SD-WAN members must have a gateway configured.

When `set default disable` is used with `set gateway enable`, FortiGate keeps the first rule in effect but causes the second rule to change to:

- FIB route to the gateway IP address must exist over any interface.

See also [Fields for configuring WAN intelligence on page 856](#).

## Implicit rule

SD-WAN rules define specific policy routing options to route traffic to an SD-WAN member. When no explicit SD-WAN rules are defined, or if none of the rules are matched, then the default implicit rule is used.

In an SD-WAN configuration, the default route usually points to the SD-WAN interface, so each active member's gateway is added to the routing table's default route. FortiOS uses equal-cost multipath (ECMP) to balance traffic between the interfaces. One of five load balancing algorithms can be selected:

|  |  |
|--|--|
| Source IP ( <code>source-ip-based</code> ) | Traffic is divided equally between the interfaces, including the SD-WAN interface. Sessions that start at the same source IP address use the same path. This is the default selection.   |
| Sessions ( <code>weight-based</code> )     | The workload is distributing based on the number of sessions that are connected through the interface.<br>The weight that you assign to each interface is used to calculate the percentage of the total sessions that are allowed to connect through an interface, and the sessions are distributed to the interfaces accordingly.<br>The sessions with the same source and destination IP are forwarded to the same path if the device model and kernel version supports route cache. However, it is not guaranteed and the route cache could be refreshed in case network events take place. In most cases where route cache is not supported, the sessions with the same source and destination IP will be load balanced between SD-WAN member interfaces.<br>An interface's weight value cannot be zero. |

|   |  |
|---|--|
| Spillover ( <i>usage-based</i> )                      | The interface is used until the traffic bandwidth exceeds the ingress and egress thresholds that you set for that interface. Additional traffic is then sent through the next SD-WAN interface member.   |
| Source-Destination IP ( <i>source-dest-ip-based</i> ) | Traffic is divided equally between the interfaces. Sessions that start at the same source IP address and go to the same destination IP address use the same path.  |
| Volume ( <i>measured-volume-based</i> )               | The workload is distributing based on the number of packets that are going through the interface.<br><br>The volume weight that you assign to each interface is used to calculate the percentage of the total bandwidth that is allowed to go through an interface, and the bandwidth is distributed to the interfaces accordingly.<br><br>An interface's volume value cannot be zero. |

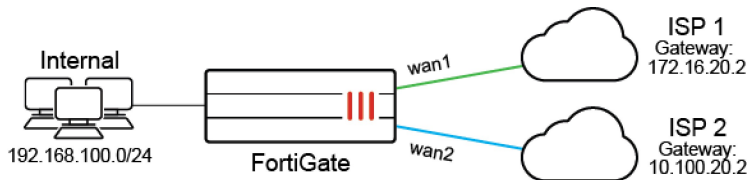


You cannot exclude an interface from participating in load balancing using the implicit rule. If the weight or volume were set to zero in a previous FortiOS version, the value is treated as a one.

When using dynamic routes for routing, sessions are distributed equally regardless of weight. Interfaces with static routes can be excluded from ECMP if they are configured with a lower priority than other static routes.

## Examples

The following four examples demonstrate how to use the implicit rules (load-balance mode).



If no SD-WAN zone is specified, members are added to the default *virtual-wan-link* zone.

### Example 1

Outgoing traffic is equally balanced between wan1 and wan2, using *source-ip-based* or *source-dest-ip-based* mode.

#### Using the GUI:

1. On the FortiGate, enable SD-WAN and add wan1 and wan2 as SD-WAN members, then add a policy and static route. See [SD-WAN quick start on page 786](#) for details.
2. Go to *Network > SD-WAN* and select the *SD-WAN Rules* tab.
3. Edit the *sd-wan* rule (the last default rule).
4. For the *Load Balancing Algorithm*, select either *Source IP* or *Source-Destination IP*.
5. Click *OK*.

## Using the CLI:

1. Enable SD-WAN and add wan1 and wan2 as SD-WAN members, then add a policy and static route. See [SD-WAN quick start on page 786](#) for details.
2. Set the load balancing algorithm:  
Source IP based:

```
config system sdwan
    set load-balance-mode source-ip-based
end
```

Source-Destination IP based:

```
config system sdwan
    set load-balance-mode source-dest-ip-based
end
```

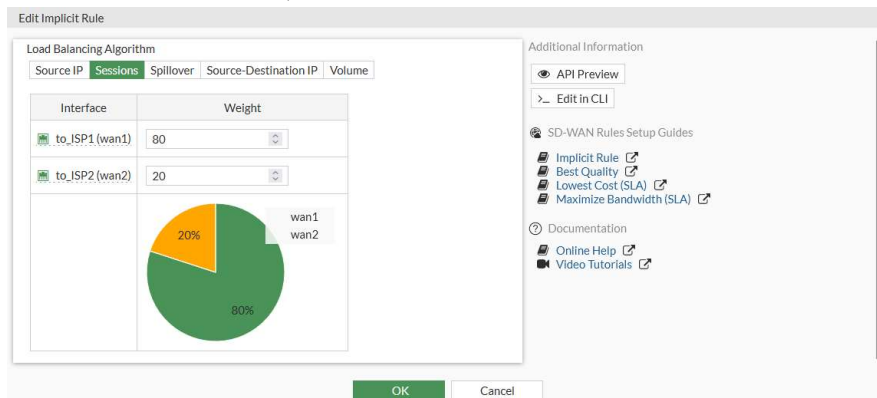
## Example 2

Outgoing traffic is balanced between wan1 and wan2 with a customized ratio, using *weight-based* mode: wan1 runs 80% of the sessions, and wan2 runs 20% of the sessions.

Sessions with the same source and destination IP addresses (`src-ip` and `dst-ip`) will be forwarded to the same path, but will still be considered in later session ratio calculations.

## Using the GUI:

1. Go to *Network > SD-WAN* and select the *SD-WAN Rules* tab.
2. Edit the *sd-wan* rule (the last default rule).
3. For the *Load Balancing Algorithm*, select *Sessions*.
4. Enter 80 in the *wan1* field, and 20 in the *wan2* field.



5. Click *OK*.

## Using the CLI:

```
config system sdwan
    set load-balance-mode weight-based
config members
    edit 1
        set interface "wan1"
        set weight 80
```

```
        next
    edit 2
        set interface "wan2"
        set weight 20
    next
end
end
```

### Example 3

Outgoing traffic is balanced between wan1 and wan2 with a customized ratio, using *measured-volume-based* mode: wan1 runs 80% of the volume, and wan2 runs 20% of the volume.

#### Using the GUI:

1. Go to *Network > SD-WAN* and select the *SD-WAN Rules* tab.
2. Edit the *sd-wan* rule (the last default rule).
3. For the *Load Balancing Algorithm*, select *Volume*.
4. Enter 80 in the *wan1* field, and 20 in the *wan2* field.
5. Click *OK*.

#### Using the CLI:

```
config system sdwan
    set load-balance-mode measured-volume-based
    config members
        edit 1
            set interface "wan1"
            set volume-ratio 80
        next
        edit 2
            set interface "wan2"
            set volume-ratio 20
        next
    end
end
```

### Example 4

Load balancing can be used to reduce costs when internet connections are charged at different rates. For example, if wan2 charges based on volume usage and wan1 charges a fixed monthly fee, we can use wan1 at its maximum bandwidth, and use wan2 for overflow.

In this example, wan1's bandwidth is 10Mbps down and 2Mbps up. Traffic will use wan1 until it reaches its spillover limit, then it will start to use wan2. Note that *auto-asic-offload* must be disabled in the firewall policy.

#### Using the GUI:

1. On the FortiGate, enable SD-WAN and add wan1 and wan2 as SD-WAN members, then add a policy and static route. See [SD-WAN quick start on page 786](#) for details.
2. Go to *Network > SD-WAN* and select the *SD-WAN Rules* tab.
3. Edit the *sd-wan* rule (the last default rule).
4. For the *Load Balancing Algorithm*, select *Spillover*.

5. Enter 10000 in the *wan1 Ingress Spillover Threshold* field, and 2000 in the *wan1 Egress Spillover Threshold* field.

| Interface      | Ingress Spillover Threshold | Egress Spillover Threshold |
|----------------|-----------------------------|----------------------------|
| to_ISP1 (wan1) | 10000 kbps                  | 2000 kbps                  |
| to_ISP2 (wan2) | 0 kbps                      | 0 kbps                     |

6. Click *OK*.

### Using the CLI:

```
config system sdwan
  set load-balance-mode usage-based
  config members
    edit 1
      set interface "wan1"
      set spillover-threshold 2000
      set ingress-spillover-threshold 10000
    next
  end
end
```

## Automatic strategy

The automatic strategy is a legacy rule that lets you select an outgoing interface based on its performance ranking compared to the other SD-WAN interfaces. This is achieved by applying a performance SLA to rank the interfaces, and then selecting the desired rank.

In this example, you have three SD-WAN interfaces to three different ISPs that all go to the public internet. WAN1 is your highest quality link and should be reserved for business critical traffic. WAN2 and WAN3 are redundant backup links. You noticed one non-critical application is taking up a lot of bandwidth and want to prioritize it to the lowest quality link at any given time.

### To configure automatic SD-WAN rules from the CLI:

```
config system sdwan
  config members
    edit 1
      set interface "wan1"
    next
    edit 2
      set interface "wan2"
    next
    edit 3
      set interface "wan3"
    next
  end
end
```

```

end
config health-check
  edit "non-critical application"
    set server "noncritical.application.com"
    set members 1 2 3
    config sla
      edit 1
        set latency-threshold 250
        set jitter-threshold 50
        set packetloss-threshold 3
      next
    end
  next
end
config service
  edit 1
    set name "non-critical application"
    set mode auto
    set quality-link 3
    set dst "non-critical-app-address-object"
    set health-check "non-critical application"
  next
end
end

```



The `auto` option is only available in the CLI. If you use the GUI to edit the rule, the `auto` option will be overwritten because you cannot select `auto` in the GUI.

## Manual strategy

In manual mode, no health checks are used. As a result, the decision making closer resembles logic than intelligence. SD-WAN manual rules are similar to regular policy-based routes, but have the added features of application-aware routing and BGP-tag routing. A manual strategy rule is comprised of the following parts:

- Defining the interfaces to be used
- Ordering the interfaces based on preference, or load balancing traffic out of the specified interfaces using a load balancing algorithm



The maximize bandwidth (`load-balance`) strategy used prior to FortiOS 7.4.1 is now known as the load balancing strategy. This strategy can be configured under the manual mode and the lowest cost (SLA) strategies.

- When the load balancing strategy is configured under the manual mode strategy, SLA targets are not used.
- When the load balancing strategy is configured under the lowest cost (SLA) strategy, SLA targets are used.

### To configure manual SD-WAN rules from the GUI:

1. Go to *Network > SD-WAN*.
2. Select the *SD-WAN Rules* tab, and click *Create New*.

3. Set the following options to create a manual rule:

|                        |   |
|------------------------|---|
| <b>Name</b>            | Type a name for the rule.   |
| <b>Source</b>          | (Optional) Specify a <i>Source address</i> and/or <i>User group</i> .   |
| <b>Destination</b>     | Specify the destination using an <i>Address</i> object or an <i>Internet Service</i> or an <i>Application</i> .   |
| <b>Zone preference</b> | Specify one or more SD-WAN interfaces or zones.<br>The order in which the interfaces or zones are specified determines their priority when the rule is matched. |

4. Set the remaining options as desired, and click *OK* to create the rule.

### To configure manual SD-WAN rules from the CLI:

```
config system sdwan
  config members
    edit 1
      set interface "wan1"
    next
    edit 2
      set interface "wan2"
    next
  end
  config service
    edit 1
      set name "manual"
      set mode manual
      set priority-members 2 1
      set dst "DC_net"
      set hold-down-time 60
    next
  end
end
```



- The command `set mode manual` will not appear in the configuration because it is the default mode.
- The command `set hold-down-time <integer>` is an optional command that controls how long to wait before switching back to the primary interface in the event of a failover.

## Load balancing strategy without SLA targets

The load balancing strategy known as maximize bandwidth (`load-balance`) prior FortiOS 7.4.1 is now configured within manual mode SD-WAN rules to achieve load balancing but without the need to configure SLA targets.

By enabling load balancing mode (`set load-balance enable`) inside the manual SD-WAN rule, SD-WAN will start to load balance traffic out of all the specified interfaces based on the configured load balancing algorithm. There is no explicit need to configure SLA targets to achieve load balancing. The load balancing algorithm, or hash method, can be one of the following:



|                                   |   |
|-----------------------------------|---|
| <code>round-robin</code>          | All traffic is distributed to selected interfaces in equal portions and circular order. This is the default method, and the only option available when using the GUI. |
| <code>source-ip-based</code>      | All traffic from a source IP is sent to the same interface.   |
| <code>source-dest-ip-based</code> | All traffic from a source IP to a destination IP is sent to the same interface.   |
| <code>inbandwidth</code>          | All traffic is distributed to a selected interface with most available bandwidth for incoming traffic.  |
| <code>outbandwidth</code>         | All traffic is distributed to a selected interface with most available bandwidth for outgoing traffic.  |
| <code>bibandwidth</code>          | All traffic is distributed to a selected interface with most available bandwidth for both incoming and outgoing traffic.  |

When the `inbandwidth`, `outbandwidth`, or `bibandwidth` load balancing algorithm is used, the FortiGate will compare the bandwidth based on the configured upstream and downstream bandwidth values.

The interface speedtest can be used to populate the bandwidth values based on the speedtest results. See [GUI speed test on page 1147](#) for details.

#### To manually configure the upstream and downstream bandwidth values:

```
config system interface
  edit <interface>
    set estimated-upstream-bandwidth <speed in kbps>
    set estimated-downstream-bandwidth <speed in kbps>
  next
end
```

#### To enable the load balancing strategy for manual mode in the GUI:

1. Go to *Network > SD-WAN*.
2. Select the *SD-WAN Rules* tab, and click *Create New*.
3. Set the *Interface selection strategy* to *Manual*.
4. Enable *Load balancing*.

Priority Rule

Settings Info

Source

Address

User group

Destination

Address

Protocol number

Internet service

Outgoing Interfaces

Interface selection strategy

**Manual**  
Manually assign outgoing interfaces.

**Best quality**  
The interface with the best measured performance is selected.

**Lowest cost (SLA)**  
The interface that meets SLA targets is selected. When there is a tie, the interface with the lowest assigned cost is selected.

Interface preference

to\_ISP2 (wan2)

to\_ISP1 (wan1)

Zone preference

Measured SLA

Required SLA target

Load balancing

5. Set the remaining options as desired, and click *OK* to create the rule.

### To enable the load balancing strategy for manual mode in the CLI:

```
config system sdwan
...
config service
edit 1
set name "manual"
set mode manual
set load-balance enable
set hash-mode round-robin
set priority-members 2 1
set dst "DC_net"
set hold-down-time 60
next
end
end
```

## Best quality strategy

When using *Best Quality* mode, SD-WAN will choose the best link to forward traffic by comparing the *link-cost-factor*. A link-cost factor is a specific metric of participating link(s) (such as, latency, packet loss, and so on) evaluated against a target that you define (such as a health-check server), for example, the latency of WAN1 and WAN2 to your datacenter. Below is a list of link-cost factors available to you:

| GUI                | CLI              | Description   |
|--------------------|------------------|---|
| Latency            | latency          | Select a link based on latency.   |
| Jitter             | jitter           | Select a link based on jitter.  |
| Packet Loss        | packet-loss      | Select a link based on packet loss.   |
| Downstream         | inbandwidth      | Select a link based on available bandwidth of incoming traffic.   |
| Upstream           | outbandwidth     | Select a link based on available bandwidth of outgoing traffic.   |
| Bandwidth          | bibandwidth      | Select a link based on available bandwidth of bidirectional traffic.  |
| Customized profile | custom-profile-1 | Select link based on customized profile. If selected, set the following weights: <ul style="list-style-type: none"> <li>packet-loss-weight: Coefficient of packet-loss.</li> <li>latency-weight: Coefficient of latency.</li> <li>jitter-weight: Coefficient of jitter.</li> <li>bandwidth-weight: Coefficient of reciprocal of available bidirectional bandwidth.</li> </ul> |

Although SD-WAN intelligence selects the best quality link according to the selected metric, by default a preference or advantage is given to the first configured SD-WAN member. This default is 10% and may be configured with the CLI command `set link-cost-threshold 10`.

Example of how `link-cost-threshold` works:

```
config system sdwan
  config members
    edit 1
      set interface "wan1"
    next
    edit 2
      set interface "wan2"
    next
  end
  config service
    edit 1
      set name "Best_Quality"
      set mode priority
      set priority-members 2 1
      set dst "DC_net"
      set health-check "DC_HealthCheck"
      set link-cost-factor latency
      set link-cost-threshold 10
    next
  end
end
```

In this example both WAN1 and WAN2 are assumed to have 200ms latency to the health-check server named `DC_HealthCheck`. Because WAN2 is specified before WAN1 in `priority-members`, SD-WAN parses the two interfaces metric as follows:

- WAN1: 200ms
- WAN2:  $200\text{ms} / (1+10\%) = \sim 182\text{ms}$

As a result, WAN2 is selected because the latency is lower.

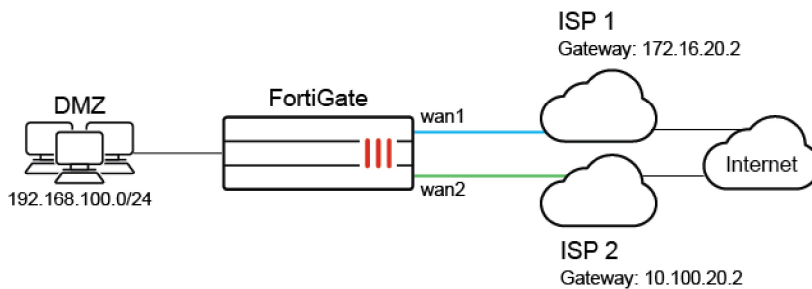
If the *Downstream* (`inbandwidth`), *Upstream* (`outbandwidth`), or *Bandwidth* (`bibandwidth`) quality criteria is used, the FortiGate uses the upstream and downstream bandwidth values configured on the member interfaces to calculate bandwidth.

The interface bandwidth configuration can be done manually, or the interface speedtest can be used to populate the bandwidth values based on the speedtest results. See [GUI speed test on page 1147](#) for details.

### To manually configure the upstream and downstream interface bandwidth values:

```
config system interface
  edit <interface>
    set estimated-upstream-bandwidth <speed in kbps>
    set estimated-downstream-bandwidth <speed in kbps>
  next
end
```

### Example



In this example, your wan1 and wan2 SD-WAN interfaces connect to two ISPs that both go to the public internet, and you want Gmail services to use the link with the least latency.

### To configure an SD-WAN rule to use Best Quality:

1. On the FortiGate, add wan1 and wan2 as SD-WAN members, then add a policy and static route. See [SD-WAN quick start on page 786](#) for more details.
2. Go to *Network > SD-WAN*, select the *Performance SLAs* tab, and click *Create New*.
3. Enter a name for the performance SLA, such as *google*, and set the *Server* to *google.com*. See [Health checks](#) for more details.
4. Click *OK*.
5. Go to *Network > SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
6. Enter a name for the rule, such as *gmail*.
7. Configure the following settings:

Priority Rule

Name

Source

Source address  +

User group  +

Destination

Address  +

Internet Service  x

Application  +

Outgoing Interfaces

Select a strategy for how outgoing interfaces will be chosen.

Manual  
Manually assign outgoing interfaces.

**Best Quality**  
The interface with the best measured performance is selected.

Lowest Cost (SLA)  
The interface that meets SLA targets is selected. When there is a tie, the interface with the lowest assigned cost is selected.

Maximize Bandwidth (SLA)  
Traffic is load balanced among interfaces that meet SLA targets.

Interface preference

to\_ISP1 (wan1) x

to\_ISP2 (wan2) x

+ +

Zone preference  +

Measured SLA

Quality criteria

Forward DSCP

Reverse DSCP

Status  Enable  Disable

Additional Information

API Preview

SD-WAN Rules Setup Guides

- [Implicit Rule](#)
- [Best Quality](#)
- [Lowest Cost \(SLA\)](#)
- [Maximize Bandwidth \(SLA\)](#)

Documentation

- [Online Help](#)
- [Video Tutorials](#)

OK Cancel

|                         |              |
|-------------------------|--------------|
| <b>Internet Service</b> | Google-Gmail |
|-------------------------|--------------|

|                 |              |
|-----------------|--------------|
| <b>Strategy</b> | Best Quality |
|-----------------|--------------|

|                             |               |
|-----------------------------|---------------|
| <b>Interface preference</b> | wan1 and wan2 |
|-----------------------------|---------------|

|                     |        |
|---------------------|--------|
| <b>Measured SLA</b> | google |
|---------------------|--------|

|                         |         |
|-------------------------|---------|
| <b>Quality criteria</b> | Latency |
|-------------------------|---------|

8. Click *OK*.

### To configure an SD-WAN rule to use priority:

```
config system sdwan
  config health-check
    edit "google"
      set server "google.com"
      set members 1 2
    next
  end
  config service
    edit 1
      set name "gmail"
      set mode priority
      set internet-service enable
      set internet-service-id 65646
      set health-check "google"
```

```

        set link-cost-factor latency
        set priority-members 1 2
    next
end
end

```

### To diagnose the Performance SLA status:

```
FGT # diagnose sys sdwan health-check google
```

```
Health Check(google):
```

```
Seq(1): state(alive), packet-loss(0.000%) latency(14.563), jitter(4.334) sla_map=0x0
```

```
Seq(2): state(alive), packet-loss(0.000%) latency(12.633), jitter(6.265) sla_map=0x0
```

```
FGT # diagnose sys sdwan service4 1
```

```
Service(1):
```

```
TOS(0x0/0x0), protocol(0: 1->65535), Mode(priority), link-cost-facotr(latency), link-
cost-threshold(10), health-check(google) Members:
```

```
1: Seq_num(2), alive, latency: 12.633, selected
```

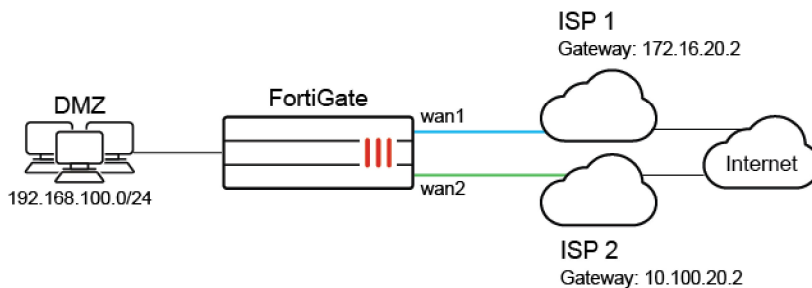
```
2: Seq_num(1), alive, latency: 14.563, selected
```

```
Internet Service: Google-Gmail(65646)
```

As wan2 has a smaller latency, SD-WAN will put Seq\_num(2) on top of Seq\_num(1) and wan2 will be used to forward Gmail traffic.

## Lowest cost (SLA) strategy

When using *Lowest Cost (SLA)* mode (`sla` in the CLI), SD-WAN will choose the lowest cost link that satisfies SLA to forward traffic. The lowest possible cost is 0. If multiple eligible links have the same cost, the *Interface preference* order will be used to select a link.



In this example, your wan1 and wan2 SD-WAN interfaces connect to two ISPs that both go to the public internet. The cost of wan2 is less than that of wan1. You want to configure Gmail services to use the lowest cost interface, but the link quality must meet a standard of latency: 10ms, and jitter: 5ms.

### To configure an SD-WAN rule to use Lowest Cost (SLA):

1. On the FortiGate, add wan1 and wan2 as SD-WAN members, then add a policy and static route. See [SD-WAN quick start on page 786](#) for details.
2. Go to *Network > SD-WAN*, select the *Performance SLAs* tab, and click *Create New*.
3. Enter a name for the performance SLA, such as *google*, and set the *Server* to *google.com*.

4. Enable *SLA Target*. Set the *Latency threshold* to 10 ms, and the *Jitter threshold* to 5 ms. See [Health checks](#) for more details.
5. Click *OK*.
6. Go to *Network > SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
7. Enter a name for the rule, such as *gmail*.
8. Configure the following settings:

|                         |              |
|-------------------------|--------------|
| <b>Internet Service</b> | Google-Gmail |
|-------------------------|--------------|

|                 |                   |
|-----------------|-------------------|
| <b>Strategy</b> | Lowest Cost (SLA) |
|-----------------|-------------------|

|                             |               |
|-----------------------------|---------------|
| <b>Interface preference</b> | wan1 and wan2 |
|-----------------------------|---------------|

|                            |        |
|----------------------------|--------|
| <b>Required SLA target</b> | google |
|----------------------------|--------|

9. Click *OK*.

#### To configure an SD-WAN rule to use SLA:

```
config system sdwan
  config members
    edit 1
      set interface "wan1"
```

```

        set cost 10
    next
    edit 2
        set interface "wan2"
        set cost 5
    next
end
config health-check
    edit "google"
        set server "google.com"
        set members 1 2
        config sla
            edit 1
                set latency-threshold 10
                set jitter-threshold 5
            next
        end
    next
end
config service
    edit 1
        set name "gmail"
        set mode sla
        set internet-service enable
        set internet-service-id 65646
        config sla
            edit "google"
                set id 1
            next
        end
        set priority-members 1 2
    next
end
end

```



If no SD-WAN zone is specified, members are added to the default *virtual-wan-link* zone.



The CLI command `set minimum-sla-meet-members` allows you to specify the number of links that must meet SLA for the rule to take effect. If the number of members is less than the minimum set with this command, the rule will not take effect.

### To diagnose the performance SLA status:

```
FGT # diagnose sys sdwan health-check status
```

```
Health Check(google):
```

```
Seq(1): state(alive), packet-loss(0.000%) latency(14.563), jitter(4.334) sla_map=0x0
```

```
Seq(2): state(alive), packet-loss(0.000%) latency(12.633), jitter(6.265) sla_map=0x0
```

```
FGT # diagnose sys sdwan service4 1
```

```
Service(1): Address Mode(IPV4) flags=0x0
```



```
TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla)
Members:<<BR>>

    1: Seq_num(2), alive, sla(0x1), cfg_order(1), selected
    2: Seq_num(1), alive, sla(0x1), cfg_order(0), selected

Internet Service: Google.Gmail(65646)
```

When both wan1 and wan2 meet the SLA requirements, Gmail traffic will only use wan2. If only wan1 meets the SLA requirements, Gmail traffic will only use wan1, even though it has a higher cost. If neither interface meets the requirements, wan2 will be used.

If both interface had the same cost and both met the SLA requirements, the first link configured in `set priority-members` would be used.

### Load balancing strategy with SLA targets

SD-WAN rules can be configured to load balance traffic out of all the interfaces that satisfy the SLA target.

The load balancing strategy known as maximize bandwidth (`load-balance`) prior FortiOS 7.4.1 is now configured within *Lowest Cost (SLA)* mode (`sla`) SD-WAN rules.

By enabling load balancing mode (`set load-balance enable`) inside the lowest cost SD-WAN rule, SD-WAN will choose all of the links that satisfy the SLA target to forward traffic based on a load balancing algorithm. The load balancing algorithm, or hash method, can be one of the following:

|                                   |   |
|-----------------------------------|---|
| <code>round-robin</code>          | All traffic is distributed to selected interfaces in equal portions and circular order. This is the default method, and the only option available when using the GUI. |
| <code>source-ip-based</code>      | All traffic from a source IP is sent to the same interface.   |
| <code>source-dest-ip-based</code> | All traffic from a source IP to a destination IP is sent to the same interface.   |
| <code>inbandwidth</code>          | All traffic is distributed to a selected interface with most available bandwidth for incoming traffic.  |
| <code>outbandwidth</code>         | All traffic is distributed to a selected interface with most available bandwidth for outgoing traffic.  |
| <code>bibandwidth</code>          | All traffic is distributed to a selected interface with most available bandwidth for both incoming and outgoing traffic.  |

When the `inbandwidth`, `outbandwidth`, or `bibandwidth` load balancing algorithm is used, the FortiGate will compare the bandwidth based on the configured upstream and downstream bandwidth values.

The interface speedtest can be used to populate the bandwidth values based on the speedtest results. See [GUI speed test on page 1147](#) for details.

#### To manually configure the upstream and downstream bandwidth values:

```
config system interface
    edit <interface>
        set estimated-upstream-bandwidth <speed in kbps>
        set estimated-downstream-bandwidth <speed in kbps>
```

```

next
end

```

## Example

Based on the same topology as the preceding example, your wan1 and wan2 SD-WAN interfaces connect to two ISPs that both go to the public internet. You want to configure Gmail services to use both of the interface, but the link quality must meet a standard of latency: 10ms, and jitter: 5ms. This can maximize the bandwidth usage by using load balancing.

### To configure an SD-WAN rule to use load balancing with SLA targets in the GUI:

1. On the FortiGate, add wan1 and wan2 as SD-WAN members, then add a policy and static route. See [SD-WAN quick start on page 786](#) for details.
2. Go to *Network > SD-WAN*, select the *Performance SLAs* tab, and click *Create New*.
3. Enter a name for the performance SLA, such as *google*, and set the *Server* to *google.com*.
4. Enable *SLA Target*. Set the *Latency threshold* to *10 ms*, and the *Jitter threshold* to *5 ms*. See [Health checks](#) for more details.
5. Click *OK*.
6. Go to *Network > SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
7. Enter a name for the rule, such as *gmail*.
8. Configure the following settings:

The screenshot shows the configuration page for a Priority Rule in the FortiGate GUI. The 'Outgoing Interfaces' section is expanded, showing three options: 'Manual', 'Best quality', and 'Lowest cost (SLA)'. The 'Lowest cost (SLA)' option is selected, with a description: 'The interface that meets SLA targets is selected. When there is a tie, the interface with the lowest assigned cost is selected.' Below this, the 'Interface preference' list contains two entries: 'to\_ISP1 (wan1)' and 'to\_ISP2 (wan2)'. The 'Required SLA target' is set to 'google'. The 'Load balancing' checkbox is checked, and the 'Quality criteria' dropdown is set to 'Latency'. The 'Forward DSCP' and 'Reverse DSCP' checkboxes are unchecked. At the bottom, there are 'OK' and 'Cancel' buttons.

|                             |                     |
|-----------------------------|---------------------|
| <b>Internet Service</b>     | Google-Gmail        |
| <b>Strategy</b>             | Lowest Cost (SLA)   |
| <b>Interface preference</b> | wan1 and wan2       |
| <b>Required SLA target</b>  | google              |
| <b>Load balancing</b>       | Enable this setting |

9. Click *OK*.

### To configure an SD-WAN rule to use load balancing with SLA targets in the CLI:

```

config system sdwan
  config members
    edit 1
      set interface "wan1"
      set cost 10
    next
    edit 2
      set interface "wan2"
      set cost 5
    next
  end
  config health-check
    edit "google"
      set server "google.com"
      set members 1 2
      config sla
        edit 1
          set latency-threshold 10
          set jitter-threshold 5
        next
      end
    next
  end
  config service
    edit 1
      set name "gmail"
      set load-balance enable
      set mode sla
      set internet-service enable
      set internet-service-name "Google-Gmail"
      config sla
        edit "google"
          set id 1
        next
      end
      set priority-members 1 2
    next
  end
end

```

**To diagnose the performance SLA status:**

```

FGT # diagnose sys sdwan health-check status
Health Check(google):
Seq(1): state(alive), packet-loss(0.000%) latency(14.563), jitter(4.334) sla_map=0x0
Seq(2): state(alive), packet-loss(0.000%) latency(12.633), jitter(6.265) sla_map=0x0

FGT # diagnose sys sdwan service4 1
Service(1): Address Mode(IPV4) flags=0x0

TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance)
Members:<<BR>>

1: Seq_num(1), alive, sla(0x1), num of pass(1), selected
2: Seq_num(2), alive, sla(0x1), num of pass(1), selected

Internet Service: Google.Gmail(65646)

```

When both wan1 and wan2 meet the SLA requirements, Gmail traffic will use both wan1 and wan2. If only one of the interfaces meets the SLA requirements, Gmail traffic will only use that interface.

If neither interface meets the requirements but the `health-check` is still alive, then wan1 and wan2 tie. The traffic will try to balance between wan1 and wan2, using both interfaces to forward traffic.



The maximize bandwidth (`load-balance`) strategy used prior to FortiOS 7.4.1 is now known as the load balancing strategy. This strategy can be configured under the manual mode and the lowest cost (SLA) strategies.

- When the load balancing strategy is configured under the manual mode strategy, SLA targets are not used.
- When the load balancing strategy is configured under the lowest cost (SLA) strategy, SLA targets are used.

The load balancing strategy functionality remains the same as the maximum bandwidth (SLA) strategy when it is configured inside the lowest cost (SLA) strategy: load balance traffic out of all the interfaces that satisfy the SLA targets. Interface cost is not considered when selecting the best path when the load balancing strategy is used.

## Load balancing strategy

The maximize bandwidth (`load-balance`) strategy used prior to FortiOS 7.4.1 is now known as the load balancing strategy. This strategy can be configured under the manual mode and the lowest cost (SLA) strategies.

- When the load balancing strategy is configured under the manual mode strategy, SLA targets are not used (see [Load balancing strategy without SLA targets](#) for an example configuration).
- When the load balancing strategy is configured under the lowest cost (SLA) strategy, SLA targets are used (see [Load balancing strategy with SLA targets](#) for an example configuration).



The load balancing strategy functionality remains the same as the maximum bandwidth (SLA) strategy when it is configured inside the lowest cost (SLA) strategy: load balance traffic out of all the interfaces that satisfy the SLA targets. Interface cost is not considered when selecting the best path when the load balancing strategy is used.

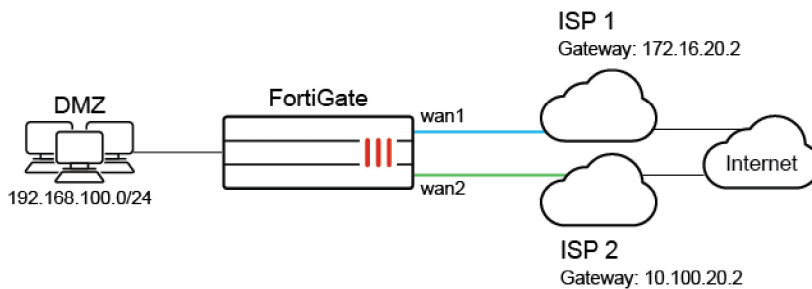
## SD-WAN traffic shaping and QoS

Use a traffic shaper in a firewall shaping policy to control traffic flow. You can use it to control maximum and guaranteed bandwidth, or put certain traffic to one of the three different traffic priorities: high, medium, or low.

An advanced shaping policy can classify traffic into 30 groups. Use a shaping profile to define the percentage of the interface bandwidth that is allocated to each group. Each group of traffic is shaped to the assigned speed limit based on the outgoing bandwidth limit configured on the interface.

For more information, see [Traffic shaping on page 1511](#).

### Sample topology



### Sample configuration

This example shows a typical customer usage where the customer's SD-WAN uses the default zone, and has two member: wan1 and wan2, each set to 10Mb/s.

An overview of the procedures to configure SD-WAN traffic shaping and QoS with SD-WAN includes:

1. Give HTTP/HTTPS traffic high priority and give FTP low priority so that if there are conflicts, FortiGate will forward HTTP/HTTPS traffic first.
2. Even though FTP has low priority, configure FortiGate to give it a 1Mb/s guaranteed bandwidth on each SD-WAN member so that if there is no FTP traffic, other traffic can use all the bandwidth. If there is heavy FTP traffic, it can still be guaranteed a 1Mb/s bandwidth.
3. Traffic going to specific destinations such as a VOIP server uses wan1 to forward, and SD-WAN forwards with an Expedited Forwarding (EF) DSCP tag 101110.

#### To configure SD-WAN traffic shaping and QoS with SD-WAN in the GUI:

1. On the FortiGate, add wan1 and wan2 as SD-WAN members, then add a policy and static route.  
See [SD-WAN quick start on page 786](#).
2. Add a firewall policy with *Application Control* enabled. See [Configuring firewall policies for SD-WAN on page 789](#).
3. Go to *Policy & Objects > Traffic Shaping*, select the *Traffic Shapers* tab, and edit *low-priority*.
  - a. Enable *Guaranteed Bandwidth* and set it to 1000 kbps.
4. Go to *Policy & Objects > Traffic Shaping*, select the *Traffic Shaping Policies* tab, and click *Create New*.
  - a. Name the traffic shaping policy, for example, *HTTP-HTTPS*.
  - b. Set the following:

|               |            |
|---------------|------------|
| <b>Source</b> | <i>all</i> |
|---------------|------------|

|                           |  |
|---------------------------|--|
| <b>Destination</b>        | <i>all</i>                             |
| <b>Service</b>            | <i>HTTP and HTTPS</i>                  |
| <b>Outgoing interface</b> | <i>virtual-wan-link</i>                |
| <b>Shared Shaper</b>      | Enable and set to <i>high-priority</i> |
| <b>Reverse Shaper</b>     | Enable and set to <i>high-priority</i> |

- c. Click OK.
5. Go to *Policy & Objects > Traffic Shaping*, select the *Traffic Shaping Policies* tab, and click *Create New*.
  - a. Name the traffic shaping policy, for example, *FTP*.
  - b. Set the following:

|                           |                                       |
|---------------------------|---------------------------------------|
| <b>Source</b>             | <i>all</i>                            |
| <b>Destination</b>        | <i>all</i>                            |
| <b>Service</b>            | <i>FTP, FTP_GET, and FTP_PUT</i>      |
| <b>Outgoing interface</b> | <i>virtual-wan-link</i>               |
| <b>Shared Shaper</b>      | Enable and set to <i>low-priority</i> |
| <b>Reverse Shaper</b>     | Enable and set to <i>low-priority</i> |

- c. Click OK
6. Go to *Network > SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
  - a. Enter a name for the rule, such as *Internet*.
  - b. In the *Destination* section, click *Address* and select the VoIP server that you created in the firewall address.
  - c. Under *Outgoing Interfaces* select *Manual*.
  - d. For *Interface preference* select *wan1*.
  - e. Click OK.
7. Use CLI commands to modify DSCP settings. See the DSCP CLI commands below.

### To configure the firewall policy using the CLI:

```

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

```

**To configure the firewall traffic shaper priority using the CLI:**

```
config firewall shaper traffic-shaper
  edit "high-priority"
    set maximum-bandwidth 1048576
    set per-policy enable
  next
  edit "low-priority"
    set guaranteed-bandwidth 1000
    set maximum-bandwidth 1048576
    set priority low
    set per-policy enable
  next
end
```

**To configure the firewall traffic shaping policy using the CLI:**

```
config firewall shaping-policy
  edit 1
    set name "http-https"
    set service "HTTP" "HTTPS"
    set dstintf "virtual-wan-link"
    set traffic-shaper "high-priority"
    set traffic-shaper-reverse "high-priority"
    set srcaddr "all"
    set dstaddr "all"
  next
  edit 2
    set name "FTP"
    set service "FTP" "FTP_GET" "FTP_PUT"
    set dstintf "virtual-wan-link"
    set traffic-shaper "low-priority"
    set traffic-shaper-reverse "low-priority"
    set srcaddr "all"
    set dstaddr "all"
  next
end
```

**To configure SD-WAN traffic shaping and QoS with SD-WAN in the CLI:**

```
config system sdwan
  set status enable
  config members
    edit 1
      set interface "wan1"
      set gateway 172.16.20.2
    next
    edit 2
      set interface "wan2"
      set gateway 10.100.20.2
    next
  end
  config service
    edit 1
      set name "SIP"
      set priority-members 1
```

```

        set dst "voip-server"
        set dscp-forward enable
        set dscp-forward-tag 101110
    next
end
end

```



If no SD-WAN zone is specified, members are added to the default *virtual-wan-link* zone.

### To use the diagnose command to check if specific traffic is attached to the correct traffic shaper:

```

# diagnose firewall iprope list 100015

policy index=1 uuid_idx=0 action=accept
flag (0):
shapers: orig=high-priority(2/0/134217728) reply=high-priority(2/0/134217728)
cos_fwd=0 cos_rev=0
group=00100015 av=00000000 au=00000000 split=00000000
host=0 chk_client_info=0x0 app_list=0 ips_view=0
misc=0 dd_type=0 dd_mode=0
zone(1): 0 -> zone(2): 36 38
source(1): 0.0.0.0-255.255.255.255, uuid_idx=6,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=6,
service(2):
    [6:0x0:0/(1,65535)->(80,80)] helper:auto
    [6:0x0:0/(1,65535)->(443,443)] helper:auto

policy index=2 uuid_idx=0 action=accept
flag (0):
shapers: orig=low-priority(4/128000/134217728) reply=low-priority(4/128000/134217728)
cos_fwd=0 cos_rev=0
group=00100015 av=00000000 au=00000000 split=00000000
host=0 chk_client_info=0x0 app_list=0 ips_view=0
misc=0 dd_type=0 dd_mode=0
zone(1): 0 -> zone(2): 36 38
source(1): 0.0.0.0-255.255.255.255, uuid_idx=6,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=6,
service(3):
    [6:0x0:0/(1,65535)->(21,21)] helper:auto
    [6:0x0:0/(1,65535)->(21,21)] helper:auto
    [6:0x0:0/(1,65535)->(21,21)] helper:auto

```

### To use the diagnose command to check if the correct traffic shaper is applied to the session:

```

# diagnose sys session list
session info: proto=6 proto_state=01 duration=11 expire=3599 timeout=3600 flags=00000000
sockflag=00000000 sockport=0 av_idx=0 use=5
origin-shaper=low-priority prio=4 guarantee 128000Bps max 1280000Bps traffic 1050Bps drops
0B
reply-shaper=
per_ip_shaper=
class_id=0 shaping_policy_id=2 ha_id=0 policy_dir=0 tunnel=/ helper=ftp vlan_cos=0/255

```



```

state=may_dirty npu npd os mif route_preserve
statistic(bytes/packets/allow_err): org=868/15/1 reply=752/10/1 tuples=2
tx speed(Bps/kbps): 76/0 rx speed(Bps/kbps): 66/0
origin->sink: org pre->post, reply pre->post dev=39->38/38->39 gwy=172.16.200.55/0.0.0.0
hook=post dir=org act=snat 10.1.100.11:58241->172.16.200.55:21(172.16.200.1:58241)
hook=pre dir=reply act=dnat 172.16.200.55:21->172.16.200.1:58241(10.1.100.11:58241)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=4
serial=0003255f tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id = 00000000
dd_type=0 dd_mode=0
npu_state=0x100000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0,
vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason: offload-denied helper
total session 1

```

### To use the diagnose command to check the status of a shared traffic shaper:

```
# diagnose firewall shaper traffic-shaper list
```

```

name high-priority
maximum-bandwidth 131072 KB/sec
guaranteed-bandwidth 0 KB/sec
current-bandwidth 0 B/sec
priority 2
tos ff
packets dropped 0
bytes dropped 0

```

```

name low-priority
maximum-bandwidth 131072 KB/sec
guaranteed-bandwidth 125 KB/sec
current-bandwidth 0 B/sec
priority 4
tos ff
packets dropped 0
bytes dropped 0

```

```

name high-priority
maximum-bandwidth 131072 KB/sec
guaranteed-bandwidth 0 KB/sec
current-bandwidth 0 B/sec
priority 2
policy 1
tos ff
packets dropped 0
bytes dropped 0

```

```

name low-priority
maximum-bandwidth 131072 KB/sec
guaranteed-bandwidth 125 KB/sec
current-bandwidth 0 B/sec
priority 4

```

```

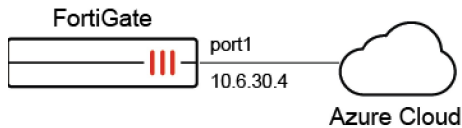
policy 2
tos ff
packets dropped 0
bytes dropped 0

```

## SDN dynamic connector addresses in SD-WAN rules

SDN dynamic connector addresses can be used in SD-WAN rules. FortiGate supports both public (AWS, Azure, GCP, OCI, AliCloud) and private (Kubernetes, VMware ESXi and NSX, OpenStack, ACI, Nuage) SDN connectors.

The configuration procedure for all of the supported SDN connector types is the same. This example uses an Azure public SDN connector.



There are four steps to create and use an SDN connector address in an SD-WAN rule:

1. Configure the FortiGate IP address and network gateway so that it can reach the Internet.
2. [Create an Azure SDN connector.](#)
3. [Create a firewall address to associate with the configured SDN connector.](#)
4. [Use the firewall address in an SD-WAN service rule.](#)

### To create an Azure SDN connector:

1. Go to *Security Fabric > External Connectors*.
2. Click *Create New*.
3. In the *Public SDN* section, click *Microsoft Azure*.
4. Enter the following:

|                        |                                      |
|------------------------|--------------------------------------|
| <b>Name</b>            | azure1                               |
| <b>Status</b>          | Enabled                              |
| <b>Update Interval</b> | Use Default                          |
| <b>Server region</b>   | Global                               |
| <b>Directory ID</b>    | 942b80cd-1b14-42a1-8dcf-4b21dece61ba |
| <b>Application ID</b>  | 14dbd5c5-307e-4ea4-8133-68738141feb1 |
| <b>Client secret</b>   | xxxxxx                               |
| <b>Resource path</b>   | disabled                             |

5. Click *OK*.


### To create a firewall address to associate with the configured SDN connector:

1. Go to *Policy & Objects > Addresses* and select *Address*.
2. Click *Create new*.

## 3. Enter the following:

|                             |                              |
|-----------------------------|------------------------------|
| <b>Name</b>                 | azure-address                |
| <b>Type</b>                 | Dynamic                      |
| <b>Sub Type</b>             | Fabric Connector Address     |
| <b>SDN Connector</b>        | azure1                       |
| <b>Addresses to collect</b> | Private                      |
| <b>Filter</b>               | SecurityGroup=edsouza-centos |
| <b>Interface</b>            | Any                          |

## New Address

|                      |   |
|----------------------|---|
| Name                 | <input type="text" value="azure-address"/>  |
| Color                |  <input type="button" value="Change"/> |
| Interface            | <input type="checkbox"/> any <input type="checkbox"/> any <input type="checkbox"/> any                                  |
| Type                 | Dynamic   |
| Sub Type             | Fabric Connector Address  |
| SDN Connector        | azure1  |
| Addresses to collect | Any Private Public  |
| Filter               | SecurityGroup=edsouza-centos  |
| Comments             | <input type="text" value="Write a comment..."/> 0/255   |

OK

Cancel

4. Click *OK*.**To use the firewall address in an SD-WAN service rule:**

1. Go to *Network > SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
2. Set the *Name* to *Azure1*.
3. For the *Destination Address* select *azure-address*.
4. Configure the remaining settings as needed. See [SD-WAN rules on page 851](#) for details.
5. Click *OK*.

**Diagnostics**

Use the following CLI commands to check the status of and troubleshoot the connector.

**To see the status of the SDN connector:**

```
# diagnose sys sdn status
SDN Connector      Type      Status      Updating      Last update
-----
azure1             azure     connected   no            n/a
```

**To debug the SDN connector to resolve the firewall address:**

```
# diagnose debug application azd -1
  Debug messages will be on for 30 minutes.

...
azd sdn connector azure1 start updating IP addresses
azd checking firewall address object azure-address-1, vd 0
  IP address change, new list:
    10.18.0.4
    10.18.0.12
    ...
    ...

# diagnose sys sdwan service4

Service(2): Address Mode(IPV4) flags=0x0
  TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Service role: standalone
  Member sub interface:
  Members:
    1: Seq_num(1), alive, selected
  Dst address:
    10.18.0.4 - 10.18.0.4
    10.18.0.12 - 10.18.0.12
    ... ..
    ... ..
    ... ..
```

## Application steering using SD-WAN rules

This topic covers how to use application steering in a topology with multiple WAN links. The following examples illustrate how to use different strategies to perform application steering to accommodate different business needs:

- [Application matching on page 886](#)
- [Static application steering with a manual strategy on page 886](#)
- [Dynamic application steering with lowest cost and best quality strategies on page 889](#)

By default, individual applications and application groups cannot be selected in SD-WAN rules. To enable this functionality in the GUI, go to *System > Feature Visibility* and enable *Application Detection Based SD-WAN*. In the CLI, enter:

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

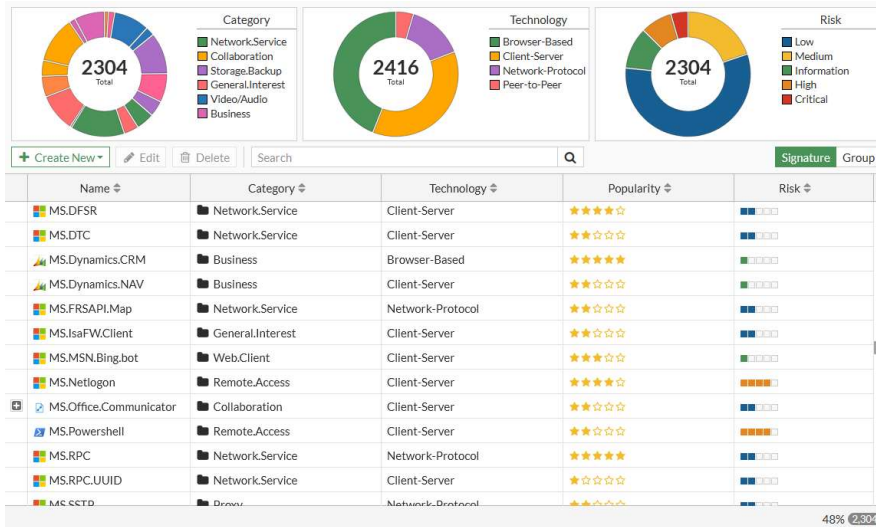


For application based steering to work, application control must be enabled in a policy. See [Application control on page 1759](#).

---

## Application matching

To apply application steering, SD-WAN service rules match traffic based on the applications that are in the application signature database. To view the signatures, go to *Security Profiles > Application Signatures* and select *Signature*.



On the first session that passes through, the IPS engine processes the traffic in the application layer to match it to a signature in the application signature database. The first session does not match any SD-WAN rules because the signature has not been recognized yet. When the IPS engine recognizes the application, it records the 3-tuple IP address, protocol, and port in the application control Internet Service ID list. To view the application and corresponding 3-tuple:

```
# diagnose sys sdwan internet-service-app-ctrl-list [app ID]
52.114.142.254
Microsoft.Teams (43541 4294837333): 52.114.142.254 6 443 Fri Jun 18 13:52:18 2021
```

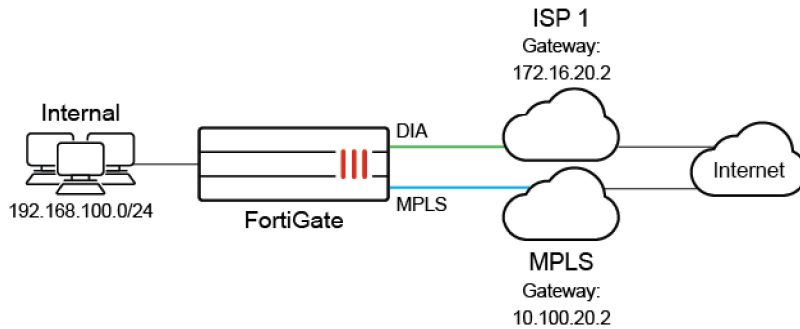
The recognized application and 3-tuple stay in the application control list for future matches to occur. If there are no hits on the entry for eight hours, the entry is deleted.



For services with multiple IP addresses, traffic might not match the expected SD-WAN rule because the traffic is destined for an IP address that has not previously been recognized by the FortiGate. The `diagnose sys sdwan internet-service-app-ctrl-list` command can be used to help troubleshoot such situations.

## Static application steering with a manual strategy

This example covers a typical usage scenario where the SD-WAN has two members: MPLS and DIA. DIA is primarily used for direct internet access to internet applications, such as Office365, Google applications, Amazon, and Dropbox. MPLS is primarily used for SIP, and works as a backup when DIA is not working.



This example configures all SIP traffic to use MPLS while all other traffic uses DIA. If DIA is not working, the traffic will use MPLS.

By default, individual applications and application groups cannot be selected in SD-WAN rules. To enable this functionality in the GUI, go to *System > Feature Visibility* and enable *Application Detection Based SD-WAN*. In the CLI, enter:

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

#### To configure an SD-WAN rule to use SIP and DIA in the GUI:

1. Add port1 (DIA) and port2 (MPLS) as SD-WAN members, and configure a static route. See [Configuring the SD-WAN interface on page 786](#) for details.
2. Create a firewall policy with an *Application Control* profile configured. See [Configuring firewall policies for SD-WAN on page 789](#) for details.
3. Go to *Network > SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
4. Enter a name for the rule, such as *SIP*.
5. Click the *Application* field and select the applicable SIP applications from the *Select Entries* panel.
6. Under *Outgoing Interfaces*, select *Manual*.
7. For *Interface preference*, select *MPLS*.
8. Click *OK*.
9. Click *Create New* to create another rule.
10. Enter a name for the rule, such as *Internet*.
11. Click the *Address* field and select *all* from the panel.
12. Under *Outgoing Interfaces*, select *Manual*.
13. For *Interface preference*, select *DIA*.
14. Click *OK*.

#### To configure the firewall policy using the CLI:

```
config firewall policy
    edit 1
        set name "1"
        set srcintf "dmz"
        set dstintf "virtual-wan-link"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
```

```

    set service "ALL"
    set utm-status enable
    set fsso disable
    set application-list "default"
    set ssl-ssh-profile "certificate-inspection"
    set nat enable
  next
end

```

### To configure an SD-WAN rule to use SIP and DIA using the CLI:

```

config system sdwan
  set status enable
  config members
    edit 1
      set interface "MPLS"
    next
    edit 2
      set interface "DIA"
    next
  end
  config service
    edit 1
      set name "SIP"
      set internet-service enable
      set internet-service-app-ctrl 34640 152305677 38938 26180 26179 30251
      set priority-members 2
    next
    edit 2
      set name "Internet"
      set dst "all"
      set priority-members 1
    next
  end
end

```

All SIP traffic uses MPLS. All other traffic goes to DIA. If DIA is broken, the traffic uses MPLS. If you use VPN instead of MPLS to run SIP traffic, you must configure a VPN interface, for example `vpn1`, and then replace member 1 from MPLS to `vpn1` for SD-WAN member.



If no SD-WAN zone is specified, members are added to the default *virtual-wan-link* zone.

---

### To use the `diagnose` command to check performance SLA status using the CLI:

```

# diagnose sys sdwan service4 1

Service(1): Address Mode(IPV4) flags=0x0

TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
Members:<<BR>>

1: Seq_num(1), alive, selected

```

```

Internet Service: SIP(4294836224 34640) SIP.Method(4294836225 152305677) SIP.Via.NAT
(4294836226 38938) SIP_Media.Type.Application(4294836227 26180) SIP_Message(4294836228
26179) SIP_Voice(4294836229 30251)

# diagnose sys sdwan service4 2

Service(2): Address Mode(IPV4) flags=0x0

TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
Members:<<BR>>

1: Seq_num(2), alive, selected

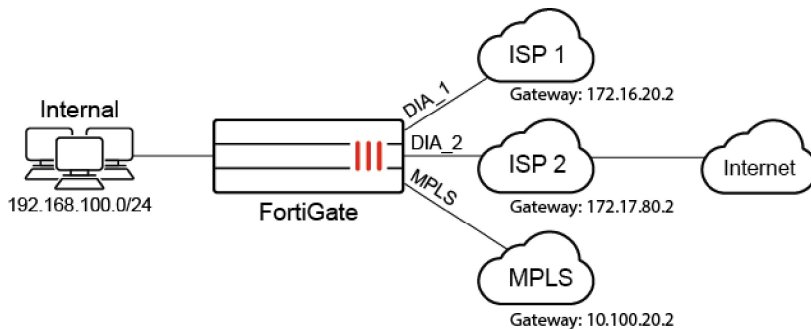
Dst address: 0.0.0.0-255.255.255.255

# diagnose sys sdwan internet-service-app-ctrl-list
Ctrl application(SIP 34640):Internet Service ID(4294836224)
Ctrl application(SIP.Method 152305677):Internet Service ID(4294836225)
Ctrl application(SIP.Via.NAT 38938):Internet Service ID(4294836226)
Ctrl application(SIP_Media.Type.Application 26180):Internet Service ID(4294836227)
Ctrl application(SIP_Message 26179):Internet Service ID(4294836228)
Ctrl application(SIP_Voice 30251):Internet Service ID(4294836229)

```

## Dynamic application steering with lowest cost and best quality strategies

In this example, the SD-WAN has three members: two ISPs (DIA\_1 and DIA\_2) that are used for access to internet applications, and an MPLS link that is used exclusively as a backup for business critical applications.



Business applications, such as Office365, Google, Dropbox, and SIP, use the *Lowest Cost (SLA)* strategy to provide application steering, and traffic falls back to MPLS only if both ISP1 and ISP2 are down. Non-business applications, such as Facebook and Youtube, use the *Best Quality* strategy to choose between the ISPs.

By default, individual applications and application groups cannot be selected in SD-WAN rules. To enable this functionality in the GUI, go to *System > Feature Visibility* and enable *Application Detection Based SD-WAN*. In the CLI, enter:

```

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

```



**To configure the SD-WAN members, static route, and firewall policy in the GUI:**

1. Add port1 (DIA\_1), port2 (DIA\_2), and port3 (MPLS) as SD-WAN members. Set the cost of DIA\_1 and DIA\_2 to 0, and MPLS to 20. See [Configuring the SD-WAN interface on page 786](#) for details.
2. Configure a static route. See [Adding a static route on page 788](#) for details.
3. Create a firewall policy to allow traffic out on SD-WAN, with an *Application Control* profile configured. See [Configuring firewall policies for SD-WAN on page 789](#) for details.

**To configure the SD-WAN rule and performance SLA checks for business critical application in the GUI:**

1. Go to *Network > SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
2. Set the name to *BusinessCriticalApps*.  
This rule will steer your business critical traffic to the appropriate link based on the *Lowest Cost (SLA)*.
3. Set *Source address* to *all*.
4. Under *Destination*, set *Application* to your required applications. In this example: *Microsoft.Office.365*, *Microsoft.Office.Online*, *Google.Docs*, *Dropbox*, and *SIP*.
5. Under *Outgoing Interfaces*, select *Lowest Cost (SLA)*.  
The lowest cost is defined in the SD-WAN member interface settings (see [Configuring the SD-WAN interface on page 786](#)). The lowest possible cost is 0, which represents the most preferred link. In this example, DIA\_1 and DIA\_2 both have a cost of 0, while MPLS has a cost of 20 because it is used for backup.
6. In *Interface preference*, add the interfaces in order of preference when the cost of the links is tied. In this example, DIA\_1, DIA\_2, then MPLS.  
MPLS will always be chosen last, because it has the highest cost. DIA\_1 and DIA\_2 have the same cost, so an interface is selected based on their order in the *Interface preference* list.
7. Set *Required SLA target* to ensure that only links that pass your SLA target are chosen in this SD-WAN rule:
  - a. Click in the *Required SLA target* field.
  - b. In the *Select Entries* pane, click *Create*. The *New Performace SLA* pane opens.
  - c. Set *Name* to *BusinessCriticalApps\_HC*.  
This health check is used for business critical applications in your SD-WAN rule.
  - d. Leave *Protocol* set to *Ping*, and add up to two servers, such as *office.com* and *google.com*.
  - e. Set *Participants* to *Specify*, and add all three interfaces: DIA\_1, DIA\_2, and MPLS.
  - f. Enable *SLA Target*.  
The attributes in your target determine the quality of your link. The SLA target of each link is compared when determining which link to use based on the lowest cost. Links that meet the SLA target are preferred over links that fail, and move to the next step of selection based on cost. If no links meet the SLA target, then they all move to the next step.  
In this example, disable *Latency threshold* and *Jitter threshold*, and set *Packet loss threshold* to 1.
  - g. Click *OK*.
  - h. Select the new performance SLA to set it as the *Required SLA target*.When multiple SLA targets are added, you can choose which target to use in the SD-WAN rule.

Priority Rule

Name: BusinessCriticalApps

Source

Source address: all

User group:

Destination

Address:

Internet Service:

Application:

- Dropbox
- Google.Docs
- Microsoft.Office.365
- Microsoft.Office.Online
- SIP

Outgoing Interfaces

Select a strategy for how outgoing interfaces will be chosen.

Manual  
 Manually assign outgoing interfaces.

Best Quality  
 The interface with the best measured performance is selected.

Lowest Cost (SLA)  
 The interface that meets SLA targets is selected. When there is a tie, the interface with the lowest assigned cost is selected.

Maximize Bandwidth (SLA)  
 Traffic is load balanced among interfaces that meet SLA targets.

Interface preference:

- DIA\_1 (port1)
- DIA\_2 (port2)
- MPLS (port3)

Zone preference:

Required SLA target: BusinessCriticalApps\_HC

Forward DSCP:

Reverse DSCP:

Status:  Enable  Disable

SLA Details

|                         | Packet Loss | Latency | Jitter |
|-------------------------|-------------|---------|--------|
| BusinessCriticalApps_HC | 1.00%       |         |        |
| DIA_1 (port1)           | 0.00%       | 12.52ms | 1.29ms |
| DIA_2 (port2)           | 0.00%       | 12.76ms | 1.45ms |
| MPLS (port3)            | 0.00%       | 12.72ms | 1.45ms |

Additional Information

API Preview

SD-WAN Rules Setup Guides

- Implicit Rule
- Best Quality
- Lowest Cost (SLA)
- Maximize Bandwidth (SLA)

Documentation

- Online Help
- Video Tutorials

OK Cancel

8. Click **OK** to create the SD-WAN rule.

### To configure the SD-WAN rule and performance SLA checks for non-business critical application in the GUI:

1. Go to *Network > SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
2. Set the name to *NonBusinessCriticalApps*.

This rule will steer your non-business critical traffic to the appropriate link based on the *Best Quality*. No SLA target must be met, as the best link is selected based on the configured quality criteria and interface preference order.

3. Set *Source address* to *all*.
4. Under *Destination*, set *Application* to your required applications. In this example: *Facebook*, and *Youtube*.
5. Under *Outgoing Interfaces*, select *Best Quality*.
6. In *Interface preference*, add the interfaces in order of preference.

By default, a more preferred link has an advantage of 10% over a less preferred link. For example, when latency is used, the preferred link's calculated latency = real latency / (1+10%).

---

The preferred link advantage can be customized in the CLI when the mode is `priority` (*Best Quality*) or `auto`:



```
config system sdwan
  config service
    edit <id>
      set link-cost-threshold <integer>
    next
  end
end
```

---

**7.** Create and apply a new performance SLA profile:

- a.** Click in the *Measured SLA* field.
- b.** In the drop-down list, click *Create*. The *New Performace SLA* pane opens.
- c.** Set *Name* to *NonBusinessCritical\_HC*.  
This health check is used for non-business critical applications in your SD-WAN rule.
- d.** Leave *Protocol* set to *Ping*, and add up to two servers, such as *youtube.com* and *facebook.com*.
- e.** Set *Participants* to *Specify*, and add the *DIA\_1* and *DIA\_2* interfaces. In this example, MPLS is not used for non-business critical applications.
- f.** Leave *SLA Target* disabled.
- g.** Click *OK*.
- h.** Select the new performance SLA from the list to set it as the *Measured SLA*.

**8.** Set *Quality criteria* as required. In this example, *Latency* is selected.

For bandwidth related criteria, such as *Downstream*, *Upstream*, and *Bandwidth* (bi-directional), the selection is based on available bandwidth. An estimated bandwidth should be configured on the interface to provide a baseline, maximum available bandwidth.

**Priority Rule**

Name: NonBusinessCriticalApps

**Source**

Source address: all

User group: +

**Destination**

Address: +

Internet Service: +

Application: Facebook, YouTube

**Outgoing Interfaces**

Select a strategy for how outgoing interfaces will be chosen.

Manual  
Manually assign outgoing interfaces.

**Best Quality**  
The interface with the best measured performance is selected.

Lowest Cost (SLA)  
The interface that meets SLA targets is selected. When there is a tie, the interface with the lowest assigned cost is selected.

Maximize Bandwidth (SLA)  
Traffic is load balanced among interfaces that meet SLA targets.

Interface preference: DIA\_1 (port1), DIA\_2 (port2)

Zone preference: +

Measured SLA: NonBusinessCriticalApps\_HC

Quality criteria: Latency

Forward DSCP:

Reverse DSCP:

Status:  Enable  Disable

Additional Information:

API Preview

SD-WAN Rules Setup Guides

- Implicit Rule
- Best Quality
- Lowest Cost (SLA)
- Maximize Bandwidth (SLA)

Documentation

- Online Help
- Video Tutorials

OK Cancel

9. Click **OK** to create the SD-WAN rule.

## To configure the SD-WAN members, static route, and firewall policy in the CLI:

1. Configure the interfaces:

```
config system interface
  edit "port1"
    set ip <class_ip&net_netmask>
    set alias "DIA_1"
    set role wan
  next
  edit "port2"
    set ip <class_ip&net_netmask>
    set alias "DIA_2"
    set role wan
  next
  edit "port3"
    set ip <class_ip&net_netmask>
    set alias "MPLS"
    set role wan
  next
end
```

2. Configure the SD-WAN members:

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

```

config members
  edit 1
    set interface "port1"
    set gateway 172.16.20.2
  next
  edit 2
    set interface "port2"
    set gateway 172.17.80.2
  next
  edit 3
    set interface "port3"
    set gateway 10.100.20.2
    set cost 20
  next
end
end

```



If no SD-WAN zone is specified, members are added to the default *virtual-wan-link* zone.

3. Configure a static route. See [Adding a static route on page 788](#) for details.
4. Create a firewall policy to allow traffic out on SD-WAN, with an *Application Control* profile configured. See [Configuring firewall policies for SD-WAN on page 789](#) for details.

### To configure the SD-WAN rule and performance SLA checks for business critical application in the CLI:

1. Configure the *BusinessCriticalApps\_HC* health-check:

```

config system sdwan
  config health-check
    edit "BusinessCriticalApps_HC"
      set server "office.com" "google.com"
      set members 1 2 3
      config sla
        edit 1
          set link-cost-factor packet-loss
          set packetloss-threshold 1
        next
      end
    next
  end
end
end

```

2. Configure the *BusinessCriticalApps* service to use *Lowest Cost (SLA)*:

```

config system sdwan
  config service
    edit 1
      set name "BusinessCriticalApps"
      set mode sla
      set src "all"
      set internet-service enable
      set internet-service-app-ctrl 17459 16541 33182 16177 34640
      config sla

```

```
        edit "BusinessCriticalApps_HC"
            set id 1
        next
    end
    set priority-members 1 2 3
next
end
end
```

## To configure the SD-WAN rule and performance SLA checks for non-business critical application in the CLI:

### 1. Configure the *nonBusinessCriticalApps\_HC* health-check:

```
config system sdwan
    config health-check
        edit "NonBusinessCriticalApps_HC"
            set server "youtube.com" "facebook.com"
            set members 1 2
        next
    end
end
```

### 2. Configure the *NonBusinessCriticalApps* service to use *Lowest Cost (SLA)*:

```
config system sdwan
    config service
        edit 4
            set name "NonBusinessCriticalApps"
            set mode priority
            set src "all"
            set internet-service enable
            set internet-service-app-ctrl 15832 31077
            set health-check "NonBusinessCriticalApps_HC"
            set priority-members 1 2
        next
    end
end
```

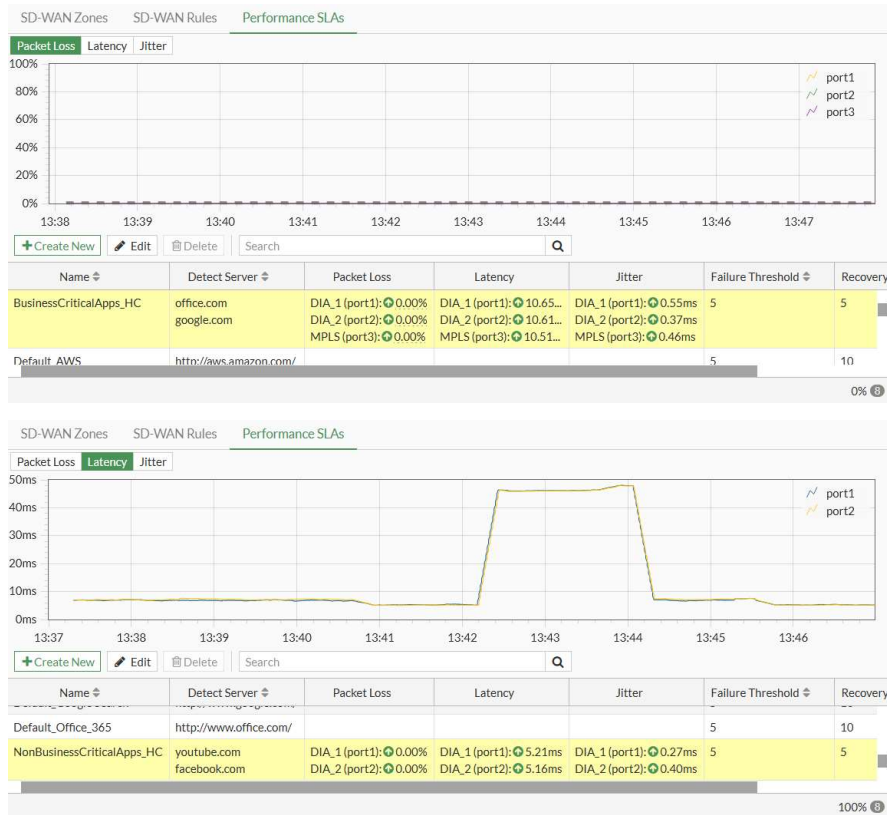
## Verification

Check the following GUI pages, and run the following CLI commands to confirm that your traffic is being steered by the SD-WAN rules.

## Health checks

To verify the status of each of the health checks in the GUI:

1. Go to *Network > SD-WAN*, select the *Performance SLAs* tab, and select each of the health checks from the list.



To verify the status of each of the health checks in the CLI:

```
# diagnose sys sdwan health-check
Health Check(BusinessCritical_HC):
Seq(1 port1): state(alive), packet-loss(0.000%) latency(12.884), jitter(0.919) sla_map=0x1
Seq(2 port2): state(alive), packet-loss(0.000%) latency(13.018), jitter(0.723) sla_map=0x1
Seq(3 port3): state(alive), packet-loss(0.000%) latency(13.018), jitter(0.923) sla_map=0x1
Health Check(NonBusinessCritical_HC):
Seq(1 port1): state(alive), packet-loss(0.000%) latency(6.888), jitter(0.953) sla_map=0x0
Seq(2 port2): state(alive), packet-loss(0.000%) latency(6.805), jitter(0.830) sla_map=0x0
```

## Rule members and hit count

To verify the active members and hit count of the SD-WAN rule in the GUI:

1. Go to *Network > SD-WAN* and select the *SD-WAN Rules* tab.

| ID              | Name                    | Source | Destination  | Criteria  | Members  | Hit Count |
|-----------------|-------------------------|--------|--|-----------|--|-----------|
| 1               | BusinessCriticalApps    | all    | Dropbox<br>Google.Docs<br>Microsoft.Office.365<br>Microsoft.Office.Online<br>SIP | SLA       | DIA_1 (port1) ✓<br>DIA_2 (port2)<br>MPLS (port3) | 45        |
| 4               | NonBusinessCriticalApps | all    | Facebook<br>YouTube  | Latency   | DIA_1 (port1) ✓<br>DIA_2 (port2)                 | 32        |
| <b>Implicit</b> |                         |        |  |           |  |           |
| sd-wan          | all                     | all    |  | Source IP | any  |           |

The interface that is currently selected by the rule has a checkmark next to its name in the *Members* column. Hover the cursor over the checkmark to open a tooltip that gives the reason why that member is selected. If multiple members are selected, only the highest ranked member is highlighted (unless the mode is *Maximize Bandwidth (SLA)*).

To verify the active members and hit count of the SD-WAN rule in the CLI:

```
# diagnose sys sdwan service4
```

```
Service(3): Address Mode(IPV4) flags=0x0
  Gen(13), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members:
    1: Seq_num(1 port1), alive, sla(0x1), cfg_order(0), cost(0), selected
    2: Seq_num(2 port2), alive, sla(0x1), cfg_order(1), cost(0), selected
    3: Seq_num(3 port3), alive, sla(0x1), cfg_order(2), cost(20), selected
  Internet Service: Dropbox(4294836727,0,0,0 17459) Google.Docs(4294836992,0,0,0 16541)
  Microsoft.Office.365(4294837472,0,0,0 33182) Microsoft.Office.Online(4294837475,0,0,0 16177)
  SIP(4294837918,0,0,0 34640)
  Src address:
    0.0.0.0-255.255.255.255
```

```
Service(4): Address Mode(IPV4) flags=0x0
  Gen(211), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(priority), link-cost-factor(latency),
  link-cost-threshold(10), heath-check(NonBusinessCritical_HC)
  Members:
    1: Seq_num(1 port1), alive, latency: 5.712, selected
    2: Seq_num(2 port2), alive, latency: 5.511, selected
  Internet Service: Facebook(4294836806,0,0,0 15832) YouTube(4294838537,0,0,0 31077)
  Src address:
    0.0.0.0-255.255.255.255
```

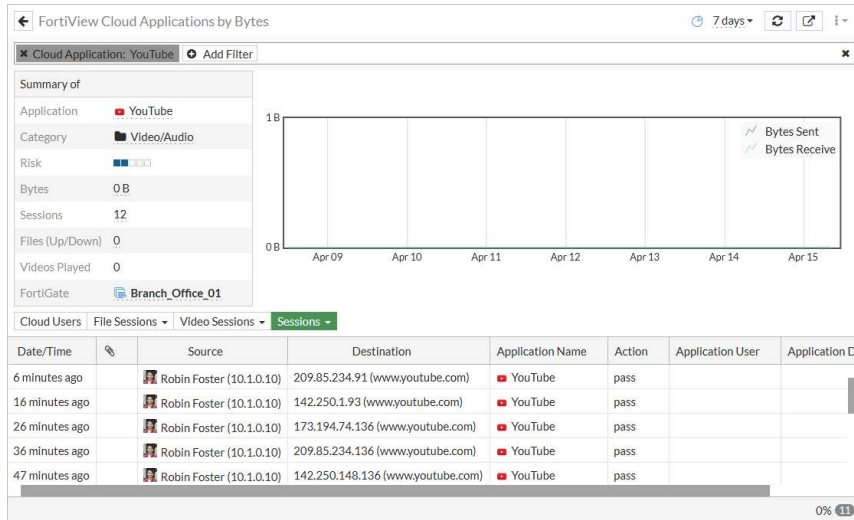
## Applications and sessions

To verify sessions in FortiView:

1. Go to a dashboard and add the *FortiView Cloud Applications* widget sorted by bytes. See [Cloud application view on page 152](#) for details.



## 2. Drill down on an application, such as *YouTube*, then select the *Sessions* tab.



### To verify applications identified by Application Control in SD-WAN:

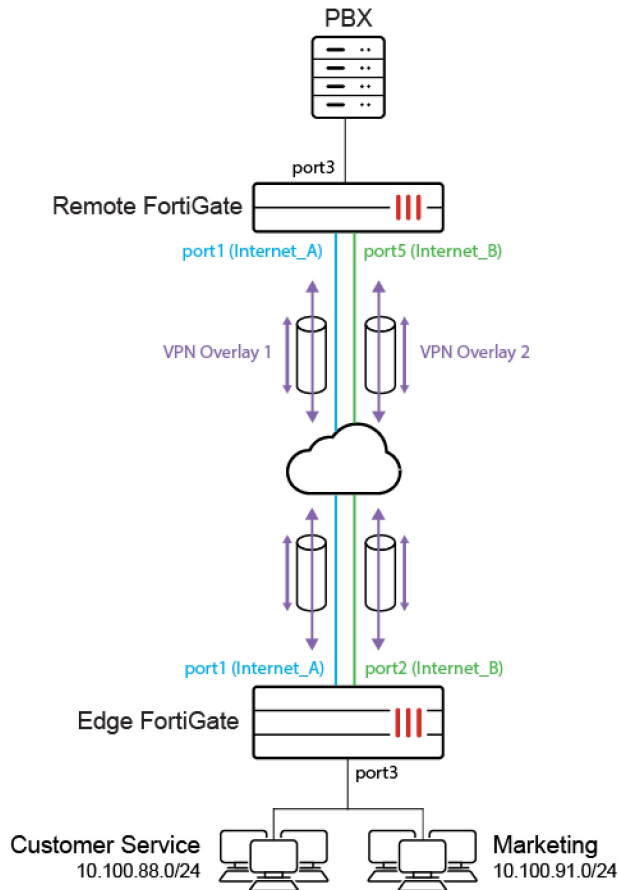
```
# diagnose sys sdwan internet-service-app-ctrl-list
```

```
Steam(16518 4294838108): 23.6.148.10 6 443 Thu Apr 15 08:51:54 2021
Netflix(18155 4294837589): 54.160.93.182 6 443 Thu Apr 15 09:13:25 2021
Netflix(18155 4294837589): 54.237.226.164 6 443 Thu Apr 15 10:04:37 2021
Minecraft(27922 4294837491): 65.8.232.41 6 443 Thu Apr 15 09:12:19 2021
Minecraft(27922 4294837491): 65.8.232.46 6 443 Thu Apr 15 09:02:07 2021
Minecraft(27922 4294837491): 99.84.244.51 6 443 Thu Apr 15 10:23:57 2021
Minecraft(27922 4294837491): 99.84.244.63 6 443 Thu Apr 15 10:03:30 2021
YouTube(31077 4294838537): 74.125.69.93 6 443 Thu Apr 15 08:52:59 2021
YouTube(31077 4294838537): 108.177.112.136 6 443 Thu Apr 15 09:33:53 2021
YouTube(31077 4294838537): 142.250.1.93 6 443 Thu Apr 15 10:35:13 2021
...
```

## DSCP tag-based traffic steering in SD-WAN

Differentiated Services Code Point (DSCP) tags can be used to categorize traffic for quality of service (QoS). SD-WAN traffic steering on an edge device can be provided based on the DSCP tags.

This section provides an example of using DSCP tag-based traffic steering using secure SD-WAN. Traffic from the customer service and marketing departments at a headquarters are marked with separate DSCP tags by the core switch and passed to the edge FortiGate. The edge FortiGate reads the tags, then steers traffic to the preferred interfaces based on the defined SD-WAN rules.



VoIP and social media traffic are steered. VoIP traffic from the customer service department is more important than social media traffic. The edge FortiGate identifies the tagged traffic based on SD-WAN rules then steers the traffic:

- VoIP traffic is marked with DSCP tag 011100 and steered to the VPN overlay with the lowest jitter, to provide the best quality voice communication with the remote PBX server.
- Social media traffic is marked with the DSCP tag 001100 and steered to the internet connection with the lowest cost.

The following is assumed to be already configured:

- Two IPsec tunnels ([IPsec VPN on page 2022](#)):
  - Branch-HQ-A on Internet\_A (port 1)
  - Branch-HQ-B on Internet\_B (port 5)
- Four SD-WAN members in two zones ([Configuring the SD-WAN interface on page 786](#)):
  - Overlay zone includes members Branch-HQ-A and Branch-HQ-B
  - virtual-wan-link zone includes members Internet\_A and Internet\_B

Internet\_A has a cost of 0 and Internet\_B has a cost of 10. When using the lowest cost strategy, Internet\_A will be preferred. Both members are participants in the Default\_DNS performance SLA.
- A static route that points to the SD-WAN interface ([Adding a static route on page 788](#)).
- Two firewall policies:

| Name | SD-WAN-OUT | Overlay-OUT |
|------|------------|-------------|
|------|------------|-------------|

|             |                  |         |
|-------------|------------------|---------|
| From        | port3            | port3   |
| To          | virtual-wan-link | Overlay |
| Source      | all              | all     |
| Destination | all              | all     |
| Schedule    | always           | always  |
| Service     | all              | all     |
| Action      | Accept           | Accept  |
| NAT         | enabled          | enabled |

After the topology is configured, you can proceed with the configuration of the edge FortiGate:

- [Configuring SD-WAN rules on page 900](#)
- [Results on page 902](#)

## Configuring SD-WAN rules

Configure SD-WAN rules to govern the steering of DSCP tag-based traffic to the appropriate interfaces. Traffic is steered based on the criteria that are configured in the SD-WAN rules.

In this example, three SD-WAN rules are configured to govern DSCP tagged traffic:

- *VoIP-Steer* for [VoIP traffic](#).
- *Facebook-DSCP-steer* for [Social media traffic](#).
- *All-traffic* for all of the [Other web traffic](#).

After configuring the rules, go to *Network > SD-WAN* and select the *SD-WAN Rules* tab to check the rules.

### VoIP traffic

VoIP traffic is steered to the *Overlay* zone.

DSCP values are usually 6-bit binary numbers that are padded with zeros at the end. VoIP traffic with DSCP tag 011100 will become 01110000. This 8-bit binary number is represented in its hexadecimal form, 0x70, as the type of service bit pattern (`tos`) value. The type of service evaluated bits (`tos-mask`) hexadecimal value of 0xf0 (11110000 in binary) is used to check the four most significant bits in the `tos` value. The four most significant bits of the `tos` (0111) are used to match the first four bits of the DSCP tag. Only the non-zero bit positions in the `tos-mask` are used for comparison; the zero bit positions are ignored.

The *Best quality* (`priority` mode) strategy is used to select the preferred interface, with the *Quality criteria* (`link-cost-members`) set to *Jitter*. The interface with the lowest amount of jitter is selected. For more information about configuring SD-WAN rules with the *Best Quality* strategy, see [Best quality strategy on page 867](#).

### To configure the rule for DSCP tagged VoIP traffic using the CLI:

```
config sys sdwan
  config service
    edit 5
      set name "VoIP-Steer"
      set mode priority
```

```

        set tos 0x70
        set tos-mask 0xf0
        set dst "all"
        set health-check "Default_DNS"
        set link-cost-factor jitter
        set priority-members 4 3
    next
end
end

```

## Social media traffic

Social media traffic is steered to the *virtual-wan-link* zone.

DSCP values are usually 6-bit binary numbers that are padded with zeros at the end. Social media traffic with DSCP tag 001100 will become 00110000. This 8-bit binary number is represented in its hexadecimal form, 0x30, as the `tos` value. The `tos-mask` hexadecimal value of 0xf0 (11110000 in binary) is used to check the four most significant bits in the `tos` value. The four most significant bits of the `tos` (0011) are used to match the first four bits of the DSCP tag. Only the non-zero bit positions in the `tos-mask` are used for comparison; the zero bit positions are ignored.

The *Manual* (manual mode) strategy is used to select the preferred interface. Internet\_B (port5, priority member 2) is set as the preferred interface to steer all social media traffic to. For more information about configuring SD-WAN rules with the manual strategy, see [Manual strategy on page 864](#).

### To configure SD-WAN rule for DSCP tagged social media traffic using the CLI:

```

config system sdwan
    config service
        edit 3
            set name "Facebook-DSCP-steer"
            set mode manual
            set tos 0x30
            set tos-mask 0xf0
            set dst "all"
            set priority-members 2 1
        next
    end
end

```

## Other web traffic

Other web traffic is steered to the *virtual-wan-link* zone.

The *Lowest Cost (SLA)* strategy (`sla` mode) is used to select the preferred interface. The interface that meets the defined SLA targets (*Default\_DNS* in this case) is selected. If there is a tie, the interface with the lowest cost is selected, Internet\_A (port1) in this case.

For more information about configuring SD-WAN rules with the *Lowest Cost (SLA)* strategy, see [Lowest cost \(SLA\) strategy on page 871](#).

### To configure SD-WAN rule for all other web traffic using the CLI:

```

config system sdwan
    config service
        edit 2

```

```

set name "All-traffic"
set mode sla
set dst "all"
config sla
    edit "Default_DNS"
        set id 1
    next
end
set priority-members 1 2
next
end
end

```

## Results

These sections show the function of SD-WAN with respect to DSCP tagged traffic steering, and can help confirm that it is running as expected:

- [Verifying the DSCP tagged traffic on FortiGate on page 902](#)
- [Verifying the service rules on page 903](#)
- [Verifying traffic steering on the SD-WAN rules on page 904](#)
- [Verifying that steered traffic is leaving from the expected interface on page 904](#)

## Verifying the DSCP tagged traffic on FortiGate

Packet sniffing is used to verify the incoming DSCP tagged traffic. See [Using the FortiOS built-in packet sniffer for more information](#).

Wireshark is used to verify that VoIP traffic is tagged with the expected DSCP tag, 0x70 or 0x30.

### VoIP traffic marked with DSCP tag 0x70:

```
# diagnose sniffer packet any '(ip and ip[1] & 0xfc == 0x70)' 6 0 1
```

The image shows a Wireshark packet capture window. The top pane displays a list of captured packets. Packet 1 is selected, showing a UDP packet from source 10.100.88.171 to destination 10.1.0.102, with length 242. The bottom pane shows the packet details for this selected packet. A red box highlights the 'Differentiated Services Field: 0x70 (DSCP: AF32, ECN: Not-ECT)' entry. Other details include Total Length: 228, Identification: 0x49de (18910), and Flags: 0x0000. The packet data pane shows the raw bytes of the packet, with a legend at the bottom indicating the DSCP field.

| No. | Time            | Source        | Destination  | Protocol | Length | Info   |
|-----|-----------------|---------------|--------------|----------|--------|--|
| 1   | 22:59:39.814674 | 10.100.88.171 | 10.1.0.102   | UDP      | 242    | 65477 → 5061 Len=200                                   |
| 2   | 22:59:39.814687 | 10.0.11.1     | 10.1.0.102   | UDP      | 242    | 65477 → 5061 Len=200                                   |
| 3   | 22:59:39.814699 | 10.100.65.101 | 10.100.67.13 | ESP      | 310    | ESP (SPI=0x9d0fc87a)                                   |
| 4   | 22:59:39.815641 | 10.100.88.171 | 10.1.0.102   | UDP      | 242    | 65477 → 5061 Len=200                                   |
| 5   | 22:59:39.815652 | 10.0.11.1     | 10.1.0.102   | UDP      | 242    | 65477 → 5061 Len=200                                   |
| 6   | 22:59:39.815674 | 10.100.65.101 | 10.100.67.13 | ESP      | 310    | ESP (SPI=0x9d0fc87a) , Shim6 (I2bis)[Malformed Packet] |
| 7   | 22:59:39.816494 | 10.100.88.171 | 10.1.0.102   | UDP      | 242    | 65477 → 5061 Len=200                                   |
| 8   | 22:59:39.816507 | 10.0.11.1     | 10.1.0.102   | UDP      | 242    | 65477 → 5061 Len=200                                   |
| 9   | 22:59:39.816519 | 10.100.65.101 | 10.100.67.13 | ESP      | 310    | ESP (SPI=0x9d0fc87a)                                   |
| 10  | 22:59:39.817452 | 10.100.88.171 | 10.1.0.102   | UDP      | 242    | 65477 → 5061 Len=200                                   |
| 11  | 22:59:39.817469 | 10.0.11.1     | 10.1.0.102   | UDP      | 242    | 65477 → 5061 Len=200                                   |
| 12  | 22:59:39.817561 | 10.100.65.101 | 10.100.67.13 | ESP      | 310    | ESP (SPI=0x9d0fc87a)                                   |
| 13  | 22:59:39.818469 | 10.100.88.171 | 10.1.0.102   | UDP      | 242    | 65477 → 5061 Len=200                                   |
| 14  | 22:59:39.818481 | 10.0.11.1     | 10.1.0.102   | UDP      | 242    | 65477 → 5061 Len=200                                   |

Frame 1: 242 bytes on wire (1936 bits), 242 bytes captured (1936 bits) on interface  
 Ethernet II, Src: Fortinet\_00:08:01 (00:09:0f:00:03:01), Dst: 00:00:00:00:00:01 (00:00:00:00:00:01)  
 Internet Protocol Version 4, Src: 10.100.88.171, Dst: 10.1.0.102  
 6100 ... = Version: 4  
 ... 0101 = Header Length: 20 bytes (5)  
 Differentiated Services Field: 0x70 (DSCP: AF32, ECN: Not-ECT)  
 Total Length: 228  
 Identification: 0x49de (18910)  
 Flags: 0x0000  
 Fragment offset: 0  
 Time to live: 127  
 Protocol: UDP (17)  
 Header checksum: 0x3345 [validation disabled]  
 [Header checksum status: Unverified]  
 Source: 10.100.88.171  
 Destination: 10.1.0.102  
 User Datagram Protocol, Src Port: 65477, Dst Port: 5061  
 Data (200 bytes)

0000 00 00 00 00 01 00 09 0f 00 03 01 00 00 45 70 .....Ep  
 Differentiated Services Field (p.dsfield), 1 byte | Packets: 111 · Displayed: 111 (100.0%) | Profile: Default

## Web traffic marked with DSCP tag 0x30:

```
# diagnose sniffer packet any '(ip and ip[1] & 0xfc == 0x30)' 6 0 1
```

The image shows a Wireshark packet capture window. The top pane displays a list of packets. Packet 6 is selected, showing a TCP segment from 10.100.91.100 to 157.240.2.174. The bottom pane shows the packet details for the selected packet, highlighting the Differentiated Services Field (DSCP) in the IP header. The DSCP value is 0x30 (AF12, ECN: Not-ECT). The packet is a TCP segment with sequence number 44513 and acknowledgment number 443.

| No. | Time            | Source        | Destination   | Protocol | Length | Info   |
|-----|-----------------|---------------|---------------|----------|--------|--|
| 1   | 22:45:39.816774 | 10.100.91.100 | 157.240.2.174 | TCP      | 66     | 44513 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1           |
| 2   | 22:45:39.828483 | 157.240.2.174 | 10.100.91.100 | TCP      | 66     | 443 → 44513 [SYN, ACK] Seq=0 Ack=1 Win=28800 Len=0 MSS=1400 SACK_PERM=1 WS=2 |
| 3   | 22:45:39.829729 | 10.100.91.100 | 157.240.2.174 | TCP      | 54     | 44513 → 443 [ACK] Seq=1 Ack=1 Win=131584 Len=0                               |
| 4   | 22:45:39.832995 | 10.100.91.100 | 157.240.2.174 | TLSv1.3  | 571    | Client Hello   |
| 5   | 22:45:39.843907 | 157.240.2.174 | 10.100.91.100 | TCP      | 54     | 443 → 44513 [ACK] Seq=1 Ack=518 Win=29184 Len=0                              |
| 6   | 22:45:39.845792 | 157.240.2.174 | 10.100.91.100 | TLSv1.3  | 1454   | Server Hello, Change Cipher Spec, Application Data                           |
| 7   | 22:45:39.845849 | 157.240.2.174 | 10.100.91.100 | TLSv1.3  | 1454   | Application Data [TCP segment of a reassembled PDU]                          |
| 8   | 22:45:39.845853 | 157.240.2.174 | 10.100.91.100 | TLSv1.3  | 645    | Application Data   |
| 9   | 22:45:39.846987 | 10.100.91.100 | 157.240.2.174 | TCP      | 54     | 44513 → 443 [ACK] Seq=518 Ack=3392 Win=131584 Len=0                          |
| 10  | 22:45:39.868813 | 10.100.91.100 | 157.240.2.174 | TLSv1.3  | 118    | Change Cipher Spec, Application Data   |
| 11  | 22:45:39.870612 | 10.100.91.100 | 157.240.2.174 | TLSv1.3  | 224    | Application Data   |
| 12  | 22:45:39.870675 | 10.100.91.100 | 157.240.2.174 | TLSv1.3  | 437    | Application Data   |
| 13  | 22:45:39.880139 | 157.240.2.174 | 10.100.91.100 | TLSv1.3  | 230    | Application Data   |
| 14  | 22:45:39.880178 | 157.240.2.174 | 10.100.91.100 | TLSv1.3  | 128    | Application Data   |

Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)  
 Ethernet II, Src: Fortinet\_00:03:01 (00:09:0f:00:03:01), Dst: 00:00:00:00:00:01 (00:00:00:00:00:01)  
 Internet Protocol Version 4, Src: 10.100.91.100, Dst: 157.240.2.174  
 0100 .... = Version: 4  
 .... 0101 = Header Length: 20 bytes (5)  
 Differentiated Services Field: 0x30 (DSCP: AF12, ECN: Not-ECT)  
 Total Length: 52  
 Identification: 0x6f56 (28502)  
 Flags: 0x4000, Don't fragment  
 Fragment offset: 0  
 Time to live: 127  
 Protocol: TCP (6)  
 Header checksum: 0x85d7 [validation disabled]  
 [Header checksum status: Unverified]  
 Source: 10.100.91.100  
 Destination: 157.240.2.174  
 Transmission Control Protocol, Src Port: 44513, Dst Port: 443, Seq: 0, Len: 0

## Verifying the service rules

To check that the expected DSCP tags and corresponding interfaces are used by the SD-WAN rules to steer traffic:

```
# diagnose sys sdwan service4
```

```
Service(5): Address Mode(IPV4) flags=0x0
  Gen(1), TOS(0x70/0xf0), Protocol(0: 1->65535), Mode(manual)
  Members:
    1: Seq_num(4 Branch-HQ-B), alive, selected
  Dst address:
    0.0.0.0-255.255.255.255
```

```
Service(3): Address Mode(IPV4) flags=0x0
  Gen(1), TOS(0x30/0xf0), Protocol(0: 1->65535), Mode(manual)
  Members:
    1: Seq_num(2 port5), alive, selected
  Dst address:
    0.0.0.0-255.255.255.255
```

```
Service(2): Address Mode(IPV4) flags=0x0
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members:
    1: Seq_num(1 port1), alive, sla(0x1), cfg_order(0), cost(0), selected
    2: Seq_num(2 port5), alive, sla(0x1), cfg_order(1), cost(10), selected
  Dst address:
    0.0.0.0-255.255.255.255
```

## Verifying traffic steering on the SD-WAN rules

Go to *Network > SD-WAN* and select the *SD-WAN Rules* tab to check the *Hit Count* on the SD-WAN interfaces.

| ID       | Name                | Source | Destination | Criteria  | Members  | Hit Count |
|----------|---------------------|--------|-------------|-----------|--|-----------|
| 5        | VoIP-Steer          |        | all         | Jitter    | VPN_B_Tunnel (Branch-HQ-B)<br>VPN_A_Tunnel (Branch-HQ-A) | 8,090     |
| 3        | Facebook-DSCP-steer |        | all         |           | Internet_B (port5)<br>Internet_A (port1)                 | 184       |
| 2        | All-traffic         |        | all         | SLA       | Internet_A (port1)<br>Internet_B (port5)                 | 23,505    |
| Implicit | sd-wan              | all    | all         | Source IP | any  | 0         |

## Verifying that steered traffic is leaving from the expected interface

To confirm that web traffic (port 443) flows through the correct underlay interface members, and VoIP traffic flows through the correct overlay interface members, go to *Dashboard > FortiView Policies* and double click on the policy name.

Web traffic is expected to leave on *Interface\_A (port1)* or *Interface\_B (port5)*:

| Source        | Device            | Destination    | Application    | Protocol | Source Port | Destination Port | Bytes     | Packets | Duration (seconds) | Destination Interface |
|---------------|-------------------|----------------|----------------|----------|-------------|------------------|-----------|---------|--------------------|-----------------------|
| 10.100.88.151 | 00:09:0f:00:03:01 | 216.58.192.226 | Google Ads     | TCP      | 28454       | 443              | 12.65 kB  | 47      | 35s                | Internet_A (port1)    |
| 10.100.88.151 | 00:09:0f:00:03:01 | 216.58.192.132 | HTTPS.BROWSER  | TCP      | 28432       | 443              | 12.85 kB  | 89      | 39s                | Internet_A (port1)    |
| 10.100.88.151 | 00:09:0f:00:03:01 | 13.249.135.106 | HTTPS.BROWSER  | TCP      | 28477       | 443              | 13.93 kB  | 30      | 36s                | Internet_A (port1)    |
| 10.100.88.151 | 00:09:0f:00:03:01 | 13.249.135.36  | HTTPS.BROWSER  | TCP      | 28485       | 443              | 7.75 kB   | 22      | 21s                | Internet_A (port1)    |
| 10.100.88.161 | 00:09:0f:00:03:01 | 157.240.2.25   | Facebook       | TCP      | 28449       | 443              | 321.46 kB | 264     | 35s                | Internet_B (port5)    |
| 10.100.88.151 | 00:09:0f:00:03:01 | 69.147.64.34   | Yahoo.Services | TCP      | 28436       | 443              | 8.80 kB   | 28      | 39s                | Internet_A (port1)    |
| 10.100.88.161 | 00:09:0f:00:03:01 | 157.240.18.19  | Facebook       | TCP      | 28413       | 443              | 8.45 kB   | 33      | 2m 13s             | Internet_B (port5)    |
| 10.100.88.161 | 00:09:0f:00:03:01 | 157.240.18.174 | Instagram      | TCP      | 28411       | 443              | 193.70 kB | 267     | 2m 14s             | Internet_B (port5)    |
| 10.100.88.161 | 00:09:0f:00:03:01 | 69.171.250.63  | Instagram      | TCP      | 28410       | 443              | 23.42 kB  | 58      | 2m 16s             | Internet_B (port5)    |
| 10.100.88.161 | 00:09:0f:00:03:01 | 69.171.250.63  | Instagram      | TCP      | 28412       | 443              | 10.87 kB  | 40      | 2m 14s             | Internet_B (port5)    |

VoIP traffic is expected to leave on the preferred *VPN\_B\_Tunnel (Branch-HQ-B)* interface:

| Source        | Device            | Destination | Application | Protocol | Source Port | Destination Port | Bytes   | Packets | Duration (seconds) | Destination Interface      |
|---------------|-------------------|-------------|-------------|----------|-------------|------------------|---------|---------|--------------------|----------------------------|
| 10.100.88.171 | 00:09:0f:00:03:01 | 10.1.0.102  | TCP/5061    | TCP      | 34779       | 5061             | 728 B   | 14      | 17s                | VPN_B_Tunnel (Branch-HQ-B) |
| 10.100.88.171 | 00:09:0f:00:03:01 | 10.1.0.102  | UDP/5061    | UDP      | 65477       | 5061             | 1.84 MB | 8,084   | 3m 16s             | VPN_B_Tunnel (Branch-HQ-B) |
| 10.100.88.171 | 00:09:0f:00:03:01 | 10.1.0.102  | UDP/5061    | UDP      | 65478       | 5061             | 32 B    | 1       | 2m 4s              | VPN_B_Tunnel (Branch-HQ-B) |

## ECMP support for the longest match in SD-WAN rule matching

The longest match SD-WAN rule can match ECMP best routes. The rule will select the egress ports on ECMP specific routes, and not the less specific routes, to transport traffic.

The service mode determines which egress port on the ECMP specific routes is selected to forward traffic:

- Manual (`manual`): The first configured alive port is selected.
- Best Quality (`priority`): The best quality port is selected.
- Lowest Cost (`sla`): The first configured or lower cost port in SLA is selected.

### Example

By default, SD-WAN selects the outgoing interface from all of the links that have valid routes to the destination. In some cases, it is required that only the links that have the best (or longest match) routes (single or ECMP) to the destination are considered.



In this example, four SD-WAN members in two zones are configured. The remote PC (PC\_2 - 10.1.100.22) is accessible on port15 and port16, even though there are valid routes for all of the SD-WAN members. A single SD-WAN service rule is configured that allows traffic to be balanced between all four of the members, but only chooses between port15 and port16 for the specific 10.1.100.22 address.

A performance SLA health check is configured to monitor 10.1.100.2. An SD-WAN service rule in Lowest Cost (SLA) mode is configured to select the best interface to steer the traffic. In the rule, the method of selecting a member if more than one meets the SLA (`tie-break`) is configured to select members that meet the SLA and match the longest prefix in the routing table (`fib-best-match`). If there are multiple ECMP routes with the same destination, the FortiGate will take the longest (or best) match in the routing table, and choose from those interface members.

### To configure the SD-WAN:

```
config system sdwan
  config zone
    edit "virtual-wan-link"
    next
    edit "z1"
    next
  end
  config members
    edit 1
    set interface "port1"
    set gateway 172.16.200.2
    next
    edit 2
    set interface "dmz"
    set gateway 172.16.208.2
    next
  end
end
```



```

edit 3
    set interface "port15"
    set zone "z1"
    set gateway 172.16.209.2
next
edit 4
    set interface "port16"
    set zone "z1"
    set gateway 172.16.210.2
next
end
config health-check
    edit "1"
        set server "10.1.100.2"
        set members 0
        config sla
            edit 1
                next
            end
        next
    end
end
config service
    edit 1
        set name "1"
        set mode sla
        set dst "all"
        set src "172.16.205.0"
        config sla
            edit "1"
                set id 1
            next
        end
        set priority-members 1 2 3 4
        set tie-break fib-best-match
    next
end
end

```

### To check the results:

1. The debug shows the SD-WAN service rule. All of the members meet SLA, and because no specific costs are attached to the members, the egress interface is selected based on the interface priority order that is configured in the rule:

```

FGT_A (root) # diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
Gen(4), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
Members(4):
  1: Seq_num(1 port1), alive, sla(0x1), gid(0), cfg_order(0), cost(0), selected
  2: Seq_num(2 dmz), alive, sla(0x1), gid(0), cfg_order(1), cost(0), selected
  3: Seq_num(3 port15), alive, sla(0x1), gid(0), cfg_order(2), cost(0), selected
  4: Seq_num(4 port16), alive, sla(0x1), gid(0), cfg_order(3), cost(0), selected
Src address(1):
  172.16.205.0-172.16.205.255

```

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

- The routing table shows that there are ECMP default routes on all of the members, and ECMP specific (or best) routes only on port15 and port16:

```
FGT_A (root) # get router info routing-table static
Routing table for VRF=0
S*      0.0.0.0/0 [1/0] via 172.16.200.2, port1
        [1/0] via 172.16.208.2, dmz
        [1/0] via 172.16.209.2, port15
        [1/0] via 172.16.210.2, port16
S       10.1.100.22/32 [10/0] via 172.16.209.2, port15
        [10/0] via 172.16.210.2, port16
```

Because `tie-break` is set to `fib-best-match`, the first configured member from port15 and port16 is selected to forward traffic to PC\_2. For all other traffic, the first configured member from all four of the interfaces is selected to forward traffic.

- On PC-1, generate traffic to PC-2:

```
ping 10.1.100.22
```

- On FGT\_A, sniff for traffic sent to PC\_2:

```
# diagnose sniffer packet any 'host 10.1.100.22' 4
interfaces=[any]
filters=[host 10.1.100.22]
2.831299 port5 in 172.16.205.11 -> 10.1.100.22: icmp: echo request
2.831400 port15 out 172.16.205.11 -> 10.1.100.22: icmp: echo request
```

Traffic is leaving on port15, the first configured member from port15 and port16.

## Override quality comparisons in SD-WAN longest match rule matching

In SD-WAN rules, the longest match routes will override the quality comparisons when all of the specific routes are out of SLA.

With this feature in an SD-WAN rule:

- Lowest Cost (`sla`):** Even though all of the egress ports on specific routes (longest matched routes) are out of SLA, the SD-WAN rule still selects the first configured or lower-cost port from the egress ports to forward traffic.
- Best Quality (`priority`):** Even though the egress ports on specific routes (longest matched routes) have worse quality than all other ports on less specific routes, the SD-WAN rule still selects the best quality port from the ports on specific routes to forward traffic.

This feature avoids a situation where, if the members on specific routes (longest matched routes) are out of SLA or have worse quality, the traffic might be forwarded to the wrong members in SLA (higher quality) on the default or aggregate routes.

## Example



In this example, four SD-WAN members in two zones are configured. The remote PC (PC\_2 - 10.1.100.22) is accessible on port15 and port16, even though there are valid routes for all of the SD-WAN members. A single SD-WAN service rule is configured that allows traffic to be balanced between all four of the members, but only chooses between port15 and port16 for the specific 10.1.100.22 address. If neither port15 nor port16 meet the SLAs, traffic will be forwarded on one of these interfaces, instead of on port1 or dmz.

A performance SLA health check is configured to monitor 10.1.100.2. An SD-WAN service rule in Lowest Cost (SLA) mode is configured to select the best interface to steer the traffic. In the rule, the method of selecting a member if more than one meets the SLA (*tie-break*) is configured to select members that meet the SLA and match the longest prefix in the routing table (*fib-best-match*). If there are multiple ECMP routes with the same destination, the FortiGate will take the longest (or best) match in the routing table, and choose from those interface members.

### To configure the SD-WAN:

```
config system sdwan
  config zone
    edit "virtual-wan-link"
    next
    edit "z1"
    next
  end
  config members
    edit 1
      set interface "port1"
      set gateway 172.16.200.2
    next
    edit 2
      set interface "dmz"
      set gateway 172.16.208.2
    next
    edit 3
      set interface "port15"
      set zone "z1"
      set gateway 172.16.209.2
    next
    edit 4
      set interface "port16"
      set zone "z1"
      set gateway 172.16.210.2
    next
  end
  config health-check
    edit "1"
      set server "10.1.100.2"
      set members 0
      config sla
```

```

        edit 1
        next
    end
next
end
config service
    edit 1
        set name "1"
        set mode sla
        set dst "all"
        set src "172.16.205.0"
        config sla
            edit "1"
                set id 1
            next
        end
        set priority-members 1 2 3 4
        set tie-break fib-best-match
    next
end
end

```

### To check the results:

1. The debug shows the SD-WAN service rule. Both port15 and port16 are up, but out of SLA:

```

FGT_A (root) # diagnose sys sdwan service4
Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
Gen(3), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
Members(4):
  1: Seq_num(1 port1), alive, sla(0x1), gid(0), cfg_order(0), cost(0), selected
  2: Seq_num(2 dmz), alive, sla(0x1), gid(0), cfg_order(1), cost(0), selected
  3: Seq_num(3 port15), alive, sla(0x0), gid(0), cfg_order(2), cost(0), selected
  4: Seq_num(4 port16), alive, sla(0x0), gid(0), cfg_order(3), cost(0), selected
Src address(1):
  172.16.205.0-172.16.205.255

Dst address(1):
  0.0.0.0-255.255.255.255

```

2. The routing table shows that there are ECMP default routes on all of the members, and ECMP specific (or best) routes only on port15 and port16:

```

FGT_A (root) # get router info routing-table static
Routing table for VRF=0
S*   0.0.0.0/0 [1/0] via 172.16.200.2, port1
      [1/0] via 172.16.208.2, dmz
      [1/0] via 172.16.209.2, port15
      [1/0] via 172.16.210.2, port16
S    10.1.100.22/32 [10/0] via 172.16.209.2, port15
      [10/0] via 172.16.210.2, port16

```

Because `tie-break` is set to `fib-best-match`, even though both port15 and port16 are out of SLA, the first configured member of the two (port15) is selected to forward traffic to PC\_2. For all other traffic, the first configured member from all of the interfaces that are in SLA is selected to forward traffic (port1).

3. On PC-1, generate traffic to PC-2:

```
ping 10.1.100.22
```

#### 4. On FGT\_A, sniff for traffic sent to PC\_2:

```
# diagnose sniffer packet any 'host 10.1.100.22' 4
interfaces=[any]
filters=[host 10.1.100.22]
2.831299 port5 in 172.16.205.11 -> 10.1.100.22: icmp: echo request
2.831400 port15 out 172.16.205.11 -> 10.1.100.22: icmp: echo request
```

Traffic is leaving on port15, the first configured member from port15 and port16, even though both are out of SLA.

## Internet service and application control steering

An application, application group, or application category can be selected as an SD-WAN service rule destination criterion for IPv4 and IPv6 address modes.

To configure from the CLI:

```
config system sdwan
  config service
    edit <id>
      set internet-service enable
      set internet-service-app-ctrl <app id> [app id]
      set internet-service-app-ctrl-group <app group> [app group]
      set internet-service-app-ctrl-category <category id> [category id]
    next
  end
end
```

To configure for IPv6 addressing mode from the CLI, enable `addr-mode ipv6`:

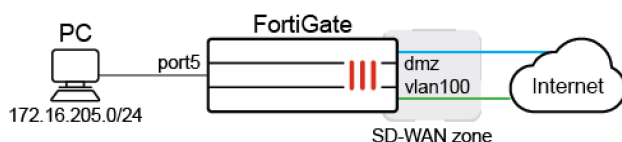
```
config system sdwan
  config service
    edit <id>
      set addr-mode ipv6
    next
  end
end
```

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

This topic includes a GUI and CLI [Example for application category on page 910](#) and a CLI [Example for IPv6 on page 915](#).

### Example for application category

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 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*.



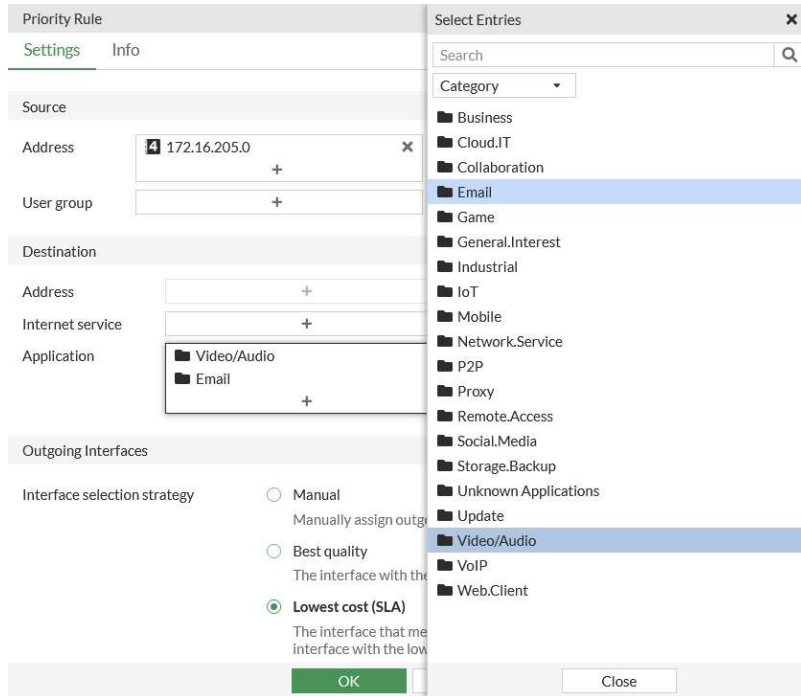
To enable GUI visibility of application detection based SD-WAN in the CLI:

```
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:

|                   |                |
|-------------------|----------------|
| <b>Name</b>       | <i>1</i>       |
| <b>Protocol</b>   | <i>DNS</i>     |
| <b>Server</b>     | <i>8.8.8.8</i> |
| <b>SLA Target</b> | <i>Enable</i>  |

- 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:

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

- c. Click **OK**.

### 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
```

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

```

## To test the configuration:

### 1. 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

```

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

```

# diagnose sys sdwan internet-service-app-ctrl-list cat-id 5
List App Ctrl Database Entry(IPv4) in Kernel:

```

```

Max_App_Ctrl_Size=32768 Num_App_Ctrl_Entry=4

```

```

YouTube(31077 4294838537): IP=142.250.217.110 6 443
YouTube(31077 4294838537): IP= 173.194.152.89 6 443
YouTube(31077 4294838537): IP= 173.194.152.170 6 443
YouTube(31077 4294838537): IP= 209.52.146.205 6 443

```

```

# diagnose sys sdwan internet-service-app-ctrl-list cat-id 21
List App Ctrl Database Entry(IPv4) in Kernel:

```

```

Max_App_Ctrl_Size=32768 Num_App_Ctrl_Entry=1

```

```

Gmail(15817 4294836957): IP=172.217.14.197 6 443

```

### 3. 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

```

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

### 5. 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

```

## Example for IPv6

In this example, SD-WAN is configured to use an IPv6 service rule to steer traffic from FGT\_A to FGT\_B based on the following application control options:

- Application Telnet
- An application group for ping
- An application category that includes SSH

When the rule is matched, traffic is steered based on the lowest cost SLA strategy. In this example, vlan100 is the preferred interface, and traffic is routed to vlan100 on FGT\_B.

### To view the configuration:

#### 1. View the SD-WAN configuration on FGT\_A:

SD-WAN has four members in the default virtual-wan-link zone, each with an IPv4 and IPv6 gateway. The SD-WAN service rule includes `internet-service-app-ctrl 16091` for the Telnet, `internet-service-app-ctrl-group "network-Ping"` for ping, and `internet-service-app-ctrl-category 15` for SSH applications.

```

(sdwan) # show
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
      set gateway6 2000:172:16:208::2
    next
    edit 2
      set interface "IPSec-1"
    next
    edit 3
      set interface "agg1"
      set gateway 172.16.203.2
      set gateway6 2000:172:16:203::2
    next
    edit 4
      set interface "vlan100"
      set gateway 172.16.206.2
      set gateway6 2000:172:16:206::2
    next
  end

```

```

end
config health-check
  edit "1"
    set addr-mode ipv6
    set server "2000::2:2:2:2"
    set members 0
    config sla
      edit 1
        next
      end
    next
  end
end
config service
  edit 1
    set name "1"
    set addr-mode ipv6
    set mode sla
    set internet-service enable
    set internet-service-app-ctrl 16091
    set internet-service-app-ctrl-group "network-Ping"
    set internet-service-app-ctrl-category 15
    config sla
      edit "1"
        set id 1
        next
      end
    set priority-members 4 1 2 3
  next
end
end
end

```

## 2. View the default route for FGT\_A:

```

config router static
  edit 5
    set distance 1
    set sdwan-zone "virtual-wan-link"
  next
end

```

## 3. View the firewall policy for FGT\_A:

The `utm-status` option is enabled to learn application 3T (3 tuple) information, and the default application profile of `g-default` is selected.

```

config firewall policy
  edit 1
    set uuid f09bddc4-def3-51ed-8517-0d8b6bc18f35
    set srcintf "any"
    set dstintf "any"
    set action accept
    set srcaddr6 "all"
    set dstaddr6 "all"
    set schedule "always"
    set service "ALL"
    set utm-status enable
    set ssl-ssh-profile "certificate-inspection"
    set application-list "g-default"
  
```

```

    next
end

```

### To verify the configuration:

#### 1. On FGT\_A, check the routing table:

The routing table has ECMP applied to default gateways for each SD-WAN member.

```

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

```

#### 2. Check the SD-WAN service:

Based on the service rule, member 4 named vlan100 is preferred. Traffic must also match the highlighted internet services.

```

# diagnose system sdwan service

Service(1): Address Mode(IPV6) flags=0x4200 use-shortcut-sla use-shortcut
Tie break: cfg
  Gen(2), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members(4):
    1: Seq_num(4 vlan100), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
    selected
    2: Seq_num(1 dmz), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
    selected
    3: Seq_num(2 IPSec-1), alive, sla(0x1), gid(0), cfg_order(2), local cost(0),
    selected
    4: Seq_num(3 aggl), alive, sla(0x1), gid(0), cfg_order(3), local cost(0), selected
  Internet Service(3): Telnet(4294837974,0,0,0,0 16091) IPv6.ICMP(4294837087,0,0,0,0
16321) Network.Service(0,15,0,0,0)

```

#### 3. Initiate traffic for ping, Telnet, and SSH to FGT\_B, then FGT\_A will learn 3T information for these applications, and use the SD-WAN rule to route traffic for the applications to the preferred interface of vlan100.

- Following is the sniffer traffic for ping application. The ping traffic flows out of DMZ before 3T information is recognized, then out from vlan100 after T3 traffic is recognized:

```

# diagnose sniffer packet any 'host 2000::2:0:0:4' 4
interfaces=[any]
filters=[host 2000::2:0:0:4]
16.952138 port5 in 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 1
[flowlabel 0x5080d]
16.954571 dmz out 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 1
[flowlabel 0x5080d]
16.954920 dmz in 2000::2:0:0:4 -> 2000:172:16:205::100: icmp6: echo reply seq 1
16.955086 port5 out 2000::2:0:0:4 -> 2000:172:16:205::100: icmp6: echo reply seq 1
17.953277 port5 in 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 2
[flowlabel 0x5080d]
17.953455 dmz out 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 2
[flowlabel 0x5080d]
17.953622 dmz in 2000::2:0:0:4 -> 2000:172:16:205::100: icmp6: echo reply seq 2
17.953722 port5 out 2000::2:0:0:4 -> 2000:172:16:205::100: icmp6: echo reply seq 2
18.959823 port5 in 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 3
[flowlabel 0x5080d]

```

```

18.960005 vlan100 out 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq
3 [flowlabel 0x5080d]
18.960015 agg1 out 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 3
[flowlabel 0x5080d]
18.960024 port4 out 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 3
[flowlabel 0x5080d]
18.960295 vlan100 in 2000::2:0:0:4 -> 2000:172:16:205::100: icmp6: echo reply seq 3
18.960449 port5 out 2000::2:0:0:4 -> 2000:172:16:205::100: icmp6: echo reply seq 3
19.983802 port5 in 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 4
[flowlabel 0x5080d]

```

- Following is the sniffer traffic for Telnet application group. The Telnet traffic flows out of **agg1** before 3T information is recognized, then out from **vlan100** after T3 traffic is recognized:

```

# diagnose sniffer packet any 'host 2000::2:0:0:4 and dst port 23' 4 interfaces=
[any]
filters=[host 2000::2:0:0:4 and dst port 23]
4.096393 port5 in 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: syn 2723132265
[flowlabel 0xd4e65] 4.096739 agg1 out 2000:172:16:205::100.43128 ->
2000::2:0:0:4.23: syn 2723132265 [flowlabel 0xd4e65]
4.096752 port4 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: syn 2723132265
[flowlabel 0xd4e65]
.....
5.503679 port5 in 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: psh 2723132345 ack
544895389 [flowlabel 0xd4e65]
5.503894 vlan100 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: psh 2723132345
ack 544895389 [flowlabel 0xd4e65]
5.503907 agg1 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: psh 2723132345 ack
544895389 [flowlabel 0xd4e65]
5.503918 port4 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: psh 2723132345 ack
544895389 [flowlabel 0xd4e65]
5.504641 port5 in 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: ack 544895390
[flowlabel 0xd4e65]
5.504713 vlan100 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: ack 544895390
[flowlabel 0xd4e65]
5.504721 agg1 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: ack 544895390
[flowlabel 0xd4e65]
5.504728 port4 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: ack 544895390
[flowlabel 0xd4e65]

```

- Following is the sniffer traffic for SSH application category. The SSH traffic flows out of **dmz** before 3T information is recognized, then out from **vlan100** after T3 traffic is recognized:

```

# diagnose sniffer packet any 'host 2000::2:0:0:4 and dst port 22' 4
interfaces=[any]
filters=[host 2000::2:0:0:4 and dst port 22]
5.910752 port5 in 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: syn 980547187
[flowlabel 0xf1403]
5.911002 dmz out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: syn 980547187
[flowlabel 0xf1403]
5.914550 port5 in 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: ack 583860244
[flowlabel 0xf1403]
5.914651 dmz out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: ack 583860244
[flowlabel 0xf1403]
.....
8.116507 port5 in 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: psh 980549261 ack
583862554 [class 0x10] [flowlabel 0xf1403]

```

```

8.116663 vlan100 out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: psh 980549261
ack 583862554 [class 0x10] [flowlabel 0xf1403]
8.116674 agg1 out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: psh 980549261 ack
583862554 [class 0x10] [flowlabel 0xf1403]
8.116685 port4 out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: psh 980549261 ack
583862554 [class 0x10] [flowlabel 0xf1403]
8.118135 port5 in 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: ack 583862598
[class 0x10] [flowlabel 0xf1403]
8.118171 vlan100 out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: ack 583862598
[class 0x10] [flowlabel 0xf1403]
8.118179 agg1 out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: ack 583862598
[class 0x10] [flowlabel 0xf1403]
8.118189 port4 out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: ack 583862598
[class 0x10] [flowlabel 0xf1403]

```

#### 4. View the IPv6 application control internet service ID list:

```
# diagnose system sdwan internet-service-app-ctrl6-list
```

```

Telnet(16091 4294837974): 2000::2:0:0:4 6 23 Thu Apr 20 17:43:00 2023
IPv6.ICMP(16321 4294837087): 2000::2:0:0:4 58 0 Thu Apr 20 17:43:00 2023

```

#### 5. View the IPv6 application control internet service ID list by category:

```
# diagnose system sdwan internet-service-app-ctrl6-category-list
```

```
SSH(16060 4294837772): 2000::2:0:0:4 6 22 Thu Apr 20 17:43:00 2023
```

## Use maximize bandwidth to load balance traffic between ADVPN shortcuts

When ADVPN is configured on a FortiGate spoke along with an SD-WAN rule set to *Maximize Bandwidth SLA* (GUI) or load balance mode (CLI) as well as `tie-break` set to `fib-best-match`, then spoke-to-spoke traffic is load balanced between multiple ADVPN shortcuts when the shortcuts are within the configured SLA conditions.

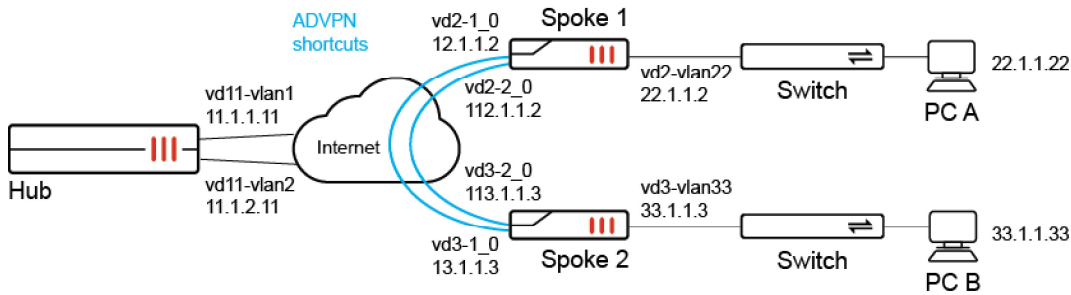
Following is an example configuration with `load-balance` enabled and `tie-break` set to `fib-best-match`:

```

config system sdwan
  config service
    edit 3
      set mode sla
      set load-balance enable
      set dst "all"
      config sla
        edit "ping"
          set id 1
        next
      end
      set priority-members 1 2
      set tie-break fib-best-match
    next
  end
end

```

## Example



In this example SD-WAN is configured between one hub and multiple spokes, and the SD-WAN configuration shows SD-WAN rule 3 with the following required settings to enable spoke-to-spoke traffic between multiple ADVPN shortcuts:

- set load-balance enable
- set tie-break fib-best-match

```
show system sdwan
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        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
                edit 2
                next
            end
        next
        edit "1"
        next
    end
    config service
        edit 1
```

```

        set dst "033"
        set priority-members 1
    next
    edit 2
        set dst "133"
        set priority-members 2
    next
    edit 3
        set mode sla
        set load-balance enable
        set dst "all"
        config sla
            edit "ping"
                set id 1
            next
        end
        set priority-members 1 2
        set tie-break fib-best-match
    next
end
end

```

To trigger spoke-to-spoke communication, run an ICMP ping on PC A with IP address 22.1.1.22 behind spoke 1 that is destined for PC B with IP address 33.1.1.33 behind spoke 2. The spoke-to-spoke traffic will be used to demonstrate load balancing between shortcuts in the CLI output of this topic.

### To verify the configuration:

#### 1. Confirm the ADVPN shortcuts are within the SLA conditions:

```

# diagnose system sdwan health-check
Health Check(ping):
Seq(1 vd2-1): state(alive), packet-loss(0.000%) latency(0.029), jitter(0.002), mos
(4.404), bandwidth-up(1999), bandwidth-dw(0), bandwidth-bi(1999) sla_map=0x3
Seq(1 vd2-1_0): state(alive), packet-loss(0.000%) latency(0.026), jitter(0.001), mos
(4.404), bandwidth-up(2000), bandwidth-dw(0), bandwidth-bi(2000) sla_map=0x3
Seq(2 vd2-2): state(alive), packet-loss(0.000%) latency(0.055), jitter(0.064), mos
(4.404), bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x3
Seq(2 vd2-2_0): state(alive), packet-loss(0.000%) latency(0.060), jitter(0.058), mos
(4.404), bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x3

```

#### 2. Confirm the settings for SD-WAN rule 3:

```

# diagnose system sdwan service 3

Service(3): Address Mode(IPV4) flags=0x4200 use-shortcut-sla use-shortcut
Tie break: fib
Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance hash-mode=round-robin)
Member sub interface(4):
    1: seq_num(1), interface(vd2-1):
        1: vd2-1_0(125)
    3: seq_num(2), interface(vd2-2):
        1: vd2-2_0(127)
Members(4):
    1: Seq_num(1 vd2-1), alive, sla(0x1), gid(2), num of pass(1), selected
    2: Seq_num(1 vd2-1_0), alive, sla(0x1), gid(2), num of pass(1), selected
    3: Seq_num(2 vd2-2), alive, sla(0x1), gid(2), num of pass(1), selected

```



```

4: Seq_num(2 vd2-2_0), alive, sla(0x1), gid(2), num of pass(1), selected
Dst address(1):
    0.0.0.0-255.255.255.255

```

### 3. Confirm firewall policing routing list:

```

# diagnose firewall proute list 2131230723
list route policy info(vf=vd2):

id=2131230723(0x7f080003) vwl_service=3 vwl_mbr_seq=1 1 2 2 dscp_tag=0xfc 0xfc
flags=0x90 load-balance hash-mode=round-robin fib-best-match tos=0x00 tos_mask=0x00
protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(4) oif=116(vd2-1) num_pass=1
oif=125(vd2-1_0) num_pass=1 oif=117(vd2-2) num_pass=1 oif=127(vd2-2_0) num_pass=1
destination(1): 0.0.0.0-255.255.255.255
source wildcard(1): 0.0.0.0/0.0.0.0
hit_count=117 last_used=2023-04-21 15:49:59

```

### 4. Confirm the 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),
01:26:14, [1/0]
                [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11),
01:26:14, [1/0]
B       1.1.1.1/32 [200/0] via 11.1.1.1 [2] (recursive via 12.1.1.1, vd2-vlan12),
01:26:14, [1/0]
B       11.11.11.11/32 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11),
01:26:14, [1/0]
                [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11),
01:26:14, [1/0]
B      33.1.1.0/24 [200/0] via 10.10.100.3 [2] (recursive is directly connected, vd2-1_
0), 01:19:41, [1/0]
                [200/0] via 10.10.200.3 [2] (recursive is directly connected, vd2-2_
0), 01:19:41, [1/0]
B       100.1.1.0/24 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11),
01:26:14, [1/0]
                [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11),
01:26:14, [1/0]

```

### 5. Check the packet sniffer output for the default setting.

This step demonstrates routing for the default setting of `set tie-break zone`. The following packet sniffer output of ICMP pings demonstrates how spoke-to-spoke traffic (ping from 22.1.1.22 to 33.1.1.13) is load balanced between all parent tunnels and shortcuts, and is not limited to shortcuts within SLA.

```

# diagnose sniffer packet any "host 33.1.1.13" 4
interfaces=[any]
filters=[host 33.1.1.13]
14.665232 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665234 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665240 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665262 vd2-1_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665274 vd3-1_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665284 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665285 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665289 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665299 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply

```

```
14.665300 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
14.665306 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
14.665314 vd3-1_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
14.665326 vd2-1_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
14.665331 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
14.665332 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
14.665337 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

24.190955 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.190957 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.190963 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.190982 vd2-2 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.190993 p2 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.191002 p2 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.191020 vd3-2 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.191031 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.191032 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.191036 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.191046 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191047 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191053 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191063 vd3-2 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191074 p2 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191079 p2 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191090 vd2-2 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191094 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191095 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191100 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

51.064984 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.064985 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.064991 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.065011 vd2-2_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.065022 vd3-2_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.065031 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.065032 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.065036 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.065046 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
51.065047 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
51.065054 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
51.065063 vd3-2_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
51.065075 vd2-2_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
51.065082 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
51.065082 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
51.065087 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

67.257123 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257125 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257131 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257150 vd2-1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257162 p1 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257170 p1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257189 vd3-1 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257199 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257200 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257205 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
```

```

67.257216 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257217 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257223 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257234 vd3-1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257245 p1 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257250 p1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257261 vd2-1 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257266 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257267 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257272 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

```

```
^C
```

```
84 packets received by filter
```

```
0 packets dropped by kernel
```

6. Check the sniffer packet output after changing the setting to `set tie-break fib-best-match`.

The following packet sniffer output of ICMP pings demonstrates how load balancing of spoke-to-spoke is limited and only occurs between shortcuts `vd2-1_0` and `vd2-2_0`, which are within SLA.

```

# diagnose sniffer packet any "host 33.1.1.13" 4

interfaces=[any]
filters=[host 33.1.1.13]
2.592392 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592394 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592400 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592420 vd2-1_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592432 vd3-1_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592441 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592442 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592447 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592484 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
2.592485 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
2.592491 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
2.592498 vd3-1_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
2.592510 vd2-1_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
2.592515 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
2.592516 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
2.592520 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

8.808792 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808793 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808799 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808816 vd2-2_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808827 vd3-2_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808838 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808838 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808842 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808852 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.808853 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.808858 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.808866 vd3-2_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.808877 vd2-2_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.808882 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.808883 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.808887 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

```

```

18.024377 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024379 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024385 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024400 vd2-1_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024411 vd3-1_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024421 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024422 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024427 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024436 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
18.024437 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
18.024443 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
18.024449 vd3-1_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
18.024459 vd2-1_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
18.024463 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
18.024464 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
18.024468 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

24.216469 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216470 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216477 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216493 vd2-2_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216506 vd3-2_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216518 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216519 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216525 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216535 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.216536 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.216542 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.216548 vd3-2_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.216559 vd2-2_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.216563 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.216564 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.216568 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
^C
70 packets received by filter
0 packets dropped by kernel

```

## 7. Check SD-WAN health.

When an ADVPN shortcut is out of SLA, traffic does not run on it. Shortcut vd2-1\_0 is out of SLA.

```

# diagnose system sdwan health-check
Health Check(ping):
Seq(1 vd2-1): state(alive), packet-loss(6.000%) latency(0.026), jitter(0.001), mos
(4.401), bandwidth-up(1999), bandwidth-dw(0), bandwidth-bi(1999) sla_map=0x0
Seq(1 vd2-1_0): state(alive), packet-loss(18.182%) latency(0.033), jitter(0.003), mos
(4.395), bandwidth-up(2000), bandwidth-dw(0), bandwidth-bi(2000) sla_map=0x0
Seq(2 vd2-2): state(alive), packet-loss(0.000%) latency(0.024), jitter(0.001), mos
(4.404), bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x3
Seq(2 vd2-2_0): state(alive), packet-loss(0.000%) latency(0.033), jitter(0.005), mos
(4.404), bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x3

```

## 8. Check the sniffer packet:

No traffic runs on Shortcut vd2-1\_0 because it is out of SLA.

```

# diagnose sniffer packet any "host 33.1.1.13" 4
interfaces=[any]

```

```
filters=[host 33.1.1.13]
8.723075 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723077 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723084 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723103 vd2-2_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723115 vd3-2_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723148 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723149 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723154 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723166 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.723166 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.723171 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.723179 vd3-2_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.723190 vd2-2_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.723195 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.723195 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.723199 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

17.202681 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202683 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202688 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202704 vd2-2_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202716 vd3-2_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202727 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202728 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202733 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202742 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
17.202743 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
17.202749 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
17.202755 vd3-2_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
17.202767 vd2-2_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
17.202771 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
17.202772 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
17.202777 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
```

## Use SD-WAN rules to steer multicast traffic

SD-WAN rules can now steer multicast traffic. When an SD-WAN member is out of SLA, multicast traffic can fail over to another SD-WAN member, and switch back when SLA recovers.

The new `pim-use-sdwan` option enables or disables the use of SD-WAN for PIM (Protocol Independent Multicast) when checking RP (Rendezvous Point) neighbors and sending packets.

```
config router multicast
    config pim-sm-global
        set pim-use-sdwan {enable | disable}
    end
end
```

When SD-WAN steers multicast traffic, ADVPN is not supported. Use the `set shortcut` option to disable shortcuts for the service:

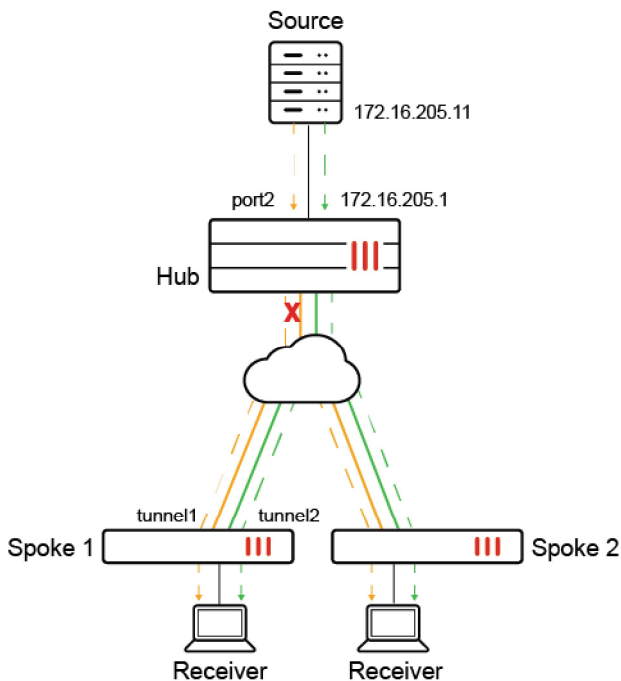


```
config system sdwan
  config service
    edit <id>
      set shortcut {enable | disable}
    next
  end
end
```

## Example 1

In this hub and spoke example, the PIM source is behind the hub FortiGate, and the RP is set to internal port (port2) of the hub firewall. Each spoke connects to the two WAN interfaces on the hub by using an overlay tunnel. The overlay tunnels are members of SD-WAN.

Receivers behind the spoke FortiGates request a stream from the source to receive traffic on tunnel1 by default. When the overlay tunnel goes out of SLA, the multicast traffic fails over to tunnel2 and continues to flow.



Following is an overview of how to configure the topology:

1. Configure the hub FortiGate in front of the PIM source. The RP is configured on internal port (port2) of the hub FortiGate.
2. Configure the spoke FortiGates.
3. Verify traffic failover.

**To configure the hub:**

1. On the hub, enable multicast routing, configure the multicast RP, and enable PIM sparse mode on each interface:

```
config router multicast
  set multicast-routing enable
  config pim-sm-global
    config rp-address
      edit 1
        set ip-address 172.16.205.1
      next
    end
  end
end
config interface
  edit "tport1"
    set pim-mode sparse-mode
  next
  edit "tagg1"
    set pim-mode sparse-mode
  next
  edit "port2"
    set pim-mode sparse-mode
  next
end
end
```

**To configure each spoke:**

1. Enable SD-WAN with the following settings:
  - Configure the overlay tunnels as member of the SD-WAN zone.
  - Configure a performance SLA health-check using ping.
  - Configure a service rule for the PIM protocol with the following settings:
    - Use the lowest cost (SLA) strategy.
    - Monitor with the ping health-check.
  - Disable ADVPN shortcut.

```
config system sdwan
  set status enable
  config zone
    edit "virtual-wan-link"
  next
end
config members
  edit 1
    set interface "tunnel1"
  next
  edit 2
    set interface "tunnel2"
  next
end
config health-check
  edit "ping"
    set server "172.16.205.1"
    set update-static-route disable
```

```

        set members 0
        config sla
            edit 1
                next
            end
        next
    end
end
config service
    edit 1
        set mode sla
        set protocol 103
        set dst "all"
        config sla
            edit "ping"
                set id 1
            next
        end
        set priority-members 1 2
        set use-shortcut-sla disable
        set shortcut disable
    next
    edit 2
        set mode sla
        set dst "all"
        config sla
            edit "ping"
                set id 1
            next
        end
        set priority-members 1 2
    next
end
end
end

```

## 2. Enable multicast routing and configure the multicast RP. Enable PIM sparse-mode on each interface:

```

config router multicast
    set multicast-routing enable
config pim-sm-global
    set spt-threshold disable
    set pim-use-sdwan enable
config rp-address
    edit 1
        set ip-address 172.16.205.1
    next
end
end
config interface
    edit "tunnell1"
        set pim-mode sparse-mode
    next
    edit "tunnel2"
        set pim-mode sparse-mode
    next
    edit "port4"
        set pim-mode sparse-mode
    next

```



```

    end
end

```

### To verify traffic failover:

With this configuration, multicast traffic starts on tunnel1. When tunnel1 becomes out of SLA, traffic switches to tunnel2. When tunnel1 is in SLA again, the traffic switches back to tunnel1.

The following health-check capture on the spokes shows tunnel1 in SLA with packet-loss (1.000%):

```

# diagnose sys sdwan health-check
Health Check(ping):
Seq(1 tunnel1): state(alive), packet-loss(0.000%) latency(0.056), jitter(0.002), mos(4.404),
bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_map=0x1
Seq(2 tunnel2): state(alive), packet-loss(0.000%) latency(0.100), jitter(0.002), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x1

```

```

# diagnose sys sdwan health-check
Health Check(ping):
Seq(1 tunnel1): state(alive), packet-loss(1.000%) latency(0.056), jitter(0.002), mos(4.404),
bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_map=0x1
Seq(2 tunnel2): state(alive), packet-loss(0.000%) latency(0.100), jitter(0.002), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x1

```

The following example shows tunnel1 out of SLA with packet-loss (3.000%):

```

# diagnose sys sdwan health-check
Health Check(ping):
Seq(1 tunnel1): state(alive), packet-loss(3.000%) latency(0.057), jitter(0.003), mos(4.403),
bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_map=0x0
Seq(2 tunnel2): state(alive), packet-loss(0.000%) latency(0.101), jitter(0.002), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x1

```

The following example shows tunnel1 back in SLA again:

```

# diagnose sys sdwan health-check
Health Check(ping):
Seq(1 tunnel1): state(alive), packet-loss(1.000%) latency(0.061), jitter(0.004), mos(4.404),
bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_map=0x0
Seq(2 tunnel2): state(alive), packet-loss(0.000%) latency(0.102), jitter(0.002), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x1

```

```

# diagnose sys sdwan health-check
Health Check(ping):
Seq(1 tunnel1): state(alive), packet-loss(0.000%) latency(0.061), jitter(0.004), mos(4.404),
bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_map=0x0
Seq(2 tunnel2): state(alive), packet-loss(0.000%) latency(0.102), jitter(0.002), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x1

```

The following example how traffic switches to tunnel2 while tunnel1 health-check is out of SLA. Source (172.16.205.11) sends traffic to the multicast group. Later the traffic switches back to tunnel1 once SLA returns to normal:

```

195.060797 tunnel1 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
195.060805 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
196.060744 tunnel1 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
196.060752 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request

```

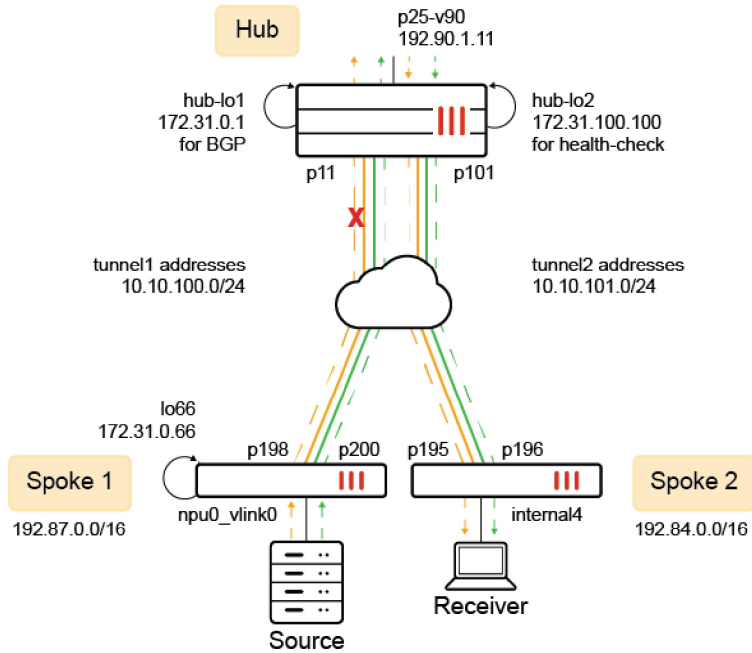
```
197.060728 tunnel1 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
197.060740 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
198.060720 tunnel2 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
198.060736 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
199.060647 tunnel2 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
199.060655 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
200.060598 tunnel2 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
200.060604 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
... ..
... ..
264.060974 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
265.060950 tunnel2 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
265.060958 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
266.060867 tunnel2 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
266.060877 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
267.060828 tunnel2 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
267.060835 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
268.060836 tunnel1 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
268.060854 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
269.060757 tunnel1 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
269.060767 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
270.060645 tunnel1 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
270.060653 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
```

## Example 2

In this hub and spoke example, the PIM source is behind spoke 1, and the RP is configured on the hub FortiGate. BGP is used for routing. The hub uses embedded SLA in ICMP probes to determine the health of each tunnel, allowing it to prioritize healthy IKE routes.

The receiver is on another spoke. Upon requesting a stream, source passes the traffic to the RP on the hub FortiGate, and routes the traffic to the receiver over tunnel1. If a tunnel falls out of SLA, the multicast traffic fails over to the other tunnel.

In this configuration, SD-WAN steers multicast traffic by using embedded SLA information in ICMP probes. See also [Embedded SD-WAN SLA information in ICMP probes](#). With this feature, the hub FortiGate can use the SLA information of the spoke's health-check to control BGP and IKE routes over tunnels.



Following is an overview of how to configure the topology:

1. Configure the hub FortiGate. The RP is configured on the hub FortiGate.
2. Configure the spoke FortiGate in front of the traffic receiver.
3. Configure the spoke FortiGate in front of the PIM source.

### To configure the hub:

1. Configure loopbacks hub-lo1 172.31.0.1 for BGP and hub-lo100 172.31.100.100 for health-check:

```
config system interface
  edit "hub-lo1"
    set vdom "hub"
    set ip 172.31.0.1 255.255.255.255
    set allowaccess ping
    set type loopback
    set snmp-index 82
  next
  edit "hub-lo100"
    set vdom "hub"
    set ip 172.31.100.100 255.255.255.255
    set allowaccess ping
    set type loopback
    set snmp-index 81
  next
end
```

2. Enable multicast routing with the following settings:

- Configure internal interface p25-v90 as RP.
- Enable interfaces for PIM sparse-mode.

```
config router multicast
  set multicast-routing enable
```

```
config pim-sm-global
  config rp-address
    edit 1
      set ip-address 192.90.1.11
    next
  end
end
config interface
  edit "p11"
    set pim-mode sparse-mode
  next
  edit "p101"
    set pim-mode sparse-mode
  next
  edit "p25-v90"
    set pim-mode sparse-mode
  next
end
end
```

**3. Enable SD-WAN with the following settings:**

- Add interfaces p11 and p101 as members.
- Configure embedded SLA health-checks to detect ICMP probes from each overlay tunnel. Prioritize based on the health of each tunnel.

```
config system sdwan
  set status enable
  config zone
    edit "virtual-wan-link"
    next
  end
  config members
    edit 1
      set interface "p11"
    next
    edit 2
      set interface "p101"
    next
  end
  config health-check
    edit "1"
      set detect-mode remote
      set probe-timeout 60000
      set recoverytime 1
      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
  edit "2"
```

```
        set detect-mode remote
        set probe-timeout 60000
        set recoverytime 1
        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
end
```

**4. Configure BGP to peer with neighbors. Neighbor group is configured for tunnel interface IP addresses:**

```
config router bgp
    set as 65505
    set router-id 172.31.0.1
    set ibgp-multipath enable
    set additional-path enable
    set recursive-inherit-priority enable
    config neighbor-group
        edit "gr1"
            set remote-as 65505
            set update-source "hub-lo1"
            set additional-path both
            set route-reflector-client enable
        next
    end
    config neighbor-range
        edit 1
            set prefix 10.10.0.0 255.255.0.0
            set neighbor-group "gr1"
        next
        edit 66
            set prefix 172.31.0.66 255.255.255.255
            set neighbor-group "gr1"
        next
    end
    config network
        ....
        edit 90
            set prefix 192.90.0.0 255.255.0.0
        next
    end
end
```

**To configure the spoke (in front of the receiver):**

1. Enable multicast routing to use SD-WAN. Configure the RP address. Enable interfaces for PIM sparse-mode.

```
config router multicast
  set multicast-routing enable
  config pim-sm-global
    set spt-threshold disable
    set pim-use-sdwan enable
  config rp-address
    edit 1
      set ip-address 192.90.1.11
    next
  end
end
config interface
  edit "p195"
    set pim-mode sparse-mode
  next
  edit "p196"
    set pim-mode sparse-mode
  next
  edit "internal4"
    set pim-mode sparse-mode
    set static-group "225-1-1-122"
  next
end
end
```

2. Configure SD-WAN with the following settings:

- Add overlay tunnel interfaces as members.
- Configure a performance SLA health-check to send ping probes to the hub.
- Configure a service rule for the PIM protocol. Use the lowest cost (SLA) strategy, and monitor with the ping health-check.
- Disable ADVPN shortcuts.

```
config system sdwan
  set status enable
  config zone
    edit "virtual-wan-link"
    next
  end
  config members
    edit 6
      set interface "p196"
    next
    edit 5
      set interface "p195"
    next
  end
```

```
config health-check
  edit "ping"
    set server "172.31.100.100"
    set update-static-route disable
    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 protocol 103
    set dst "all"
    config sla
      edit "ping"
        set id 1
      next
    end
    set priority-members 5 6
    set use-shortcut-sla disable
    set shortcut disable
  next
  edit 2
    set mode sla
    set dst "all"
    config sla
      edit "ping"
        set id 1
      next
    end
    set priority-members 5 6
  next
end
end
```

### 3. Configure BGP and set neighbors to the overlay gateway IP address on the hub:

```
config router bgp
  set as 65505
  set router-id 122.1.1.122
  set ibgp-multipath enable
  set additional-path enable
  config neighbor
    edit "10.10.100.254"
      set soft-reconfiguration enable
```

```

        set remote-as 65505
        set connect-timer 10
        set additional-path both
    next
    edit "10.10.101.254"
        set soft-reconfiguration enable
        set remote-as 65505
        set connect-timer 10
        set additional-path both
    next
end
config network
    edit 3
        set prefix 192.84.0.0 255.255.0.0
    next
end
end

```

4. Configure the default gateway to use the SD-WAN zone. Other routes are for the underlay to route traffic to the hub's WAN interfaces:

```

config router static
    edit 10
        set distance 1
        set sdwan-zone "virtual-wan-link"
    next
    ....
    next
end

```

### To configure the spoke (in front of the source):

1. Enable multicast routing to use SD-WAN. Configure the RP address. Enable interfaces for PIM sparse-mode:

```

config router multicast
    set multicast-routing enable
config pim-sm-global
    set pim-use-sdwan enable
    config rp-address
        edit 1
            set ip-address 192.90.1.11
        next
    end
end
config interface
    edit "p198"
        set pim-mode sparse-mode
    next
    edit "p200"
        set pim-mode sparse-mode
    next
    edit "npu0_vlink0"
        set pim-mode sparse-mode

```



```

    next
  end
end

```

## 2. Configure loopback interface lo66 for BGP and sourcing SD-WAN traffic:

```

config system interface
  edit "lo66"
    set vdom "root"
    set ip 172.31.0.66 255.255.255.255
    set allowaccess ping
    set type loopback
    set snmp-index 21
  next
end

```

## 3. Configure SD-WAN:

- Add overlay tunnel interfaces as members.
- Configure a performance SLA health-check to send ping probes to the hub.
- Configure a service rule for the PIM protocol. Use the lowest cost (SLA) strategy, and monitor with the ping health-check.
- Disable the use of an ADVPN shortcut.

In the following example, 11.11.11.11 is the underlay address for one of the WAN links on the hub, and 172.31.100.100 is the loopback address on the server.

```

config system sdwan
  set status enable
  config zone
    edit "virtual-wan-link"
    next
    edit "overlay"
    next
  end
  config members
    edit 1
      set interface "p198"
      set zone "overlay"
      set source 172.31.0.66
    next
    edit 2
      set interface "p200"
      set zone "overlay"
      set source 172.31.0.66
    next
  end
  config health-check
    edit "ping"
      set server "11.11.11.11"
      set members 0
      config sla
        edit 1
          set link-cost-factor latency
          set latency-threshold 100
        next
      end
    next
  end

```

```
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 protocol 103  
        set dst "all"  
        config sla  
            edit "ping"  
                set id 1  
            next  
        end  
        set priority-members 1 2  
        set use-shortcut-sla disable  
        set shortcut disable  
    next  
    edit 2  
        set mode sla  
        set dst "all"  
        config sla  
            edit "ping"  
                set id 1  
            next  
        end  
        set priority-members 1 2  
    next  
end  
end
```

#### 4. Configure BGP:

```
config router bgp  
    set as 65505  
    set router-id 123.1.1.123  
    set ibgp-multipath enable  
    set additional-path enable  
    config neighbor  
        edit "172.31.0.1"  
            set next-hop-self enable  
            set soft-reconfiguration enable  
            set remote-as 65505  
            set update-source "lo66"  
        next  
    end  
    config network  
        edit 3  
            set prefix 192.87.0.0 255.255.0.0
```

```

    next
  end
end

```

5. Configure the default gateway to use the SD-WAN zone. Other routes are for the underlay to route to the hub's WAN interfaces:

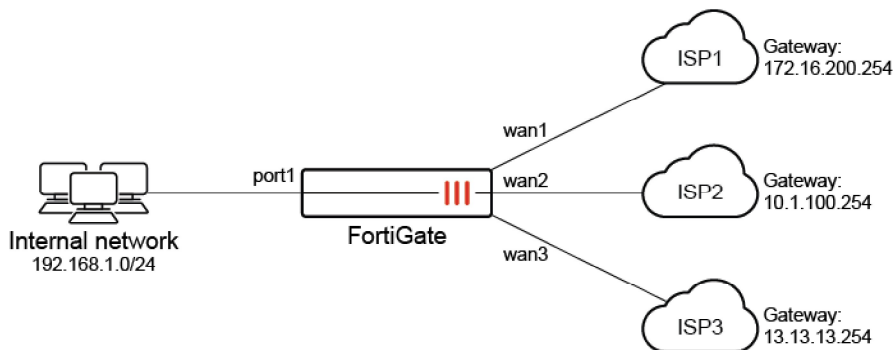
```

config router static
  edit 10
    set distance 1
    set sdwan-zone "virtual-wan-link" "overlay"
  next
  ...
next
end

```

## Use SD-WAN rules for WAN link selection with load balancing

This example covers a use case where a user has multiple WAN links and wants to optimize the WAN link selection and performance while limiting the use of more expensive and bandwidth intensive interfaces, such as 5G or LTE.



In this scenario, the user has three WAN links. The goal is to balance the load between wan1 and wan2; however, wan3, which is quite costly to operate, should only be used if both wan1 and wan2 are unavailable.

**This configuration involves the following steps:**

1. [Configuring the SD-WAN members](#)
2. [Configuring the manual SD-WAN rule](#)
3. [Configuring a static route](#)
4. [Configuring a firewall policy for SD-WAN](#)
5. [Verifying the configuration](#)

## Configuring the SD-WAN members

SD-WAN must be enabled first, and member interfaces must be selected and added to a zone. See [Configuring the SD-WAN interface on page 786](#) for more information.

**To configure the SD-WAN members in the GUI:**

1. Configure the wan1, wan2, and wan3 interfaces (see [Interface settings on page 164](#) for more details).
  - a. Set the wan1 interface *IP/Netmask* to *172.16.200.1 255.255.255.0*.
  - b. Set the wan2 interface *IP/Netmask* to *10.1.100.1 255.255.255.0*.
  - c. Set the wan3 interface *IP/Netmask* to *13.13.13.1 255.255.255.0*.
2. Go to *Network > SD-WAN*, select the *SD-WAN Zones* tab, and click *Create New > SD-WAN Member*.
3. Configure the wan1 SD-WAN member:
  - a. Set the *Interface* to *wan1*.
  - b. Leave the *SD-WAN Zone* as *virtual-wan-link*.
  - c. Set the *Gateway* to *172.16.200.254*.
  - d. Set the *Status* to *Enable*
  - e. Click *OK*.
4. Repeat step 3 for wan2 and wan3.
  - a. For wan2, set the *Gateway* to the ISP's gateway, *10.1.100.254*.
  - b. For wan3, set the *Gateway* to the ISP's gateway, *13.13.13.254*.

**To configure the SD-WAN members in the CLI:**

```

config system sdwan
  set status enable
  config zone
    edit "virtual-wan-link"
    next
  end
  config members
    edit 1
      set interface "wan1"
      set gateway 172.16.200.254
    next
    edit 2
      set interface "wan2"
      set gateway 10.1.100.254
    next
    edit 3
      set interface "wan3"
      set gateway 13.13.13.254
    next
  end
end

```

**Configuring the manual SD-WAN rule**

SD-WAN rules define specific routing options to route traffic to an SD-WAN member. See [SD-WAN rules on page 851](#) and [Manual strategy on page 864](#) for more information.

**To configure a manual SD-WAN rule in the GUI:**

1. Go to *Network > SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
2. Configure the following settings:

|                                     |                      |
|-------------------------------------|----------------------|
| <b>Name</b>                         | <i>test</i>          |
| <b>Source &gt; Address</b>          | <i>all</i>           |
| <b>Destination &gt; Address</b>     | <i>all</i>           |
| <b>Interface selection strategy</b> | <i>Manual</i>        |
| <b>Interface preference</b>         | <i>wan1, wan2</i>    |
| <b>Load balancing</b>               | Enable this setting. |

3. Configure the other settings as needed.
4. Click *OK*.

#### To configure a manual SD-WAN rule in the CLI:

```
config system sdwan
  config service
    edit 1
      set name "test"
      set load-balance enable
      set dst "all"
      set src "all"
      set priority-members 1 2
    next
  end
end
```

## Configuring a static route

A default route for SD-WAN must be configured. See [Adding a static route on page 788](#) for more information.

#### To configure a static route for SD-WAN in the GUI:

1. Go to *Network > Static Routes* and click *Create New*. The *New Static Route* page opens.
2. Set the *Destination* to *Subnet*, and leave the IP address and subnet mask as *0.0.0.0/0.0.0.0*.
3. Set the *Interface* to the SD-WAN zone, *virtual-wan-link*.
4. Set the *Status* to *Enabled*.
5. Click *OK*.

#### To configure a static route for SD-WAN in the CLI:

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

## Configuring a firewall policy for SD-WAN

A firewall policy must be configured that allows traffic from the organization's internal network to the SD-WAN zone. See [Configuring firewall policies for SD-WAN on page 789](#) for more information.

### To configure the firewall policy for SD-WAN in the GUI:

1. Go to *Policy & Objects > Firewall Policy* and click *Create New*.
2. Configure the following settings:

|                              |                                       |
|------------------------------|---------------------------------------|
| <b>Name</b>                  | <i>sd-wan</i>                         |
| <b>Incoming interface</b>    | <i>port1</i>                          |
| <b>Outgoing interface</b>    | <i>virtual-wan-link</i>               |
| <b>Source</b>                | <i>all</i>                            |
| <b>Destination</b>           | <i>all</i>                            |
| <b>Schedule</b>              | <i>always</i>                         |
| <b>Service</b>               | <i>ALL</i>                            |
| <b>Action</b>                | <i>ACCEPT</i>                         |
| <b>NAT</b>                   | Enable and select <i>NAT</i> .        |
| <b>IP Pool Configuration</b> | <i>Use Outgoing Interface Address</i> |
| <b>Enable this policy</b>    | Enable this setting.                  |

3. Configure the other settings as needed.
4. Click *OK*.

### To configure the firewall policy for SD-WAN in the CLI:

```
config firewall policy
  edit 1
    set name "sd-wan"
    set srcintf "port1"
    set dstintf "virtual-wan-link"
    set action accept
    set srcaddr "all"
    set dstaddr "all"
    set schedule "always"
    set service "ALL"
    set nat enable
  next
end
```

## Verifying the configuration

### To verify the SD-WAN member status:

```
# diagnose sys sdwan member
Member(1): interface: wan1, flags=0x0 , gateway: 172.16.200.254, priority: 1 1024, weight: 0
Member(2): interface: wan2, flags=0x0 , gateway: 10.1.100.254, priority: 1 1024, weight: 0
Member(3): interface: wan3, flags=0x0 , gateway: 13.13.13.254, priority: 1 1024, weight: 0
```

### To verify the configuration when both wan1 and wan2 are up:

#### 1. Verify the SD-WAN service rules status:

```
# diagnose sys sdwan service4
Service(1): Address Mode(IPV4) flags=0x24200 use-shortcut-sla use-shortcut
Tie break: cfg
Gen(1), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(manual hash-
mode=round-robin)
Members(2):
  1: Seq_num(2 wan2 virtual-wan-link), alive, gid(1), selected
  2: Seq_num(1 wan1 virtual-wan-link), alive, gid(1), selected
Src address(1):
  0.0.0.0-255.255.255.255

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

This output indicates that both wan1 and wan2 are operational.

#### 2. Verify the policy route list:

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

id=2130706433(0x7f000001) vwl_service=1(test) vwl_mbr_seq=1 2 dscp_tag=0xfc 0xfc
flags=0x10 load-balance hash-mode=round-robin tos=0x00 tos_mask=0x00 protocol=0
port=src(0->0):dst(0->0) iif=0(any)
path(2): oif=3(wan1) num_pass=0, oif=6(wan2) num_pass=0
source(1): 0.0.0.0-255.255.255.255
destination(1): 0.0.0.0-255.255.255.255
hit_count=154 last_used=2023-11-09 06:16:
```

This output indicates that both wan1 and wan2 are used to steer traffic.

### To verify the configuration when wan2 is down and wan1 is up:

#### 1. Verify the SD-WAN service rules status:

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x24200 use-shortcut-sla use-shortcut
Tie break: cfg
Gen(1), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(manual hash-
mode=round-robin)
Members(2):
  1: Seq_num(1 wan1 virtual-wan-link), alive, gid(1), selected
  2: Seq_num(2 wan2 virtual-wan-link), dead, gid(1)
```

```

Src address(1):
    0.0.0.0-255.255.255.255

Dst address(1):
    0.0.0.0-255.255.255.255

```

This output indicates that wan1 is operational, and wan2 is not.

## 2. Verify the policy route list:

```

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

id=2130706433(0x7f000001) vwl_service=1(test) vwl_mbr_seq=1 dscp_tag=0xfc 0xfc
flags=0x10 load-balance hash-mode=round-robin tos=0x00 tos_mask=0x00 protocol=0
port=src(0->0):dst(0->0) iif=0(any)
path(1): oif=3(wan1) num_pass=0
source(1): 0.0.0.0-255.255.255.255
destination(1): 0.0.0.0-255.255.255.255
hit_count=482 last_used=2023-11-09 06:27:08

```

This output indicates that wan1 is used to steer traffic.

## To verify the configuration when wan1 is down and wan2 is up:

### 1. Verify the SD-WAN service rules status:

```

# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x24200 use-shortcut-sla use-shortcut
Tie break: cfg
Gen(1), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(manual hash-
mode=round-robin)
Members(2):
    1: Seq_num(2 wan2 virtual-wan-link), alive, gid(1), selected
    2: Seq_num(1 wan1 virtual-wan-link), dead, gid(1)
Src address(1):
    0.0.0.0-255.255.255.255

Dst address(1):
    0.0.0.0-255.255.255.255

```

This output indicates that wan2 is operational, and wan1 is not.

### 2. Verify the policy route list:

```

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

id=2130706433(0x7f000001) vwl_service=1(test) vwl_mbr_seq=2 dscp_tag=0xfc 0xfc
flags=0x10 load-balance has
h-mode=round-robin tos=0x00 tos_mask=0x00 protocol=0 port=src(0->0):dst(0->0) iif=0
(any)
path(1): oif=6(wan2) num_pass=0
source(1): 0.0.0.0-255.255.255.255
destination(1): 0.0.0.0-255.255.255.255
hit_count=903 last_used=2023-11-09 06:41:55

```

This output indicates that wan2 is used to steer traffic.



**To verify the configuration when both wan1 and wan2 down, and traffic is steered using wan3:**

```
# diagnose sniffer packet wan3
Using Original Sniffing Mode
interfaces=[wan3]
filters=[none]
3.144417 13.13.13.1.52665 -> 204.79.197.239.443: 1610731732 ack 236747780
3.155250 204.79.197.239.443 -> 13.13.13.1.52665: ack 1610731733
5.047264 13.13.13.1.52613 -> 20.185.212.106.443: 1421254032 ack 3784884456
5.126008 20.185.212.106.443 -> 13.13.13.1.52613: ack 1421254033
```

This output indicates that wan3 is used to steer traffic.

**To verify the configuration when either wan1 or wan2 is restored, and traffic ceases to be steered through wan3:****1. Verify the SD-WAN service rules status:**

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x24200 use-shortcut-sla use-shortcut
Tie break: cfg
Gen(1), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(manual hash-
mode=round-robin)
Members(2):
  1: Seq_num(1 wan1 virtual-wan-link), alive, gid(1), selected
  2: Seq_num(2 wan2 virtual-wan-link), dead, gid(1)
Src address(1):
  0.0.0.0-255.255.255.255

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

This output indicates that wan1 is operational.

**2. Verify the policy route list:**

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

id=2130706433(0x7f000001) vwl_service=1(test) vwl_mbr_seq=1 dscp_tag=0xfc 0xfc
flags=0x10 load-balance has
h-mode=round-robin tos=0x00 tos_mask=0x00 protocol=0 port=src(0->0):dst(0->0) iif=0
(any)
path(1): oif=3(wan1) num_pass=0
source(1): 0.0.0.0-255.255.255.255
destination(1): 0.0.0.0-255.255.255.255
hit_count=986 last_used=2023-11-09 06:45:13
```

This output indicates that wan1 is used to steer traffic.

## Advanced routing

The following topics provide instructions on SD-WAN advanced routing: