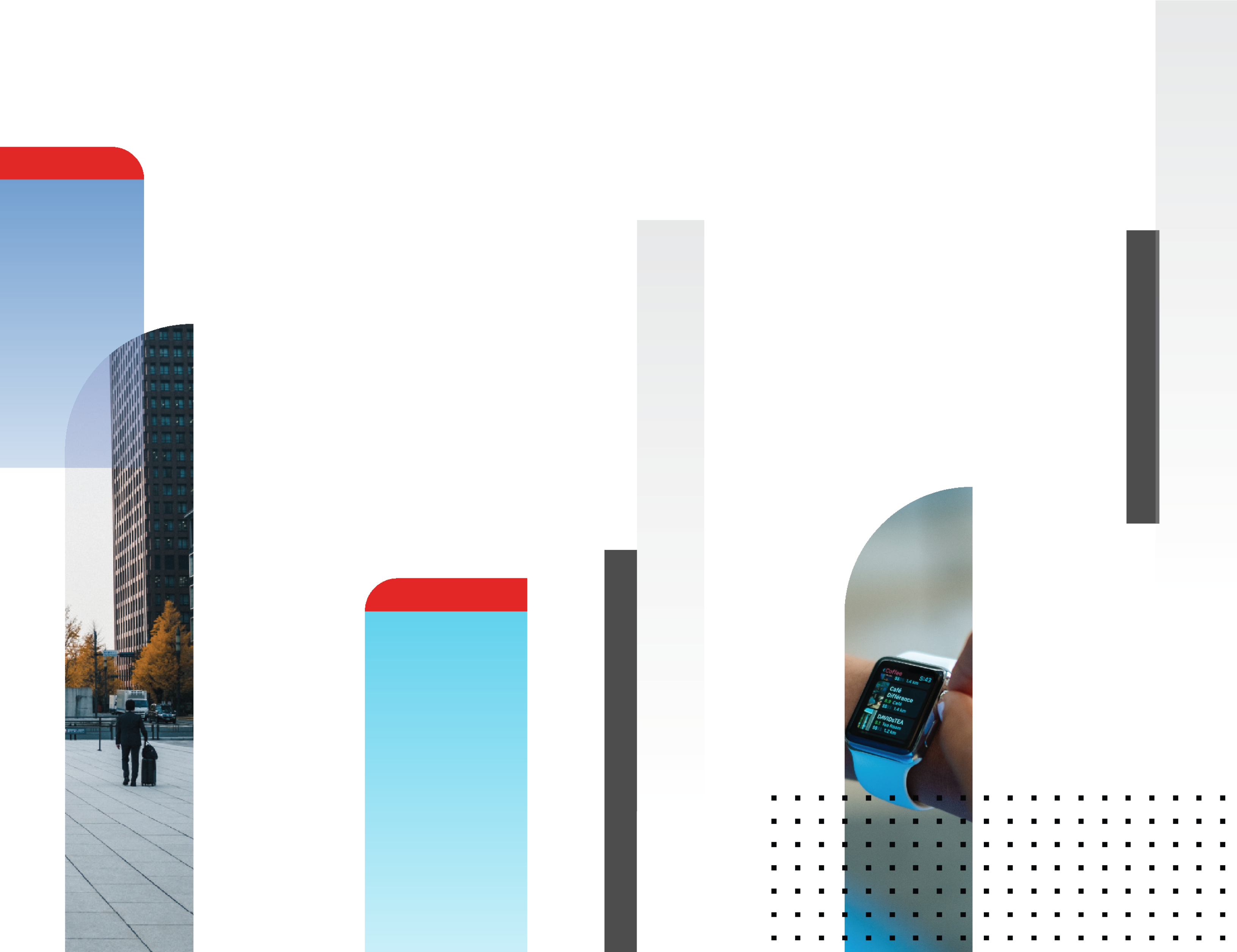
SD-WAN self-healing with BGP

FortiOS 7.0



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

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

FORTINET BLOG
https://blog.fortinet.com

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

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

NSE INSTITUTE
https://training.fortinet.com

FORTIGUARD CENTER
https://www.fortiguard.com

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

FEEDBACK
Email: techdoc@fortinet.com

November 05, 2021
FortiOS 7.0 SD-WAN self-healing with BGP
01-700-721860-20211105

TABLE OF CONTENTS

Change Log 4
Overview 5
Hub 5
Branches 5
VPN configurations 7
Routing configurations for traffic from the hub to the branches 10
Hub configuration 10
Branch configuration 12
SD-WAN configurations for steering traffic from the hub to the branches 15
Configure SD-WAN members 15
Configure performance SLA health checks 16
Configure SD-WAN neighbors 16
Configure service rules 17
Firewall policies 19
Routing configurations for traffic from the branches to the hub 20
Testing and verification 21
Traffic flow during normal operation 21
Traffic flow when Branch1 ISP1 is impaired 25
Traffic flow when Branch1 ISP1 and ISP2 are impaired 28

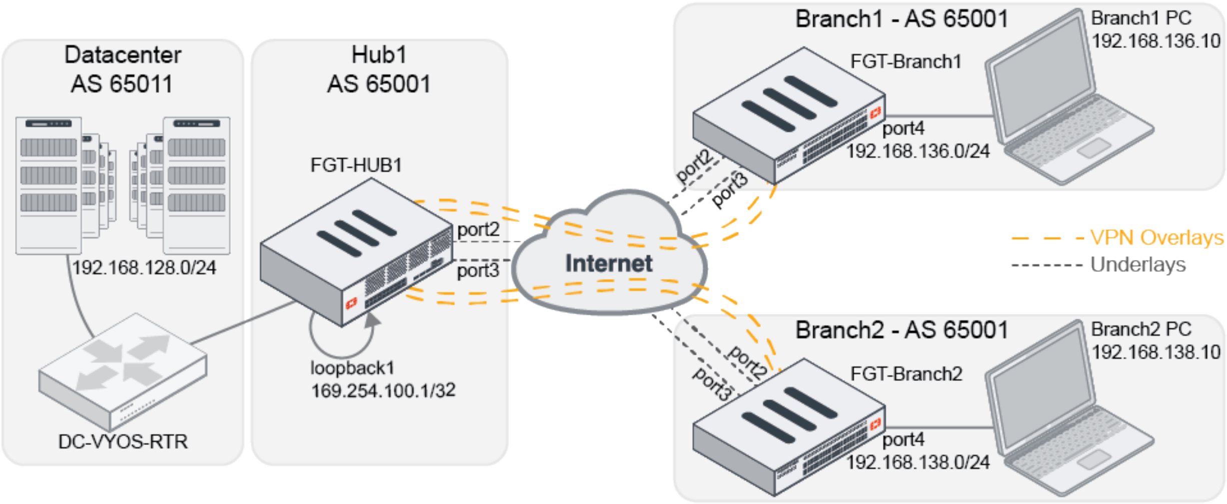
Change Log

Date	Change Description
2021-11-05	Initial release.

Overview

This example demonstrates a scalable configuration using options that help simplify head-end traffic-steering in an SD-WAN setup that uses a hub and spoke topology. In this example, the hub and branches have basic configurations, with one set of SD-WAN rules on the hub to cover all branch instances.

The hub does not need to reference branch addresses in the SD-WAN rules to steer traffic to each branch over the healthy VPN overlay. It also does not need to run health checks to the branches to determine what paths are healthy. Instead, the branches configure health checks to monitor the links, and use BGP and BGP communities to satisfy both requirements by updating the hub with the status of the links over BGP. This avoids manual maintaining health checks from the head-end, allowing for better scalability.



Hub

The hub is connected to two ISPs, and forms VPN overlays with each branch over two interfaces. When it learns the routes from the branches, it matches the BGP communities and assigns route-tags to them. Using SD-WAN service rules it steers traffic to a branches subnet based on the active route-tag. It does not require health checks to determine the healthy path because this is handled by each branch.

Branches

Each branch is connected to two ISPs, and for each ISP a VPN overlay connects to the hub. The branch monitors the health status of its VPN overlays. It advertises the branch subnet over iBGP to its peer on the hub that is acting as a route reflector. Each branch selectively advertises its preferred overlay using route-maps that tag the advertised routes with BGP communities. When the primary link is out of SLA, it advertises a failed BGP community, and the secondary link advertises its normal BGP community.

There are two branches in this example, but the solution can be scaled up with more branches that use the same configuration.

VPN configurations

Two ADVPN tunnels, VPN1 and VPN2, are created on the hub for the WAN interfaces. VPN1 assigns IP addresses from 169.254.16.10 to 169.254.16.250 and VPN2 assigns IP addresses from 169.254.17.10 to 169.254.17.250. BGP neighbors are formed over the VPN overlays.

To configure the hub:

1. Configure the phase1 and phase2 settings for VPN1:

```
config vpn ipsec phase1-interface
    edit "VPN1"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes256-sha256
        set add-route disable
        set dpd on-idle
        set auto-discovery-sender enable
        set network-overlay enable
        set network-id 1
        set ipv4-start-ip 169.254.16.10
        set ipv4-end-ip 169.254.16.250
        set ipv4-netmask 255.255.255.0
        set psksecret <secret>
        set dpd-retryinterval 60
    next
end
```

```
config vpn ipsec phase2-interface
    edit "VPN1"
        set phase1name "VPN1"
        set proposal aes256-sha256
    next
end
```

2. Configure the phase1 and phase2 settings for VPN2:

```
config vpn ipsec phase1-interface
    edit "VPN2"
        set type dynamic
        set interface "port3"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes256-sha256
        set add-route disable
        set dpd on-idle
        set auto-discovery-sender enable
        set network-overlay enable
        set network-id 2
        set ipv4-start-ip 169.254.17.10
        set ipv4-end-ip 169.254.17.250
        set ipv4-netmask 255.255.255.0
        set psksecret <secret>
        set dpd-retryinterval 60
    next
end
```

```
config vpn ipsec phase2-interface
    edit "VPN2"
        set phase1name "VPN2"
        set proposal aes256-sha256
    next
end
```

To configure the branches:

1. Configure the phase1 and phase2 settings for HUB-VPN1:

```
config vpn ipsec phase1-interface
    edit "HUB1-VPN1"
        set interface "port2"
        set ike-version 2
        set peertype any
        set net-device enable
        set mode-cfg enable
        set proposal aes256-sha256
        set add-route disable
        set localid "FGT-Branch1_ISP1"
        set idle-timeout enable
        set auto-discovery-receiver enable
        set network-overlay enable
        set network-id 1
        set remote-gw 192.168.100.2
        set psksecret <secret>
        set dpd-retrycount 2
        set dpd-retryinterval 2
    next
end
```

```
config vpn ipsec phase2-interface
    edit "HUB1-VPN1"
        set phase1name "HUB1-VPN1"
        set proposal aes256-sha256
        set auto-negotiate enable
    next
end
```

2. Configure the phase1 and phase2 settings for HUB-VPN2:

```
config vpn ipsec phase1-interface
    edit "HUB1-VPN2"
        set interface "port3"
        set ike-version 2
        set peertype any
```

```
            set net-device enable
            set mode-cfg enable
            set proposal aes256-sha256
            set add-route disable
            set localid "FGT-Branch1_ISP2"
            set idle-timeout enable
            set auto-discovery-receiver enable
            set network-overlay enable
            set network-id 2
            set remote-gw 192.168.101.2
            set psksecret <secret>
            set dpd-retrycount 2
            set dpd-retryinterval 2
        next
    end
```

```
    config vpn ipsec phase2-interface
        edit "HUB1-VPN2"
            set phase1name "HUB1-VPN2"
            set proposal aes256-sha256
            set auto-negotiate enable
        next
    end
```

3. Repeat the configuration on all of the other branches.

Routing configurations for traffic from the hub to the branches

Dynamic routing over BGP is configured to automatically advertise each branch network to the hub based on the health status of each VPN overlay. Health checks are configured in SD-WAN configurations for steering traffic from the hub to the branches on page 15.

The hub learns the good and bad VPN overlays using the advertised community strings for each prefix. It then tags the prefixes with the route-tags that are used for steering by the SD-WAN. Traffic from the hub to the branches is steered by SD-WAN to the healthy links without needing any health checks to be configured on the hub.

Hub configuration

The hub forms neighborship with each branch over the VPN overlays. Based on the routes that it learns from each overlay on each branch, the hub tags the prefix with a route-tag based on its community string.

To configure the hub:

1. Configure the AS:

```
config router bgp
       set as 65001
end
```

2. For the datacenter connection, configure the eBGP neighbor. Routes learned from this neighbor are assigned the community 65001:101.

```
config router route-map
    edit "BGP-TAG-HUB1"
        config rule
            edit 1
                set set-community "65001:101"
                unset set-ip-nexthop
                unset set-ip6-nexthop
                unset set-ip6-nexthop-local
                unset set-originator-id
            next
        end
    next
end
```

```
config neighbor
    edit "192.168.132.2"
        set description "HUB1-VYOS"
        set remote-as 65011
        set route-map-in "BGP-TAG-HUB1"
    next
end
```

3. Configure the community list with the following three communities:

Object	Community	Description
HUB1-VPN1	65001:1	Used to match VPN1 when under SLA.
HUB1-VPN2	65001:2	Used to match VPN2 when under SLA.
HUB1-SLA-Fail	65001:11	Used to match failed SLA.

```
config router community-list
    edit "HUB1-VPN1"
        config rule
            edit 1
                set action permit
                set match "65001:1"
            next
        end
    next
    edit "HUB1-VPN2"
        config rule
            edit 1
                set action permit
                set match "65001:2"
            next
        end
    next
    edit "HUB1-SLA-Fail"
        config rule
            edit 1
                set action permit
                set match "65001:11"
            next
        end
    next
end
```

4. Configure a new route map:

The hub assigns route-tags by matching the previously configured communities. The route tags are used in SD-WAN rules.

Route-tag	Match Community Object	Community String
1	HUB1-VPN1	65001:1
2	HUB1-VPN2	65001:2
11	HUB1-SLA-Fail	65001:11

```
config router route-map
    edit "BGP-Route-Tag"
        config rule
            edit 1
                set match-community "HUB1-VPN1"
                set set-route-tag 1
            next
            edit 2
```

```
                set match-community "HUB1-VPN2"
                set set-route-tag 2
            next
            edit 3
                set match-community "HUB1-SLA-Fail"
                set set-route-tag 11
            next
        end
    next
end
```

5. Create neighbor groups for remote AS 65001, and apply route-map-in on learned routes to add route-tags:

```
config router bgp
    config neighbor-group
        edit "VPN1"
            set remote-as 65001
            set route-map-in "BGP-Route-Tag"
         next
         edit "VPN2"
            set remote-as 65001
            set route-map-in "BGP-Route-Tag"
         next
    end
end
```

6. Apply the neighbor groups to the neighbor ranges.

```
config router bgp
    config neighbor-range
        edit 1
            set prefix 169.254.16.0 255.255.255.0
            set neighbor-group "VPN1"
        next
        edit 2
            set prefix 169.254.17.0 255.255.255.0
            set neighbor-group "VPN2"
        next
    end
end
```

Branch configuration

Each branch forms neighborship with the hub on each overlay. Each advertised route is tagged with the appropriate community string to reflect the health status of that link.

- When the health of VPN1 is good, it advertises community 65001:1.
- When the health of VPN2 is good, it advertises community 65001:2.
- When the health of either VPN is bad, it advertises community 65001:11.

To configure a branch:

1. Configure the initial BGP configurations. The neighbors are reachable through the VPN interfaces HUB1-VPN1 and HUB1-VPN2.

```
config router bgp
    set as 65001
    config neighbor
        edit "169.254.16.1"
            set description "HUB1-VPN1"
            set interface "HUB1-VPN1"
            set remote-as 65001
        next
        edit "169.254.17.1"
            set description "HUB1-VPN2"
            set interface "HUB1-VPN2"
            set remote-as 65001
        next
    end
end
```

2. Configure the route maps for setting the respective communities:

```
config router route-map
    edit "Primary_DC_Fail_BGP_Health_Check"
        config rule
            edit 1
                set set-community "65001:11"
            next
        end
    next
    edit "Primary_DC_Primary_VPN"
        config rule
            edit 1
                set set-community "65001:1"
            next
        end
    next
    edit "Primary_DC_Secondary_VPN"
        config rule
            edit 1
                set set-community "65001:2"
            next
        end
    next
end
```

3. Configure the preferred and default route maps. These are used with the SD-WAN configuration.

```
config router bgp
    config neighbor
        edit "169.254.16.1"
            set advertisement-interval 1
            set route-map-out "Primary_DC_Fail_BGP_Health_Check"
            set route-map-out-preferable "Primary_DC_Primary_VPN"
        next
        edit "169.254.17.1"
            set advertisement-interval 1
            set route-map-out "Primary_DC_Fail_BGP_Health_Check"
            set route-map-out-preferable "Primary_DC_Secondary_VPN"
        next
    end
end
```

When performance SLA health checks are under the threshold, the preferred route maps are used (`route-map-out-preferable`). When the health checks are over the threshold, the default route maps are used (`route-map-out`).

Setting the advertisement interval allows updates to be sent to the hub in one second.

4. Repeat the configuration on all of the other branches.

SD-WAN configurations for steering traffic from the hub to the branches

On each branch, a health check is configured to monitor the status of the loopback interface on the hub over HUB1-VPN1 and HUB1-VPN2. If either fails, BGP will advertise the failed community string 65001:11 to the hub.

No health checks are needed on the hub. Instead, SD-WAN service rules are configured to match each route-tag and steer traffic to the corresponding VPN overlay when the neighbor link is healthy. Because all branches advertise the same community strings and route-tags, only one set of service rules is required for all of the branches, and no additional settings are needed for additional branches, making this solution easily scalable.

Configure SD-WAN members

To configure the hub:

1. Put both VPNs into the overlay SD-WAN zone:

```
config system sdwan
    set status enable
    config zone
        edit "Overlay"
        next
    end
    config members
        edit 1
            set interface "VPN1"
            set zone "Overlay"
        next
        edit 2
            set interface "VPN2"
            set zone "Overlay"
        next
    end
end
```

To configure a branch:

1. Put both HUB1-VPN1 and HUB1-VPN2 into the OverlayHUB1 SD-WAN zone:

```
config system sdwan
    set status enable
    config zone
        edit “OverlayHUB1”
        next
    end
    config members
        edit 1
```

```
            set interface "HUB1-VPN1"
            set zone "OverlayHUB1"
            set comment "Mapping to HUB1 through ISP1"
        next
        edit 2
            set interface "HUB1-VPN2"
            set zone "OverlayHUB1"
            set comment "Mapping to HUB1 through ISP2"
        next
    end
end
```

The zones and underlays used for internal breakout are not shown here.

Configure performance SLA health checks

Performance SLA health checks are only configured on the branches. Each branch monitors the status of the loopback interface on the hub over HUB1-VPN1 and HUB1-VPN2. The loopback address, 169.254.100.1, is routable over both VPN interfaces.

To configure a branch:

```
config system sdwan
    config health-check
        edit "HUB1_HC"
            set server "169.254.100.1"
            set update-static-route disable
            set sla-fail-log-period 10
            set sla-pass-log-period 10
            set members 1 2
            config sla
                edit 1
                    set latency-threshold 150
                    set jitter-threshold 30
                    set packetloss-threshold 2
                next
            end
        next
    end
end
```

Configure SD-WAN neighbors

The SD-WAN neighbors are only configured on the branches. The neighbor settings allow the branch to attach a health check to individual BGP neighbors based on the neighbor ID.

With the previously configured BGP settings, when a health check is below the threshold, BGP applies preferred route maps (`route-map-out-preferable`) and tags the advertised routes with the community string that is associated with the VPN interface. When a health check is above the threshold, BGP applies the default route map (`route-map-out`) and tags the advertised routes with the community string that is associated with the failed threshold.

To configure a branch:

```
config system sdwan
    config neighbor
        edit "169.254.16.1"
            set member 1
            set health-check "HUB1_HC"
            set sla-id 1
        next
        edit "169.254.17.1"
            set member 2
            set health-check "HUB1_HC"
            set sla-id 1
        next
    end
end
```

Configure service rules

Service rules define how traffic is steered based on criteria such as the source and destination addresses, service, and route-tag. In this example, the direction of traffic is from the hub to the branches, so service rules must be defined on the hub to correctly steer the traffic to the branches over the healthy VPN overlay.

The hub learns the health status of each VPN overlay from each branch based on the advertised community string. The hub converts the community strings to route-tags, and uses the route-tags to determine the status of the link.

For example, if Branch1 advertises 65001:1 over HUB1-VPN1, the hub converts this to route-tag 1. When the service rules are matched, the service rule with route-tag 1 will have the destination network 192.168.136.0/24, and traffic to Branch1 will match the service rule and be steered to VPN1. If Branch2's HUB1-VPN1 is above the threshold, it advertises 65001:11 in its network prefix. The hub converts this to route-tag 11, and does not match the traffic destined for Branch2 to the service rule corresponding to route-tag 1. Branch2's HUB1-VPN2 is healthy, so it advertises 65001:2 in its network prefix. The hub converts this to route-tag 2, matches the traffic to the corresponding server rule, and steers traffic to VPN2.

To configure the hub:

```
config system sdwan
    config service
        edit 10
            set name "ToBranches1"
            set route-tag 1
            set src "all"
            set priority-members 1
        next
        edit 11
            set name "ToBranches2"
            set route-tag 2
            set src "all"
            set priority-members 2
        next
        edit 12
            set name "ToBranches3"
            set route-tag 11
            set src "all"
            set priority-members 1 2
        next
    end
end
```

Firewall policies

Standard firewall policies should be configured on the hub and the branches. On the hub, make sure that traffic is allowed from the overlays to the Datacenter network and the loopback interface.

See Policies in the FortiOS Administration Guide for information about configuring policies.

Routing configurations for traffic from the branches to the hub

The configurations for steering traffic from the branches to the hub are not included in this example. As each branch is configured with performance SLA health checks to the hub, standard SD-WAN rules can be used on the branches to steer traffic based on the status of the local health checks.

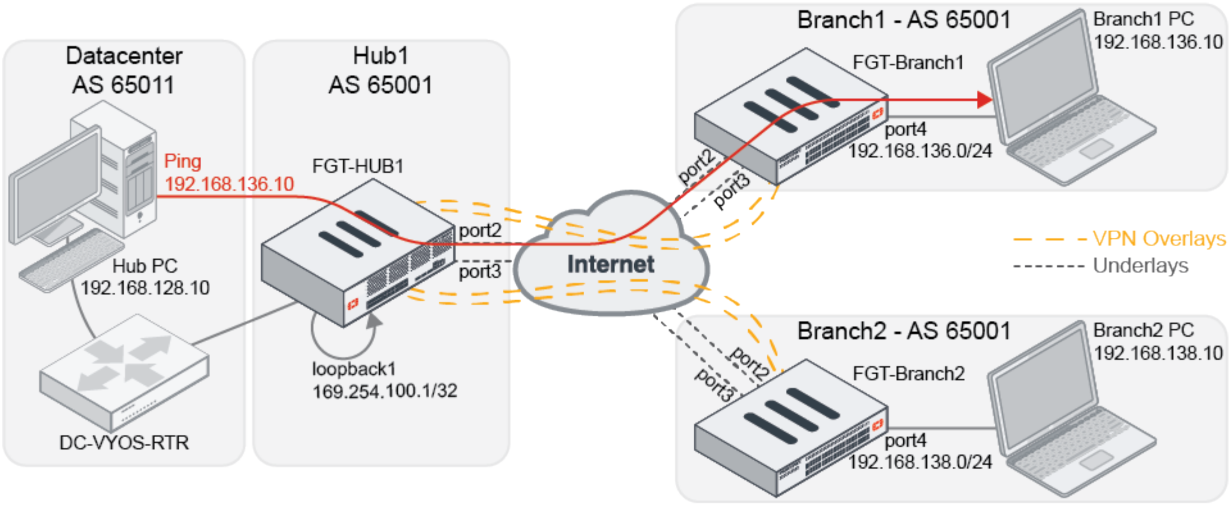
See SD-WAN rules in the FortiOS Administration Guide for information about configuring SD-WAN rules.

Testing and verification

1. Traffic flow during normal operation on page 21
2. Traffic flow when Branch1 ISP1 is impaired on page 25
3. Traffic flow when Branch1 ISP1 and ISP2 are impaired on page 28

Traffic flow during normal operation

These steps go through debug commands on the hub and branch FortiGates during normal operation. Simple traffic is generated to observe the steering that is occurring on the hub. The routing and session tables are shown to help understand the traffic flow. Artificial delays are added to show delays in the ISP links, but the delays do no exceed the health check's thresholds.



To test the traffic flow:

1. On the Hub PC, start a continuous ping to the Branch1 PC at 192.168.136.10:

```
hub1:~$ ping 192.168.136.10
PING 192.168.136.10 (192.168.136.10) 56(84) bytes of data.
64 bytes from 192.168.136.10: icmp_seq=1 ttl=61 time=74.0 ms
64 bytes from 192.168.136.10: icmp_seq=2 ttl=61 time=68.7 ms
...
```

2. On FGT-Branch1, check the status of the health check:

```
FGT-Branch1 # diagnose sys sdwan health-check HUB1_HC
Health Check(HUB1_HC):
Seq(1 HUB1-VPN1): state(alive), packet-loss(0.000%) latency(66.847), jitter(0.139) sla_
map=0x1
```

```
Seq(2 HUB1-VPN2): state(alive), packet-loss(0.000%) latency(73.992), jitter(0.352) sla_
map=0x1
```

Both HUB1-VPN1 and HUB1-VPN2 are alive and below the threshold.

3. On FGT-Branch2, check the status of the health check:

   ```
   FGT-Branch2 # diagnose sys sdwan health-check HUB1_HC
   Health Check(HUB1_HC):
   Seq(1 HUB1-VPN1): state(alive), packet-loss(0.000%) latency(77.116), jitter(0.206) sla_
   map=0x1
   Seq(2 HUB1-VPN2): state(alive), packet-loss(0.000%) latency(83.929), jitter(0.275) sla_
   map=0x1
   ```

   Both HUB1-VPN1 and HUB1-VPN2 are alive and below threshold.

4. On FGT-HUB1, check the BGP routes learned from FGT-Branch1:

   ```
   FGT-HUB1 # get router info bgp network 192.168.136.0/24
   VRF 0 BGP routing table entry for 192.168.136.0/24
   Paths: (2 available, best #2, table Default-IP-Routing-Table)
     Advertised to non peer-group peers:
      192.168.132.2
     Advertised to peer-groups:
     VPN1 VPN2
     Original VRF 0
     Local, (Received from a RR-client)
       169.254.16.11 from 169.254.16.11 (192.168.136.1)
         Origin IGP metric 0, route tag 1, localpref 100, valid, internal, best
         Community: 65001:1
         Advertised Path ID: 1
          Last update: Thu Jul 22 09:38:32 2021

     Original VRF 0
     Local, (Received from a RR-client)
       169.254.17.11 from 169.254.17.11 (192.168.136.1)
         Origin IGP metric 0, route tag 2, localpref 100, valid, internal, best
         Community: 65001:2
         Advertised Path ID: 2
          Last update: Thu Jul 22 09:38:31 2021
   ```

   The route was received by FGT-HUB1 from BGP neighbors 169.254.16.11 and 169.254.17.11, which correspond to the two tunnels on FGT-Branch1. Each is advertising their respective communities for successful health checks, and the hub assigns them route-tags 1 and 2.

   Repeat the command for network 192.168.138.0/24 on Branch2 to get similar outputs.

5. On FGT-HUB1, check all of the BGP routes installed in the routing table:

   ```
   FGT-HUB1 # get router info routing-table bgp
   Routing table for VRF=0
   B       192.168.128.0/24 [20/0] via 192.168.132.2, port4, 1d01h58m
   B       192.168.136.0/24 [200/0] via 169.254.16.11, VPN1, 1d01h47m
                            [200/0] via 169.254.17.11, VPN2, 1d01h47m
   B       192.168.138.0/24 [200/0] via 169.254.16.10, VPN1, 1d01h47m
                            [200/0] via 169.254.17.10, VPN2, 1d01h47m
   ```

6. On FGT-HUB1, check the status of the SD-WAN service rules:

   ```
   FGT-HUB1 # diagnose sys sdwan service
   ```

```
Service(10): Address Mode(IPV4) flags=0x200
  Gen(5), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Members(1):
    1: Seq_num(1 VPN1), alive, selected
  Src address(1):
        0.0.0.0-255.255.255.255

  Route tag address(2):
        192.168.136.0-192.168.136.255
        192.168.138.0-192.168.138.255


Service(11): Address Mode(IPV4) flags=0x200
  Gen(5), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Members(1):
    1: Seq_num(2 VPN2), alive, selected
  Src address(1):
        0.0.0.0-255.255.255.255

  Route tag address(2):
        192.168.136.0-192.168.136.255
        192.168.138.0-192.168.138.255


Service(12): Address Mode(IPV4) flags=0x200
  Gen(5), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Service disabled caused by no destination.
  Members(2):
    1: Seq_num(1 VPN1), alive, selected
    2: Seq_num(2 VPN2), alive, selected
  Src address(1):
        0.0.0.0-255.255.255.255
```

SD-WAN service rules 10, 11, and 12 are shown:

- Rule 10 matches traffic going to networks with BGP route-tag 1, currently 192.168.136.0/24 and 192.168.138.0/24. The SD-WAN rule uses manual mode to steer traffic to VPN1. Traffic that is being sent to both branches should match service rule 10 and be steered to VPN1.
- Rule 11 matches traffic going to networks with BGP route-tag 2, currently also 192.168.136.0/24 and 192.168.138.0/24, but because this rule is lower that rule 10, traffic does not hit it.

7. On FGT-HUB1, check the SD-WAN service rules in the policy route format:

```
FGT-HUB1 # diagnose firewall proute list
list route policy info(vf=root):
…
id=2130771978(0x7f01000a) vwl_service=10(ToBranches1) vwl_mbr_seq=1 dscp_tag=0xff 0xff
flags=0x40 order-addr tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0 dport=1-
65535 oif=10(VPN1)
source(1): 0.0.0.0-255.255.255.255
destination(2): 192.168.136.0-192.168.136.255 192.168.138.0-192.168.138.255
hit_count=1 last_used=2021-07-23 11:02:46

id=2130771979(0x7f01000b) vwl_service=11(ToBranches2) vwl_mbr_seq=2 dscp_tag=0xff 0xff
flags=0x40 order-addr tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0 dport=1-
65535 oif=11(VPN2)
source(1): 0.0.0.0-255.255.255.255
```

```
destination(2): 192.168.136.0-192.168.136.255 192.168.138.0-192.168.138.255
hit_count=0 last_used=2021-07-22 09:27:22
…
```

Traffic hits SD-WAN service rule 10, steering it to VPN1, while no traffic hit rule 11.

8. On both FGT-HUB1 and FGT-Branch1, check the session table to again confirm that traffic is hitting the right tunnel:
    - Hub:

```
FGT-HUB1 # diagnose sys session filter dst 192.168.136.10
FGT-HUB1 # diagnose sys session list

session info: proto=1 proto_state=00 duration=1 expire=59 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=VPN1_1/ vlan_cos=0/255
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 86/0 rx speed(Bps/kbps): 86/0
orgin->sink: org pre->post, reply pre->post dev=6->10/10->6
gwy=169.254.16.11/192.168.132.2
hook=pre dir=org act=noop 192.168.128.10:3422->192.168.136.10:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.136.10:3422->192.168.128.10:0(0.0.0.0:0)
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=00004d74 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=1 sdwan_service_id=10
rpdb_link_id=ff00000a rpdb_svc_id=0 ngfwid=n/a
npu_state=0x3040000
total session 1
```

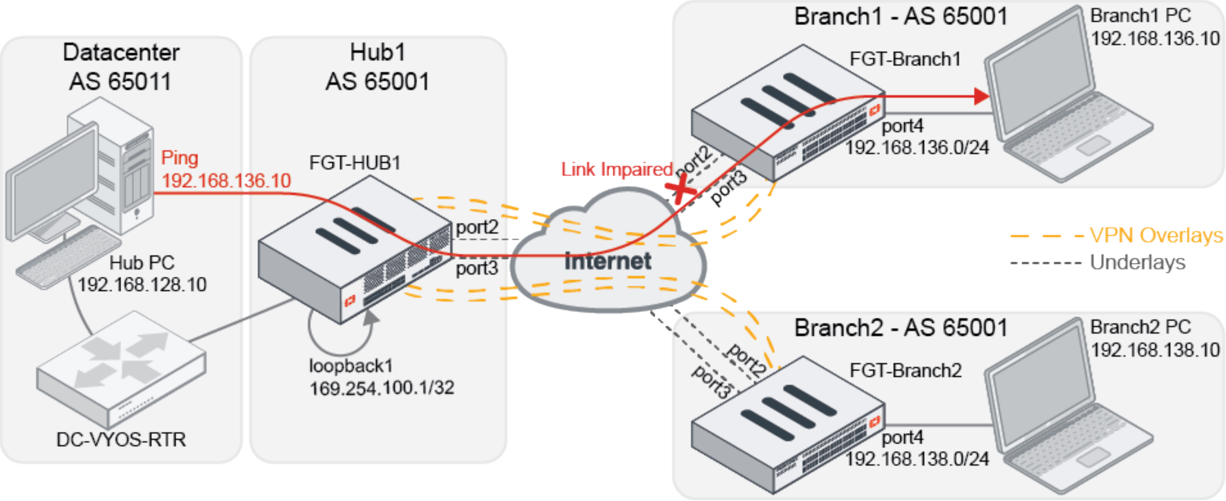
    - Branch:

```
FGT-Branch1 # diagnose sys session filter dst 192.168.136.10
FGT-Branch1 # diagnose sys session list

session info: proto=1 proto_state=00 duration=61 expire=59 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/HUB1-VPN1 vlan_cos=0/255
state=log may_dirty npu f00 app_valid
statistic(bytes/packets/allow_err): org=5124/61/1 reply=5124/61/1 tuples=2
tx speed(Bps/kbps): 84/0 rx speed(Bps/kbps): 84/0
orgin->sink: org pre->post, reply pre->post dev=9->6/6->9
gwy=192.168.136.10/169.254.16.1
hook=pre dir=org act=noop 192.168.128.10:3422->192.168.136.10:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.136.10:3422->192.168.128.10:0(0.0.0.0:0)
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=0000287e tos=ff/ff app_list=2000 app=24466 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x3041008
total session 1
```

Traffic flow when Branch1 ISP1 is impaired

Impairment to Branch1 ISP1 is simulated by adding additional latency that exceeds the Branch1 health check's thresholds. A continuous ping is used to show the effects that this has on traffic flow.



To test the traffic flow:

1. On the Hub PC, start a continuous ping to the Branch1 PC at 192.168.136.10.
2. Impair the ISP1 link on Branch1. In this example, latency is increased to 180ms.
3. On FGT-Branch1, check the status of the health check:

```
FGT-Branch1 # diagnose sys sdwan health-check HUB1_HC
Health Check(HUB1_HC):
Seq(1 HUB1-VPN1): state(alive), packet-loss(0.000%) latency(186.852), jitter(0.156) sla_
map=0x0
Seq(2 HUB1-VPN2): state(alive), packet-loss(0.000%) latency(73.891), jitter(0.168) sla_
map=0x1
```

   HUB1-VPN1 is above the latency threshold of 150ms. Its sla_map=0x0 indicates that the SLA is exceeded.

4. On FGT-HUB1, check the routes learned from FGT-Branch1:

```
FGT-HUB1 # get router info bgp network 192.168.136.0/24
VRF 0 BGP routing table entry for 192.168.136.0/24
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
   192.168.132.2
  Advertised to peer-groups:
  VPN1 VPN2
  Original VRF 0
  Local, (Received from a RR-client)
    169.254.16.11 from 169.254.16.11 (192.168.136.1)
      Origin IGP metric 0, route tag 11, localpref 100, valid, internal, best
      Community: 65001:11
      Advertised Path ID: 1
       Last update: Fri Jul 23 12:00:53 2021
  Original VRF 0
  Local, (Received from a RR-client)
    169.254.17.11 from 169.254.17.11 (192.168.136.1)
      Origin IGP metric 0, route tag 2, localpref 100, valid, internal, best
      Community: 65001:2
      Advertised Path ID: 2
       Last update: Thu Jul 22 09:38:31 2021
```

The route learned over VPN1 from Branch1 has community string 65001:11, showing that the health checks for this link has failed. FGT-HUB1 assigns route-tag 11 to the route. The same route learned over VPN2 from Branch1 is still assigned route-tag 2 based on community string 65001:2. The Branch1 FortiGate has successfully updated the hub on the status of its VPN overlays.

5. On FGT-HUB1, check all of the BGP routes installed in the routing table:

```
FGT-HUB1 # get router info routing-table bgp
Routing table for VRF=0
B       192.168.128.0/24 [20/0] via 192.168.132.2, port4, 1d02h52m
B       192.168.136.0/24 [200/0] via 169.254.16.11, VPN1, 00:18:57
                         [200/0] via 169.254.17.11, VPN2, 00:18:57
B       192.168.138.0/24 [200/0] via 169.254.16.10, VPN1, 1d02h41m
                         [200/0] via 169.254.17.10, VPN2, 1d02h41m
```

The routes to 192.168.136.0/24 over VPN1 is still installed, but its route-rag has changed.

6. On FGT-HUB1, check the status of the SD-WAN service rules:

```
FGT-HUB1 # diagnose sys sdwan service

Service(10): Address Mode(IPV4) flags=0x200
  Gen(6), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Members(1):
    1: Seq_num(1 VPN1), alive, selected
  Src address(1):
        0.0.0.0-255.255.255.255

  Route tag address(1):
        192.168.138.0-192.168.138.255


Service(11): Address Mode(IPV4) flags=0x200
  Gen(6), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Members(1):
    1: Seq_num(2 VPN2), alive, selected
  Src address(1):
        0.0.0.0-255.255.255.255

  Route tag address(2):
        192.168.136.0-192.168.136.255
        192.168.138.0-192.168.138.255


Service(12): Address Mode(IPV4) flags=0x200
  Gen(6), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Members(2):
    1: Seq_num(1 VPN1), alive, selected
    2: Seq_num(2 VPN2), alive, selected
  Src address(1):
        0.0.0.0-255.255.255.255

  Route tag address(1):
        192.168.136.0-192.168.136.255
```

Service rule 10, which matches route-tag 1, does not show the destination route-tag address of 192.168.136.0/24. Service rule 11, which matches route-tag 2, does not show the destination route-tag address of 192.168.136.0/24, as expected. The HUB PC's ping to 192.168.136.10 matches service rule 11 now, and not rule 10.

7. On the Hub PC, confirm that the pings are still flowing, and check the instant when the delay was introduced. Some pings might have continued to pass through VPN1 as the health check and routing updated occurred:

```
64 bytes from 192.168.136.10: icmp_seq=123 ttl=61 time=188 ms
64 bytes from 192.168.136.10: icmp_seq=124 ttl=61 time=188 ms
64 bytes from 192.168.136.10: icmp_seq=125 ttl=61 time=189 ms
64 bytes from 192.168.136.10: icmp_seq=126 ttl=61 time=75.8 ms
64 bytes from 192.168.136.10: icmp_seq=127 ttl=61 time=75.6 ms
64 bytes from 192.168.136.10: icmp_seq=128 ttl=61 time=75.7 ms
```

8. On FGT-HUB1, check the SD-WAN service rules in the policy route format:

```
FGT-HUB1 # diagnose firewall proute list
list route policy info(vf=root):
…
id=2130771978(0x7f01000a) vwl_service=10(ToBranches1) vwl_mbr_seq=1 dscp_tag=0xff 0xff
flags=0x40 order-addr tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0 dport=1-
65535 oif=10(VPN1)
source(1): 0.0.0.0-255.255.255.255
destination(1): 192.168.138.0-192.168.138.255
hit_count=3 last_used=2021-07-23 11:58:54

id=2130771979(0x7f01000b) vwl_service=11(ToBranches2) vwl_mbr_seq=2 dscp_tag=0xff 0xff
flags=0x40 order-addr tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0 dport=1-
65535 oif=11(VPN2)
source(1): 0.0.0.0-255.255.255.255
destination(2): 192.168.136.0-192.168.136.255 192.168.138.0-192.168.138.255
hit_count=2 last_used=2021-07-23 12:00:59
…
```

The proute that corresponds to service rule 10 does not list 192.168.136.0/24 as a destination. The proute that corresponds to service rule 11 includes 192.168.136.0/24 as a destination, and indicates that there are hits on that policy.

9. On both FGT-HUB1 and FGT-Branch1, check the session table to again confirm that traffic is hitting the right tunnel:
   - Hub:

```
FGT-HUB1 # diagnose sys session filter dst 192.168.136.10
FGT-HUB1 # diagnose sys session list

session info: proto=1 proto_state=00 duration=35 expire=59 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=VPN2_1/ vlan_cos=0/255
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=3024/36/1 reply=3024/36/1 tuples=2
tx speed(Bps/kbps): 84/0 rx speed(Bps/kbps): 84/0
```

```
orgin->sink: org pre->post, reply pre->post dev=6->11/11->6
gwy=169.254.17.11/192.168.132.2
hook=pre dir=org act=noop 192.168.128.10:3550->192.168.136.10:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.136.10:3550->192.168.128.10:0(0.0.0.0:0)
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=00004f53 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=2 sdwan_service_id=11
rpdb_link_id=ff00000b rpdb_svc_id=0 ngfwid=n/a
npu_state=0x3040000
total session 1
```

The session information shows that traffic is passing through tunnel VPN2_1, and is steered by SD-WAN service rule 11.

- Branch:

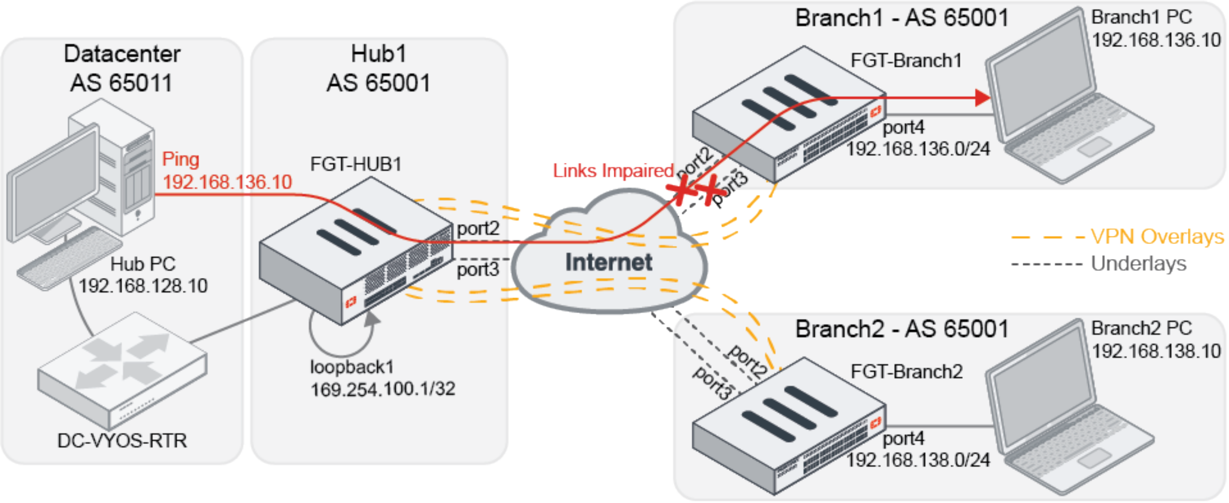
```
FGT-Branch1 # diagnose sys session filter dst 192.168.136.10
FGT-Branch1 # diagnose sys session list

session info: proto=1 proto_state=00 duration=89 expire=59 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/HUB1-VPN2 vlan_cos=0/255
state=log may_dirty npu f00 app_valid
statistic(bytes/packets/allow_err): org=7560/90/1 reply=7560/90/1 tuples=2
tx speed(Bps/kbps): 84/0 rx speed(Bps/kbps): 84/0
orgin->sink: org pre->post, reply pre->post dev=10->6/6->10
gwy=192.168.136.10/169.254.17.1
hook=pre dir=org act=noop 192.168.128.10:3550->192.168.136.10:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.136.10:3550->192.168.128.10:0(0.0.0.0:0)
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=00002910 tos=ff/ff app_list=2000 app=24466 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x3041008
total session 1
```

The session on Branch1 shows that traffic is passing through tunnel HUB1-VPN2.

Traffic flow when Branch1 ISP1 and ISP2 are impaired

Impairment to Branch1 ISP1 and ISP2 are simulated by adding additional latency that exceeds the Branch1 health check's thresholds. Because both ISPs are impaired, traffic matches SD-WAN service rule 12 on the hub, and flows over VPN1.



To test the traffic flow:

1. On the Hub PC, start a continuous ping to the Branch1 PC at 192.168.136.10.
2. Impair the ISP1 and ISP2 links on Branch1. In this example, ISP1 latency is increased to 180ms and ISP2 latency is increased to 195ms
3. On FGT-Branch1, check the status of the health check:

```
FGT-Branch1 # diag sys sdwan health-check HUB1_HC
Health Check(HUB1_HC):
Seq(1 HUB1-VPN1): state(alive), packet-loss(0.000%) latency(186.795), jitter(0.199) sla_
map=0x0
Seq(2 HUB1-VPN2): state(alive), packet-loss(0.000%) latency(203.939), jitter(0.235) sla_
map=0x0
```

   HUB1-VPN1 and HUB1-VPN2 are both above the latency threshold of 150ms. Its sla_map=0x0 indicates that the SLA is exceeded.

4. On FGT-HUB1, check the routes learned from FGT-Branch1:

```
FGT-HUB1 # get router info bgp network 192.168.136.0/24
VRF 0 BGP routing table entry for 192.168.136.0/24
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
   192.168.132.2
  Advertised to peer-groups:
  VPN1 VPN2
  Original VRF 0
  Local, (Received from a RR-client)
    169.254.16.11 from 169.254.16.11 (192.168.136.1)
      Origin IGP metric 0, route tag 11, localpref 100, valid, internal, best
      Community: 65001:11
      Advertised Path ID: 1
       Last update: Fri Jul 23 12:44:41 2021

  Original VRF 0
  Local, (Received from a RR-client)
    169.254.17.11 from 169.254.17.11 (192.168.136.1)
      Origin IGP metric 0, route tag 11, localpref 100, valid, internal, best
```

```
        Community: 65001:11
        Advertised Path ID: 2
         Last update: Fri Jul 23 12:59:28 2021
```

The route learned over both VPN1 and VPN2 from Branch1 has community string 65001:11, showing that the health checks for these links have failed. FGT-HUB1 assigns route-tag 11 to the routes. The Branch1 FortiGate has successfully updated the hub on the status of its VPN overlays.

5. On FGT-HUB1, check the status of the SD-WAN service rules:

```
FGT-HUB1 # diagnose sys sdwan service

Service(10): Address Mode(IPV4) flags=0x200
  Gen(9), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Members(1):
    1: Seq_num(1 VPN1), alive, selected
  Src address(1):
        0.0.0.0-255.255.255.255

  Route tag address(1):
        192.168.138.0-192.168.138.255


Service(11): Address Mode(IPV4) flags=0x200
  Gen(9), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Members(1):
    1: Seq_num(2 VPN2), alive, selected
  Src address(1):
        0.0.0.0-255.255.255.255

  Route tag address(1):
        192.168.138.0-192.168.138.255


Service(12): Address Mode(IPV4) flags=0x200
  Gen(9), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Members(2):
    1: Seq_num(1 VPN1), alive, selected
    2: Seq_num(2 VPN2), alive, selected
  Src address(1):
        0.0.0.0-255.255.255.255

  Route tag address(1):
        192.168.136.0-192.168.136.255
```

Service rules 10 and 11 do not show the destination route-tag address of 192.168.136.0/24, so neither rule is matched. Traffic falls to service rule 12, which catches failed route-tags. Traffic is steered in the order of VPN1 and VPN2.

6. On the Hub PC, confirm that the pings are still flowing, and check the instant when the delays were introduced. Some pings might have experienced small delays and continued to pass through VPN2, but eventually theroutes were updated on the hub, and traffic passed through VPN1 again :

```
64 bytes from 192.168.136.10: icmp_seq=910 ttl=61 time=205 ms
64 bytes from 192.168.136.10: icmp_seq=911 ttl=61 time=205 ms
64 bytes from 192.168.136.10: icmp_seq=912 ttl=61 time=205 ms
64 bytes from 192.168.136.10: icmp_seq=913 ttl=61 time=188 ms
```

```
64 bytes from 192.168.136.10: icmp_seq=914 ttl=61 time=188 ms
64 bytes from 192.168.136.10: icmp_seq=915 ttl=61 time=188 ms
```

7. On FGT-HUB1, check the SD-WAN service rules in the policy route format:

```
FGT-HUB1 # diagnose firewall proute list
list route policy info(vf=root):

id=2130771978(0x7f01000a) vwl_service=10(ToBranches1) vwl_mbr_seq=1 dscp_tag=0xff 0xff
flags=0x40 order-addr tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0 dport=1-
65535 oif=10(VPN1)
source(1): 0.0.0.0-255.255.255.255
destination(1): 192.168.138.0-192.168.138.255
hit_count=5 last_used=2021-07-23 12:44:19

id=2130771979(0x7f01000b) vwl_service=11(ToBranches2) vwl_mbr_seq=2 dscp_tag=0xff 0xff
flags=0x40 order-addr tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0 dport=1-
65535 oif=11(VPN2)
source(1): 0.0.0.0-255.255.255.255
destination(1): 192.168.138.0-192.168.138.255
hit_count=5 last_used=2021-07-23 12:44:47

id=2130771980(0x7f01000c) vwl_service=12(ToBranches3) vwl_mbr_seq=1 2 dscp_tag=0xff 0xff
flags=0x40 order-addr tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0 dport=1-
65535 oif=10(VPN1) oif=11(VPN2)
source(1): 0.0.0.0-255.255.255.255
destination(1): 192.168.136.0-192.168.136.255
hit_count=2 last_used=2021-07-23 12:59:33
```

8. On both FGT-HUB1 and FGT-Branch1, check the session table to again confirm that traffic is hitting the right tunnel:
   - Hub:

```
FGT-HUB1 # diagnose sys session filter dst 192.168.136.10
FGT-HUB1 # diagnose sys session list

session info: proto=1 proto_state=00 duration=2440 expire=59 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=VPN1_1/ vlan_cos=0/255
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=204792/2438/1 reply=204792/2438/1 tuples=2
tx speed(Bps/kbps): 83/0 rx speed(Bps/kbps): 83/0
orgin->sink: org pre->post, reply pre->post dev=6->10/10->6
gwy=169.254.16.11/192.168.132.2
hook=pre dir=org act=noop 192.168.128.10:3550->192.168.136.10:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.136.10:3550->192.168.128.10:0(0.0.0.0:0)
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=00004f53 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=1 sdwan_service_id=12
rpdb_link_id=ff00000c rpdb_svc_id=0 ngfwid=n/a
npu_state=0x3040000
total session 1
```

   - Branch:

```
FGT-Branch1 # diagnose sys session filter dst 192.168.136.10
FGT-Branch1 # diagnose sys session list

session info: proto=1 proto_state=00 duration=2462 expire=59 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/HUB1-VPN1 vlan_cos=0/255
state=log may_dirty npu f00 app_valid
statistic(bytes/packets/allow_err): org=206640/2460/1 reply=206640/2460/1 tuples=2
tx speed(Bps/kbps): 83/0 rx speed(Bps/kbps): 83/0
orgin->sink: org pre->post, reply pre->post dev=9->6/6->9
gwy=192.168.136.10/169.254.16.1
hook=pre dir=org act=noop 192.168.128.10:3550->192.168.136.10:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.136.10:3550->192.168.128.10:0(0.0.0.0:0)
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=00002910 tos=ff/ff app_list=2000 app=24466 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x3041008
total session 1
```

www.fortinet.com