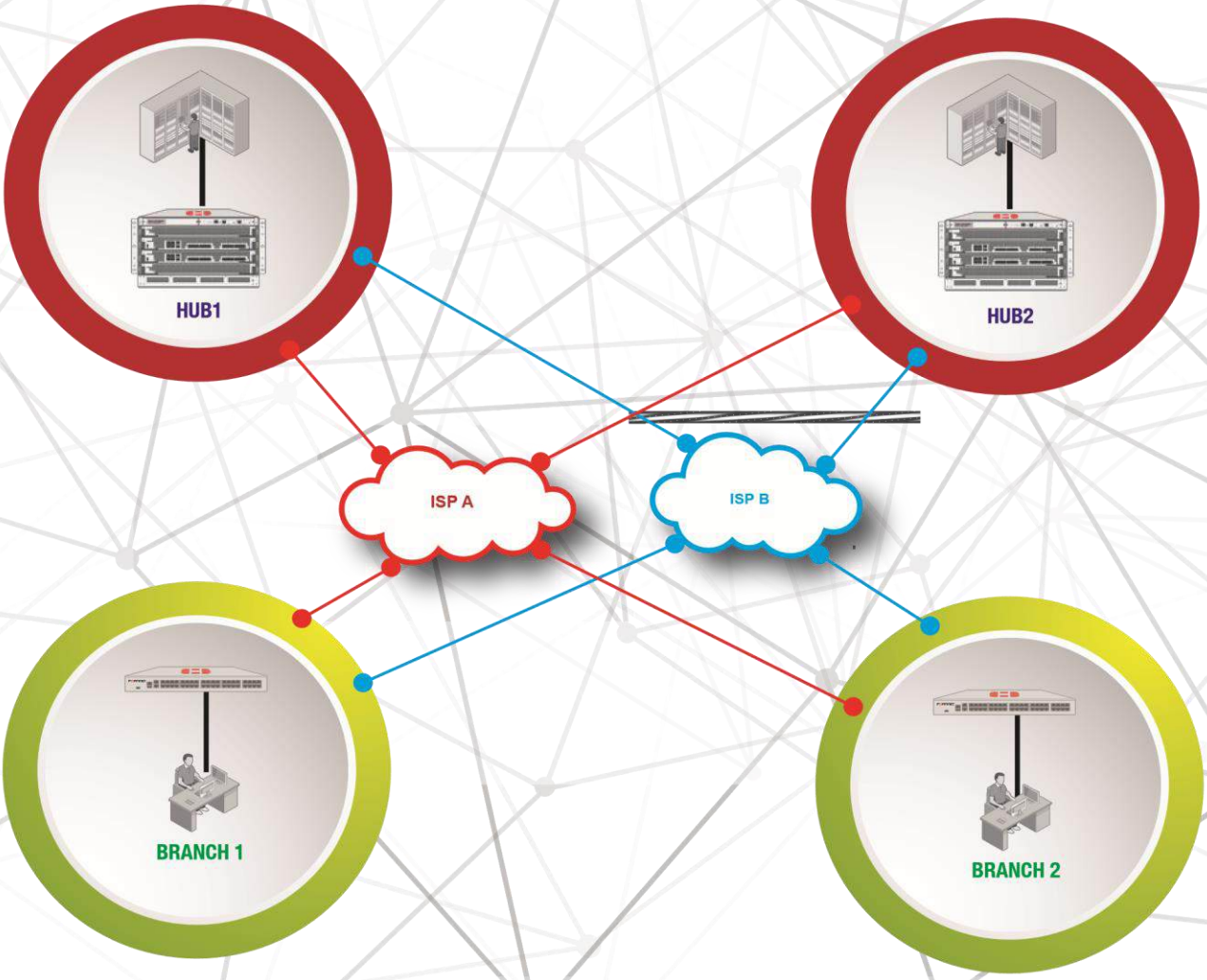
FORTINET SD-WAN DEMYSTIFIED

Ultimate Technical Deployment Guide



Nicanory Ateya CCIE Sec 61792

Warning and Disclaimer

This book focuses on deploying Fortinet SD-WAN solution and great effort has been made to make the information as accurate as possible, but no warranty or fitness is implied. The information is provided on "as is" basis.

The opinions expressed in this book belong to the author and are not necessarily those of Fortinet.

About The Author

Nicanory Ateya, CCIE 61792, is a solution architect with over 15 years of diverse experience in design, deployment and support of multivendor network, data center and security technologies. He holds a Bachelor's degree in Computer Engineering. Nick has worked at multiple organizations including Fortinet, as a systems engineer with a Cisco Gold partner and a product manager at a leading value-add distributor . Training and sharing his knowledge in the most simplified way has been a core part of his journey and the motivation for this book. After involvement in multiple Fortinet SDWAN projects from 10s to 100s of sites, the book represents the practical approach required to successfully deliver and support similar projects.

Dedication

To my two sets of parents for their guidance and prayers. To my wife, Catherine, for her love and support and to my daughters Maleen and Nicole for making each day special. Finally, to my brother and mentor Gerald Simila for imprinting excellence as a bare minimum in all endeavors.

Who Is This Book For?

You are likely to benefit from reading this book if you fall in any of the categories below:

- **A Field engineer new to Fortinet SDWAN:** For field deployment engineers involved in the deployment of new and existing Fortinet SDWAN setups, the book will guide on the building blocks to effectively install and resolve various operational issues.
- **A Presales engineer:** Invaluable information is shared to guide the presales team in coming up with sound technical proposal of Fortinet components and the required links to execute projects successfully.
- **A client interested in Fortinet SDWAN:** The book shows the value Fortinet SDWAN has to offer, makes comparison with other vendors easier if still at the evaluation stage and becomes a key reference for greenfield deployments and troubleshooting of existing setups.

Recommendation

The book has a clear focus on Fortinet SDWAN with a brief description of the key components. If you are not experienced in FortiGate firewalls, FortiManager, FortiAnalyzer and Fortinet SDWAN, it is recommended to go through the available free training on Fortinet's training portal.

Basic understanding of BGP as a routing protocol is also recommended since it is the preferred protocol. At a minimum high-level knowledge on iBGP, route reflectors, route-maps and communities. This information summarized in the document "SD-WAN self-healing with BGP - FortiOS 7.0" as a quick reference.

The book has been simplified with images and configuration output to coalesce the information.

Contents

Chapter 1: Introduction to Fortinet SD-WAN

Software-defined Wide Area Networking provides secure, reliable connectivity between HUBs/Main sites and remote locations.

Fortinet's design includes the following key components:

i. **FortiManager**: Provides centralized management of devices. It performs the following:
   a. Configuration revision control and tracking for FortiGate, FortiAP, FortiSwitch and FortiExtender from a central dashboard.
   b. Firmware management.
   c. Scripting.

ii. **FortiAnalyzer**: Centralized logging reporting and security analytics

iii. **FortiGate**: Firewall that performs routing, security policies and SD-WAN.

Both FortiManager and FortiAnalyzer are not mandatory for an SD-WAN solution to work. As demonstrated later in the book, it is possible to deploy an SD-WAN solution using FortiGate firewalls only. The main hurdle with this approach is the great effort required to design, configure and manage the hundreds of commands between the HUB and branches. Although not recommended, it is possible to set up small deployments with 2-5 branches without FortiManager.

For deployments with 10+ sites or use cases with frequent topology changes, it is mandatory to have the FortiGate units managed by FortiManager to derive the full benefits of the solution.

In summary, FortiManager and FortiAnalyzer are not required for SDWAN to work. For deployments with FortiManager for Centralized provisioning and FortiAnalyzer for centralized logging, normal traffic flow between the sites is maintained when the two are switched off or unavailable.

SD-WAN is link agnostic utilizing any available media to steer traffic. It is possible to use MPLS, Dark Fiber, Starlink, 3/4/5G links to establish an SDWAN underlay network. A common misconception is the assumption that MPLS links are not compatible with SDWAN solutions. This is driven by campaigns that promote SDWAN as a replacement for MPLS.

SDWAN Building Blocks.

Interface Members

Interface members include all the virtual and physical interfaces that participate in SD-WAN. They are grouped into 'SD-WAN Zones' to simplify firewall policy management.

Underlay Network.

This is the path connecting the different sites. It could be any Internet link or private circuits from the ISP

Overlay Network.

These are secure tunnels established over the underlay network to securely connect them. They are simply dialup IPsec tunnels.

Performance SLA

Performance SLAs define the health of the links that are connected to SD-WAN member interfaces by sending probes to a target and measuring the link quality based on latency, jitter, and packet loss. SD-WAN rules reference the individual SLAs to allow per application steering.

SD-WAN Rules

SD-WAN rules define how to steer network traffic. By using information collected from define Performance SLAs SDWAN makes decisions in real time to select the best paths.

Traffic is evaluated top to bottom and the configuration logic is similar to firewall policies; more specific rules should be listed at the top, while more generic rules can be listed lower. Without a matching SD-WAN rule, an implicit "ALLOW ALL" at the bottom sends all traffic to the route table for normal processing.

Like policy-based routing, SD-WAN rules are only involved in link selection, firewall policies are still required to block/allow traffic. If any policy-based rule is configured, it will take precedence over any configured SD-WAN rules.

SD-WAN traffic is identified and matched based on a number of different criteria:

- Source
- Destination
- Service
- User group
- Application Identification
- ISDB
- ToS or DSCP
- Route tags

SD-WAN Steering Strategies define how the different paths are selected.

Manual: Static assignment of one or more interfaces to be used for traffic that matches source and destination. With this option, no health checks or link quality assessment is performed on the interface(s) selected. As long as the selected interface is up, it is used for all traffic that matches this rule. **This option is used with route tags at the HUB to steer traffic to the branches.**

Best Quality: Selects the best performing interface based on the quality criteria selected. The quality criteria could be latency, jitter, packet loss, available in/out bandwidth or a weighted combination of all four options. With this strategy, only the best performing link is selected.

This steering strategy requires defining a Performance SLA to gather link quality information to determine latency, jitter and/or packet loss.

Lowest Cost (SLA): Lowest Cost (SLA) steering option does not choose the best performing link. Instead, it considers:

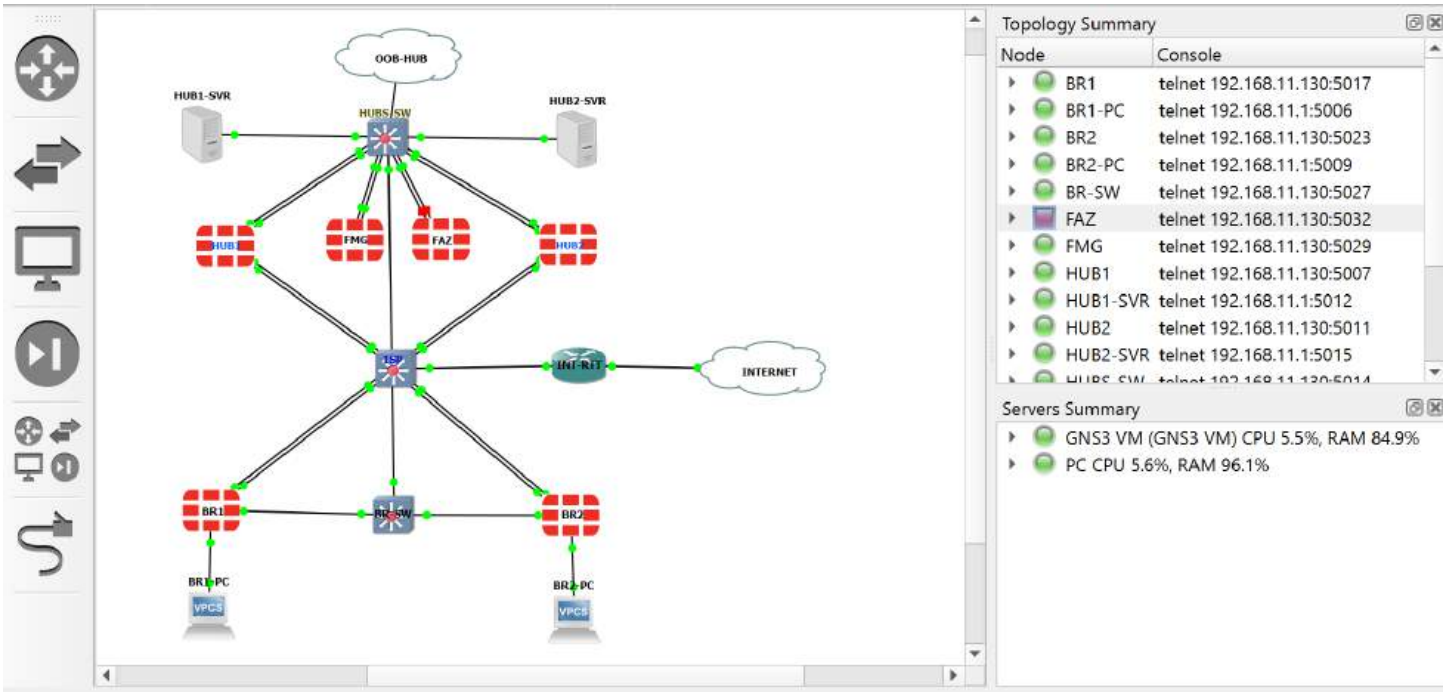
i. Whether the selected interface is meeting the defined SLA
ii. The lowest cost between the selected links.

This implies that a better link may not be selected for the traffic at a given time if it has a cost higher than the other link or out of SLA. In the event of a tie breaker (multiple interfaces with the same cost and SLA status), the SDWAN rule order preference will be taken into consideration.

Maximum Bandwidth (SLA): Per-session load balancing to all interfaces selected in the rule, as long as they are meeting their defined SLA. Interfaces not meeting their defined SLA are removed as eligible interfaces and traffic distributed to the remaining members.

LAB Environment Overview.

The devices used in the demo are licensed with all features. The labs are done in a GNS3 environment for flexibility and better revision control using snapshots.



The above topology supports single-hub and dual-hub scenarios with internet access for the underlay network. Great effort is made to simulate real deployment scenarios. This includes:

- All sites have redundant links.
- Branches will register to FortiManager over the internet.
- Support for HUB1-to-HUB2 traffic.

The connection details are listed below for reference.

HUB-SW1		VLAN			BR-SW1		VLAN
e0/0	OOB-CLOUD	5			e0/0	HUB-SW-e1/2	5
e0/1	HUB1-10	5	192.168.11.201		e0/1	BR1-10	5
e0/2	HUB2-10	5	192.168.11.202		e0/2	BR2-10	5
e0/3	FMG-4	5	192.168.11.15		e0/3		
e1/0	FAZ-4	5	192.168.11.16		e1/0		
e1/1	FAC-4	5	192.168.11.17		e1/1		
e1/2					e1/2		
e1/3	HUB1-SVR	2	10.10.100.10		e1/3		
e2/0	HUB1-3	2	10.10.100.1		e2/0		
e2/1	FMG-1	2	10.10.100.2		e2/1		
e2/2	FAZ-1	2	10.10.100.3		e2/2		
e2/3	FAC-1	2	10.10.100.4		e2/3		
e3/0	HUB2-3	3	10.10.200.1		e3/0		
e3/1	HUB1-SVR	3	10.10.200.10		e3/1		
e3/2					e3/2		
e3/3					e3/3		

ISP SW		IP	VLAN
e0/0	HUB1-1	100.1.100.0/30	10
e0/1	HUB1-2	200.2.100.0/30	20
e0/2	HUB2-1	100.1.200.0/30	30
e0/3	HUB2-2	200.2.200.0/30	40
e1/0	BR1-1	100.1.1.0/30	50
e1/1	BR1-2	200.2.1.0/30	60
e1/2	BR2-1	100.1.2.0/30	70
e1/3	BR2-2	200.2.2.0/30	80
e2/0			
e2/1			
e2/2			
e2/3			
e3/0	INT-RTR-0/1	20.10.10.100/24	100
e3/1			
e3/2			
e3/3			

Overlay Tunnel mapping

TUNNNEL	SUBNET	HUB1 ISP	BRANCHx ISP
TUN1	10.50.1.0/24	ISPA (port1)	ISPA (port1)
TUN2	10.50.2.0/24	ISPB (port2)	ISPB (port2)
TUN3	10.50.3.0/24	ISPA (port1)	ISPB (port2)
TUN4	10.50.4.0/24	ISPB (port2)	ISPA (port1)

Branch IDs and Underlay network

DEVICE	ID (X)	LOCATION	UNDERLAY	
HUB1	100	NAIROBI	ISP1	100.1.X.0/30
HUB2	200	THIKA	ISP2	200.2.X.0/31
BR1	1	NAKURU	OVERLAY	10.50.TUN#.0/24
BR2	2	MOMBASA		
DATA	10.10.X.0/24			

Tunnel naming convention

HUBs

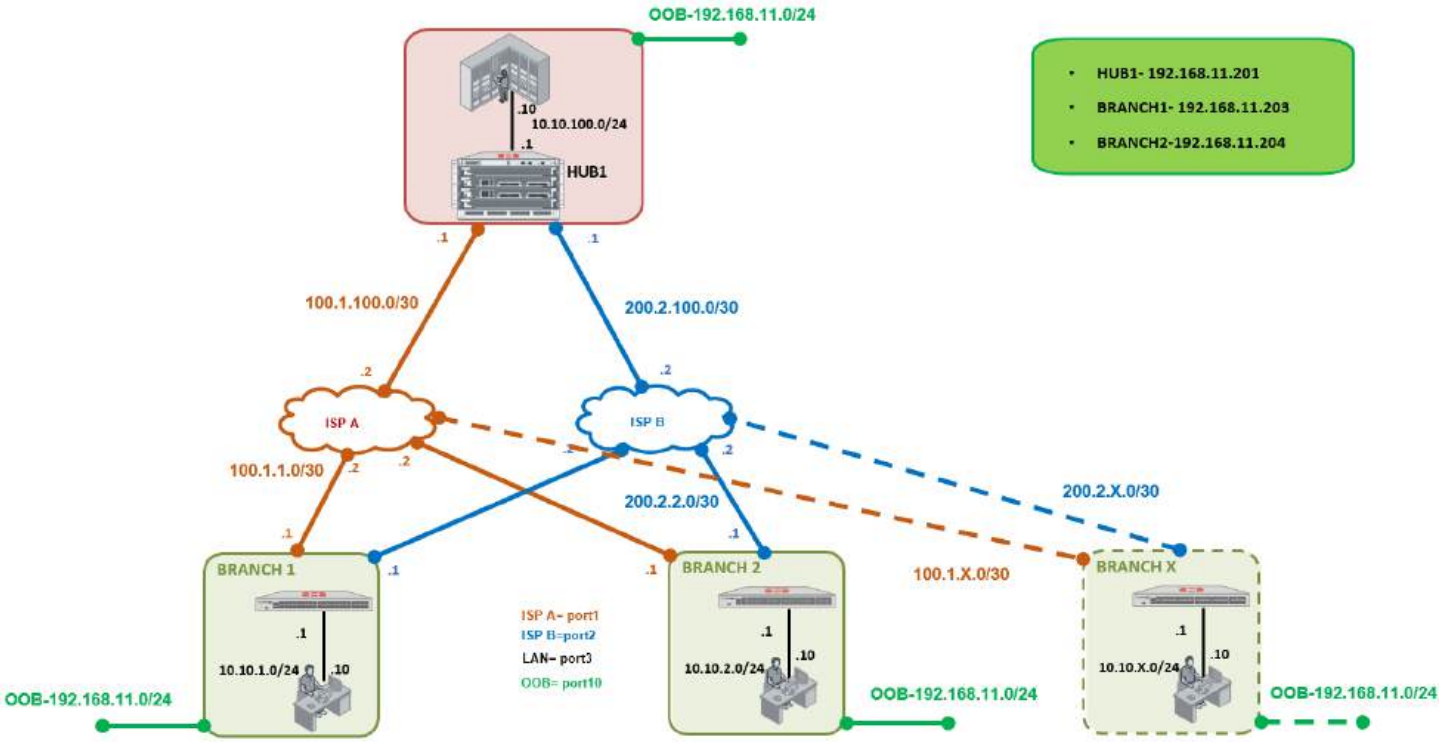
HUB VPN tunnels are named VPNx where X is the tunnel number e.g. VPN1, VPN2 etc.

Branches

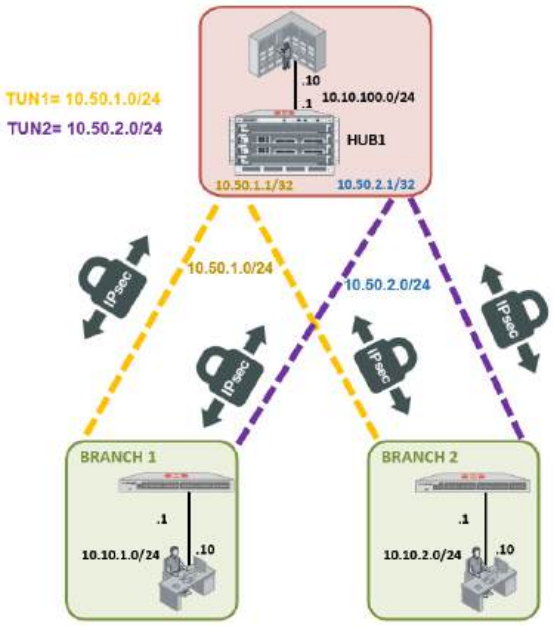
Branch tunnels are named HUBy-VPNx where y is the HUB site identifier and x the tunnel number e.g. The first tunnel to HUB1 would be HUB1-VPN1 while the first to HUB2 will be HUB2-VPN5. This will be elaborated in the LABs

Chapter 2: Deployment using GUI & CLI

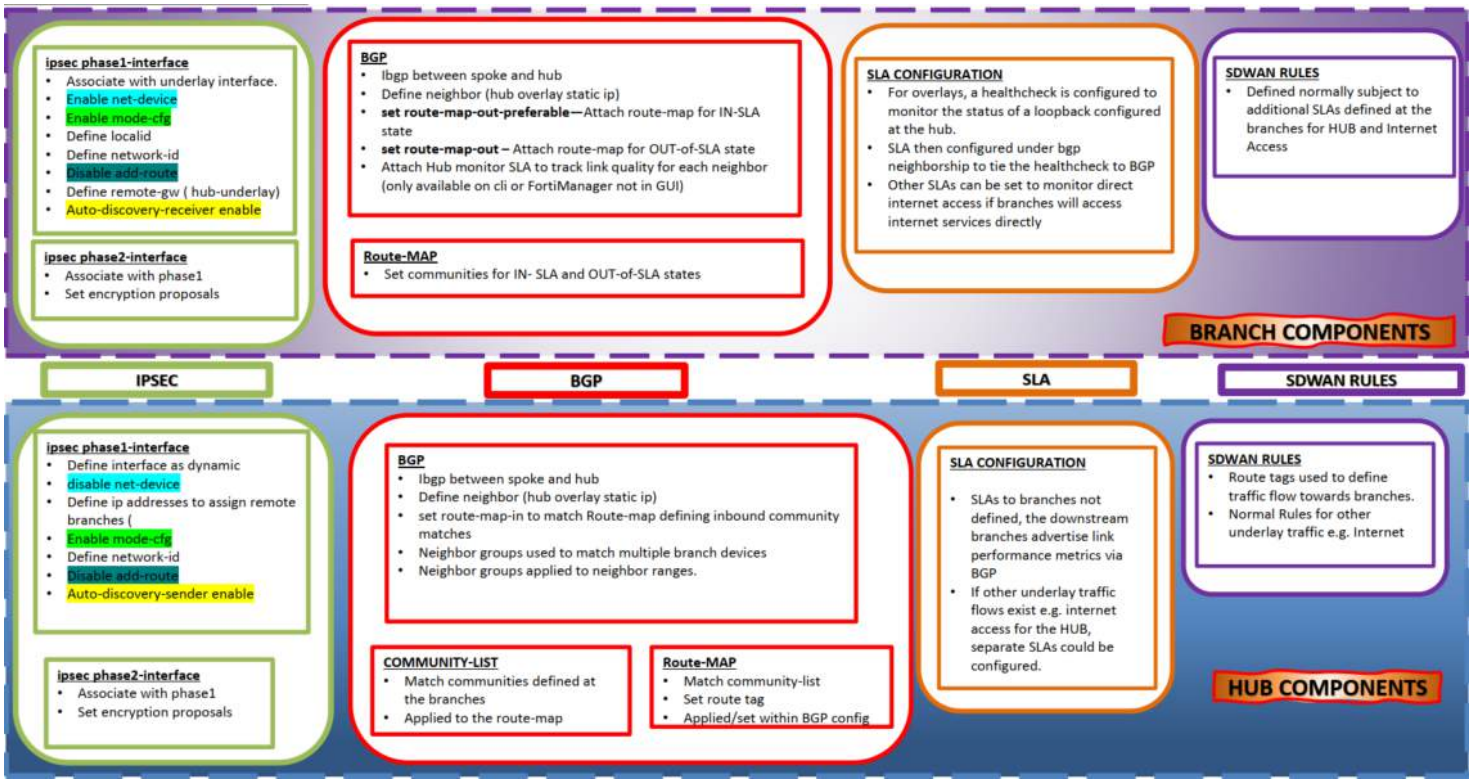
The diagram below represents the reference topology for the GUI & CLI setup made up of two branches and a single hub. Each site has two undelay/ISP links. The underlay path has a single overlay network from the branches to the HUB.



The topology will be extended to include different deployment scenarios. The initial setup has two overlay networks for each branch for secure IPSec tunnels that are used as the overlay networks. iBGP runs on the overlay network to advertise the prefixes at each site and share link performance metrics from each branch to the HUBs.

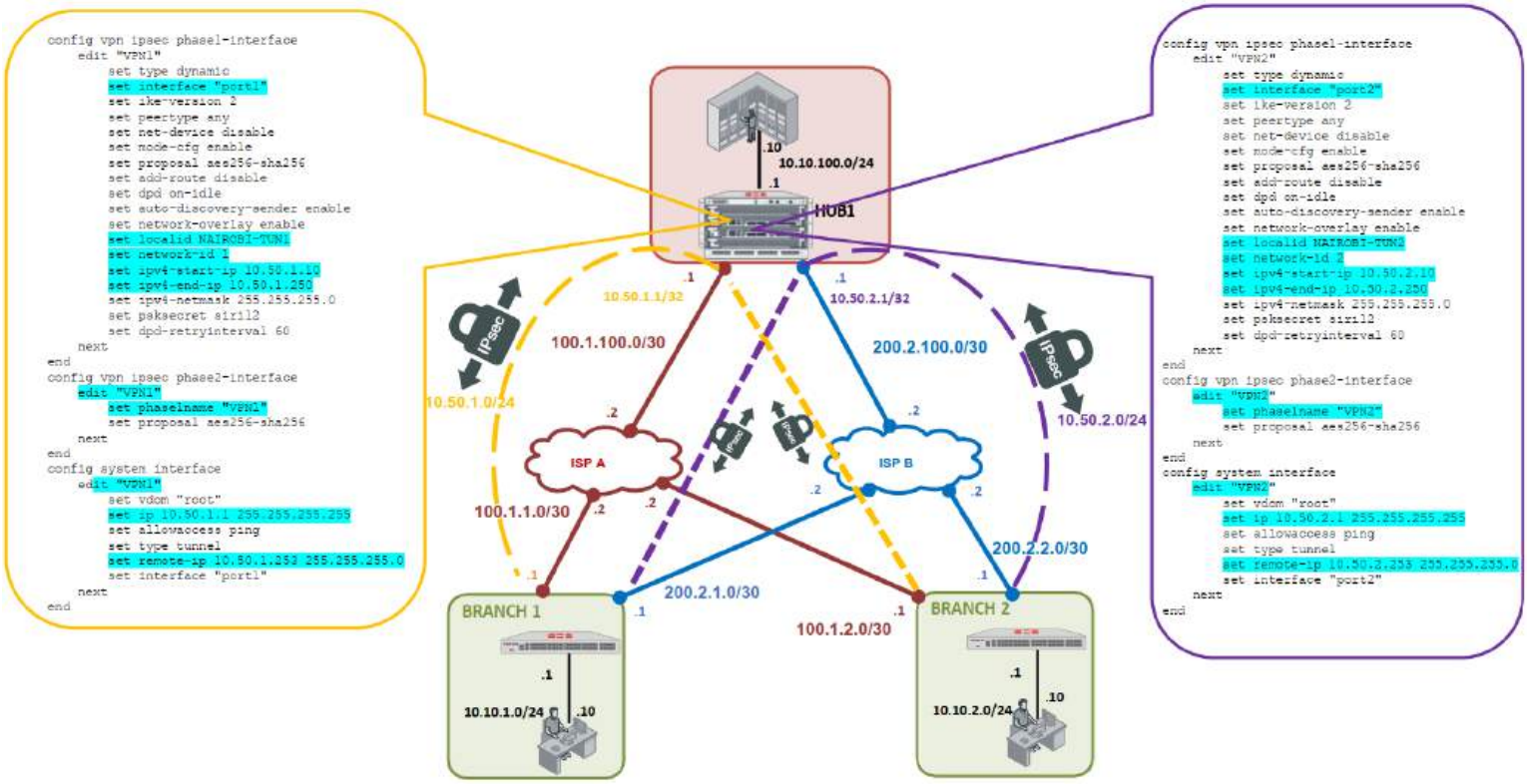


Fortinet's SD-WAN has four main components that are configured at the HUB and respective branches. These include **IPsec, BGP, SLA and SD-WAN rules**. Each has sub-components as highlighted in the diagram below. This will be referenced to configure HUB1 to connect to the two branches via CLI and GUI.



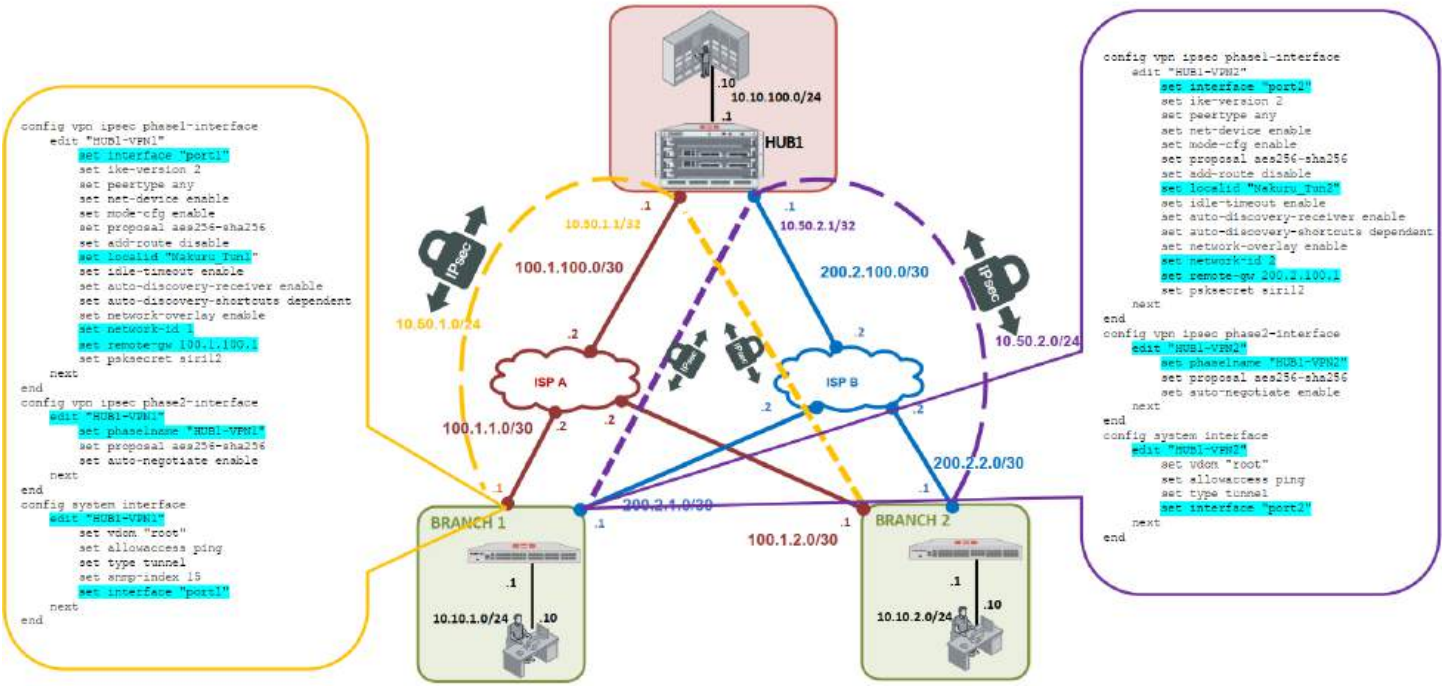
IPsec Configuration

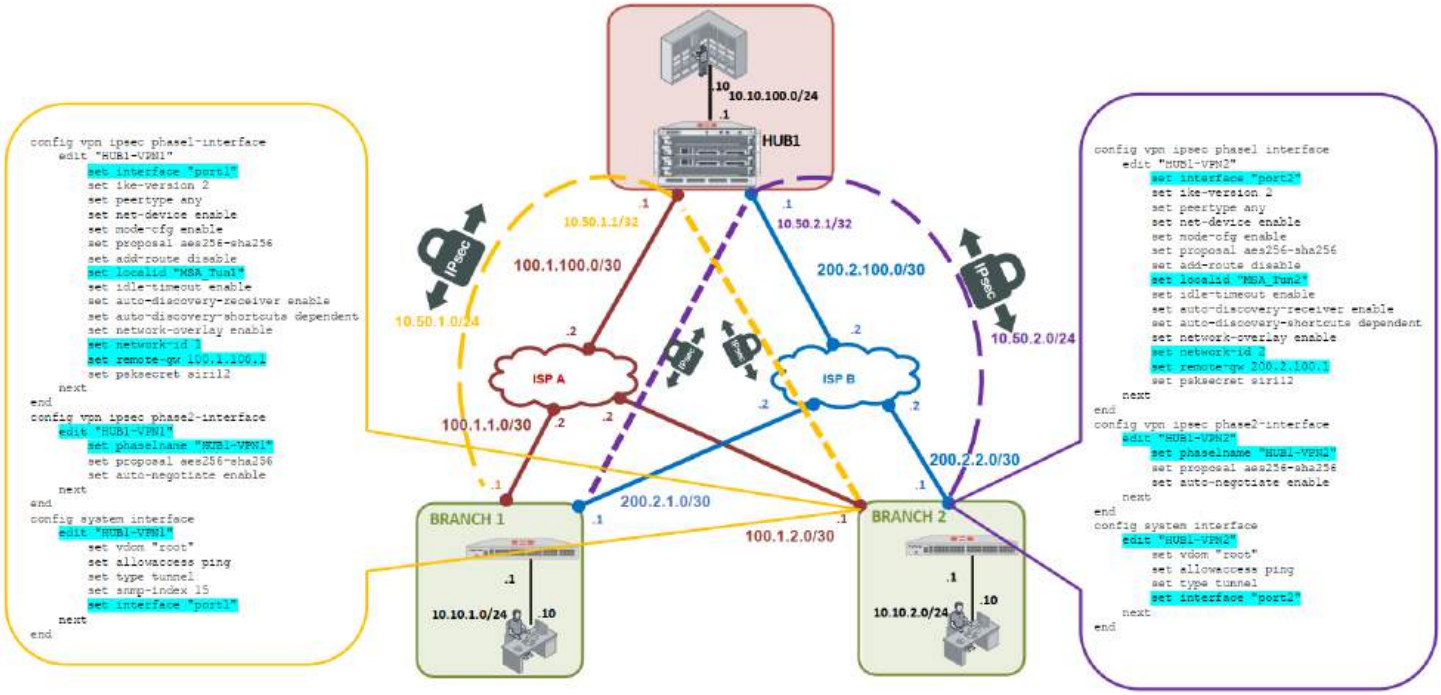
The configuration starts with IPsec to bring up the overlay tunnels. The diagram below represents the IPsec configurations required for VPN1 and VPN2 interfaces at HUB1.



VPN1 terminates on port1 of HUB1 while VPN2 is on port2. The highlighted sections represent the configuration differences between VPN1 and VPN2. VPN1 assigns IP addresses from 10.50.1.10 to 10.50.1.250 and VPN2 assigns IP addresses from 10.50.2.10 to 10.50.2.250 . VPN1 has ip address 10.50.1.1/32 with the remote ip 10.50.1.253/24. The interface ip must be a /32 and the remote ip dictates the network range the interface is assigned in. The same logic applies to VPN2 with ip address 10.50.2.1/32 and 10.50.2.253/24 for the remote ip. BGP neighbors are formed over the VPN overlays.

Branch 1 HUB1-VPN1 and HUB1-VPN2 configurations are shown below. This will configure two dialup tunnels to connect to HUB1's VPN1 and VPN2.





Branch 2 HUB1-VPN1 and HUB1-VPN2 configurations are shown above. This configures two dialup tunnels to connect to HUB1's VPN1 and VPN2. The two tunnels dial to the HUBs VPN1 and VPN2 for ip address assignment. The network ID must match e.g. id 1 for the HUB's VPN1 and HUB1-VPN1 at the branch. Local IDs are used to define the tunnels for easier visibility and troubleshooting. ICMP traffic must be allowed on the branch HUB-VPNx interfaces to support ADVPN. This is discussed in detail later.

Mismatch in PSK

The tunnel won't come up if we have a mismatch in the PSK, from the debug shown it indicates **probable pre-shared key mismatch** at HUB1

```
HUB1 # diagnose debug application ike -255
Debug messages will be on for 30 minutes.

HUB1 # ike 0: invalid IKE request SPI 6eb047813bd34762/522d644ea0e17e1f:00000001

HUB1 #
HUB1 #
HUB1 # diagnose debug enable

HUB1 #
HUB1 # ike 0: invalid IKE request SPI 6eb047813bd34762/522d644ea0e17e1f:00000001
ike 0:VPN1:54: auth verify done
ike 0:VPN1:54: PSK auth failed: probable pre-shared key mismatch
ike Negotiate SA Error: ike ike  [7263]
```

Mismatch in Network ID

The network-IDs of the HUB and the branch must match for each tunnel i.e. VPNx and HUB1-VPNx, and need to be unique. The ID must be unique at the HUB but branches have the same IDs for the respective tunnels.

```
HUB1 # diagnose debug application ike -1
Debug messages will be on for 30 minutes.

HUB1 # diagnose debug enable
```

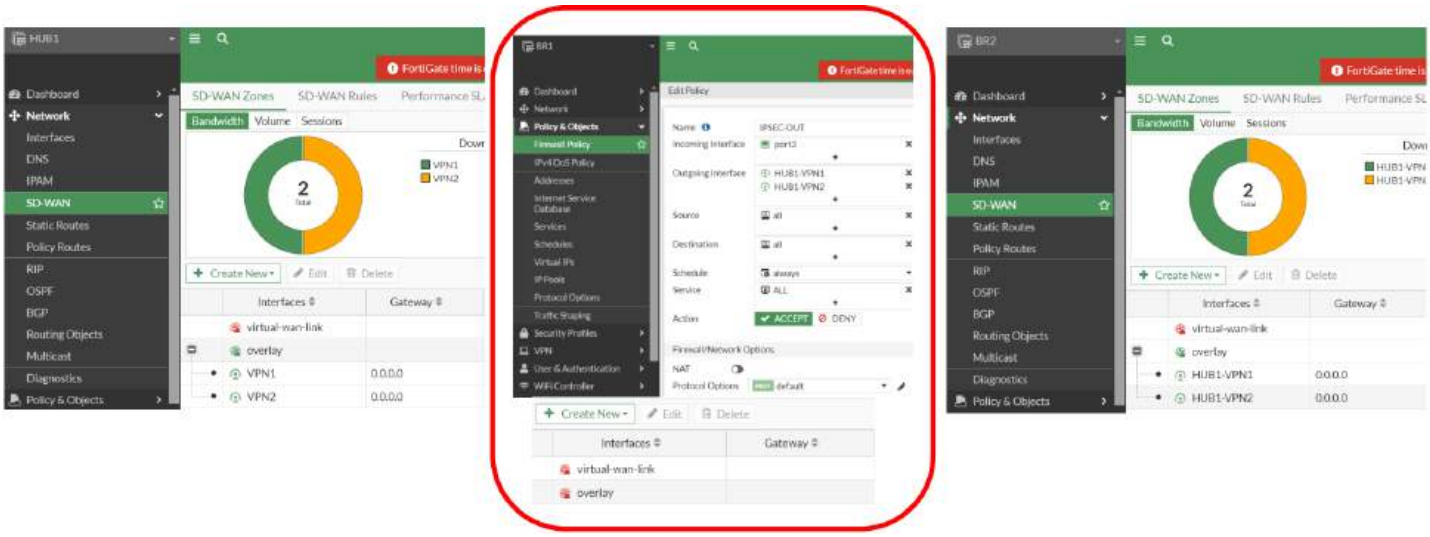
Enable the debugs above to check the VPN state. The output below is given when there is a network –ID mismatch between the HUB and branch i.e. **no proposal chosen** with the debugs done at the HUB.

```
HUB1 (phase1-interface) # ike 0:21719df552f6f274/0000000000000000:27: no proposal chosen
ike Negotiate SA Error: ike ike  [11064]
ike 0:21719df552f6f274/0000000000000000:28: no proposal chosen
ike Negotiate SA Error: ike ike  [11064]
ike 0:21719df552f6f274/0000000000000000:29: no proposal chosen
ike Negotiate SA Error: ike ike  [11064]
ike 0:21719df552f6f274/0000000000000000:30: no proposal chosen
ike Negotiate SA Error: ike ike  [11064]
```

Requirement for firewall policies at the HUB and Branches

After IPsec tunnels are configured, they will remain disabled/down unless the following requirements are met:

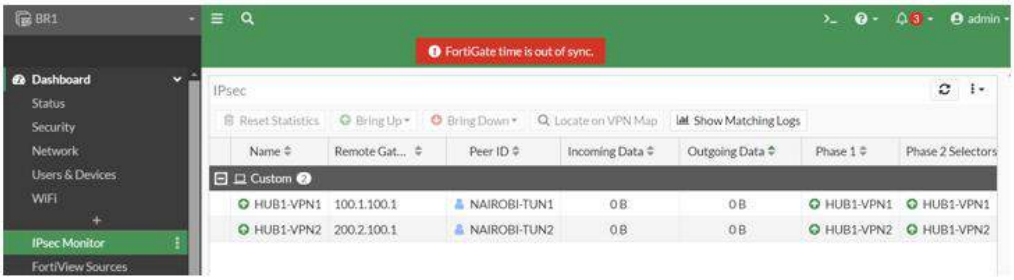
- Configure an outbound policy at the branch referencing the VPN tunnels as the destination and an inbound policy at the HUB with the VPN interfaces as the source interface.
- Bundle the VPN interfaces at the branches and HUB in respective SDWAN zones.
- A combination of the two, on side with the policy, the other with the SDWAN zone grouping.

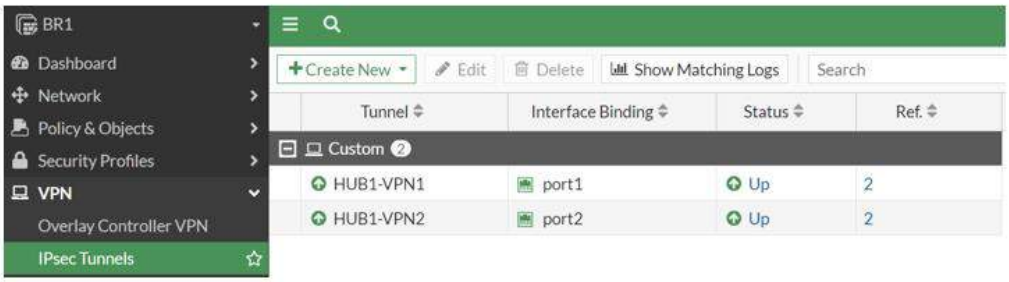


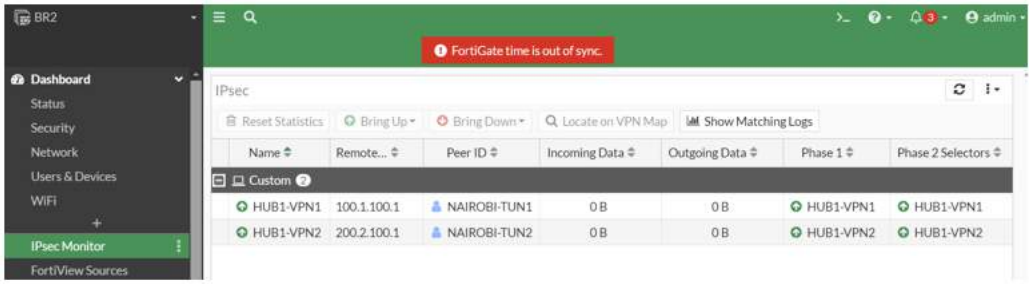
The setup above will not require policies at HUB1 and BR2 since they have SDWAN zones configured while BR1 will require outbound firewall policies to bring up the tunnels because it's HUB1-VPN1 and HUB1-VPN2 interfaces are not included in any SDWAN zone.

Branches IPSec Monitor

After configuring required firewall policies or assignment to SD-WAN zones at the HUB and branches, the tunnels will be active. This is seen under IPsec Monitor Dashboard or the configuration section for the VPNs at the branches. The peer ID shown at the branches is the local ID configured at the HUB.

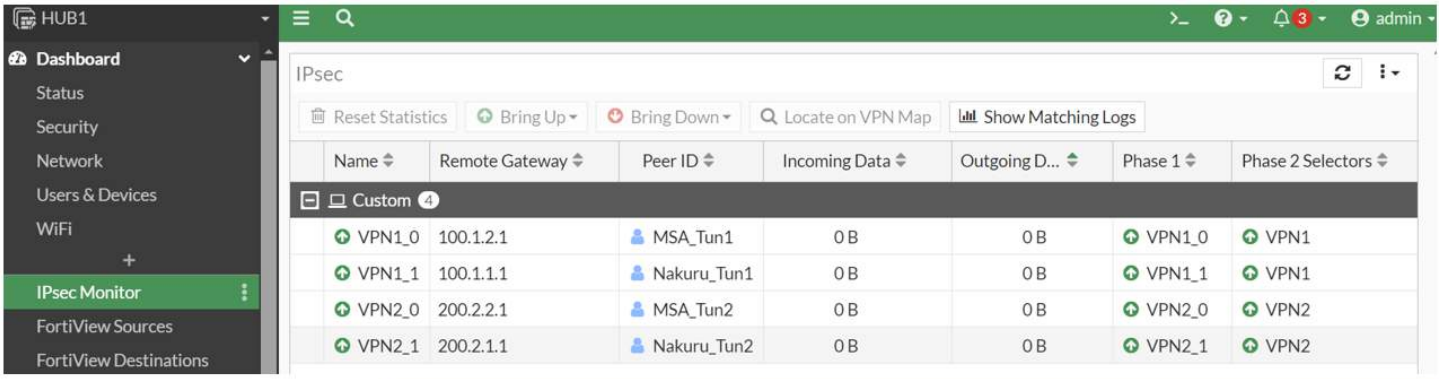


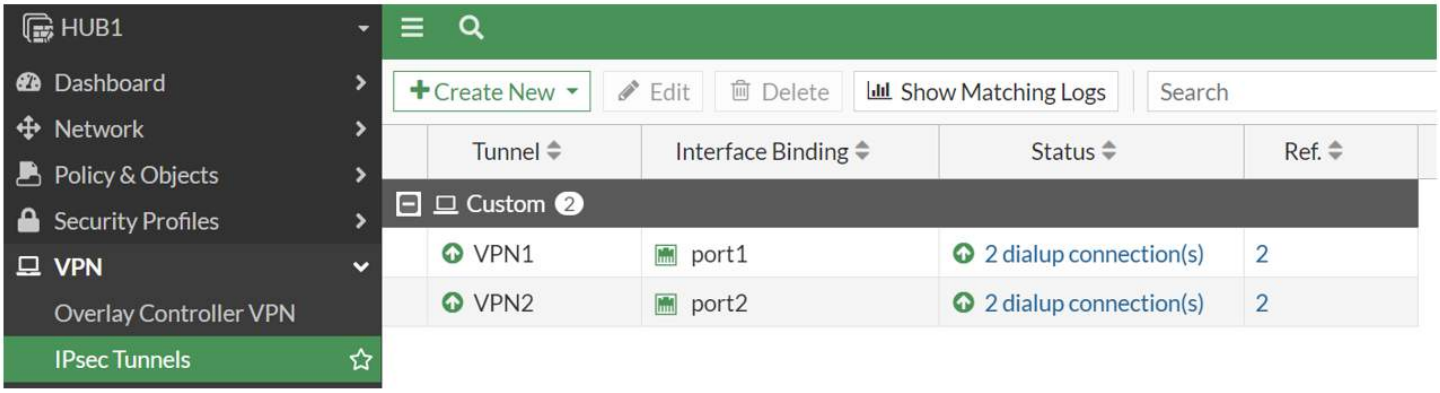




HUB IPSec Monitor

IPSec Monitor should be enabled to track the status of the tunnels at the HUB.





Since we have two branches configured, the HUB will have two dialup connections for each VPN interface as shown above.

BRANCH IPSec Command Details

```
config vpn ipsec phase1-interface
edit "HUB1-VPN1"
set interface "port1"
set ike-version 2—IKE version 2
set peertype any
set net-device enable--A dedicated interface is created for each shortcuts(ADVPN).
set mode-cfg enable-- Get ip assignment from HUB.
set proposal aes256-sha256
set add-route disable--ensures that IKE does not automatically  add a route back over the spoke, use dynamic routing (BGP).
set localid "Nakuru-Tun1"—Unique tunnel identifier, should be descriptive for easy tunnel monitoring in dashboards and reports.
set idle-timeout enable
set auto-discovery-shortcuts dependent -Delete ADVPN tunnel when the parent tunnel goes down.
set auto-discovery-receiver enable--To indicate that this IPsec tunnel wishes to  participate in an Auto-Discovery VPN (i.e., receive SHORTCUT-
OFFER)
set network-overlay enable -- required to ensure Network Overlay ID differs between tunnels.
set network-id 1 -Unique ID assignment must match between HUB and branch per tunnel.
set remote-gw 100.1.100.1- Underlay IP at Hub.
set psksecret siri12
set dpd-retrycount 2
set dpd-retryinterval 2
next
end

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

config system interface
edit "HUB1-VPN1"—Created Automatically after Ipsec phase1 configuration
set allowaccess ping- No Ip assigned since it will be relayed from HUB
set type tunnel
set interface "port1"
next
end
```

HUB IPSec Command Details

```
config vpn ipsec phase1-interface
edit "VPN1"
set type dynamic
set interface "port1—Underlay interface
set ike-version 2 —IKE version 2
set peertype any
set net-device disable--Disable creating a dedicated interface for each branch tunnel
set mode-cfg enable--Automatically offer IPs to Branches ( Command required on branches too)
set proposal aes256-sha256
set add-route disable--Ensures that IKE does not automatically add a route back  over the spoke and instead leaves routing dynamic routing (BGP)
set dpd on-idle
set localid NAIROBI-TUN1
set auto-discovery-sender enable--Indicates that when IPsec traffic transits the Hub it should  send a SHORTCUT-OFFER to the initiator of the traffic
set network-overlay enable-- Required to ensure Network Overlay ID differs between tunnels
set network-id 1- Unique per tunnel
set ipv4-start-ip 10.50.1.10
set ipv4-end-ip 10.50.1.250
set ipv4-netmask 255.255.255.0
set psksecret siri12
set dpd-retryinterval 60
next
end

config system interface
edit "VPN1"
set ip 10.50.1.1/32--The mask for the local ip can only be /32 So, the mask for the overlay subnet must be specified in 'remote-ip'.
set remote-ip 10.50.1.253/24--Any free ip in the overlay subnet, A tunnel interface  can only be a point-to-point interface /32 above.
next
end
```

BGP Configuration

iBGP is the preferred protocol and performs the following roles:

- Advertise the branch and HUB networks to neighbors.
- Using SD-WAN healthcheck tracking at the branch, advertise the quality of the link to the HUB using communities. This avoids duplicate link monitoring from the HUB to the branches.

BRANCH IN-SLA-STATE

When overlay tunnel are within the defined performance thresholds of (Latency, Jitter and Packet loss), the IN-SLA communities are set for the prefixes advertised. The required community is configured e.g. 65001:1 for Tunnel 1 when within the SLA, it is attached to the appropriate Route map which is then applied for the specific neighbor. This is done using the command below:

```
set route-map-out-preferable [route-map name]
```

For each tunnel, we define unique communities and route-maps in the following format, community **65001:X** and route-map-out-preferable of **IN-SLA-TUNX** where **X** is the tunnel number.

BRANCH OUT-OF-SLA-STATE

In the case the tunnels don't meet the defined SLA, a single out of SLA community is configured for this document, 65001:11 is used in all cases. The community is attached to an out of SLA route map using the command below for each neighbor:

```
set route-map-out [route-map name]
```

HUB ROUTE-MAP-IN

The hub forms neighborships with all branches on the overlay tunnels. At a high level, the HUB matches the configured communities defined at the branch and converts them to route tags to be used in SD-WAN rules. A single route-map is defined with multiple rules to match the different communities. The route-map is applied inbound for at the neighbor group level with the command below:

```
set route-map-in [route-map name]
```

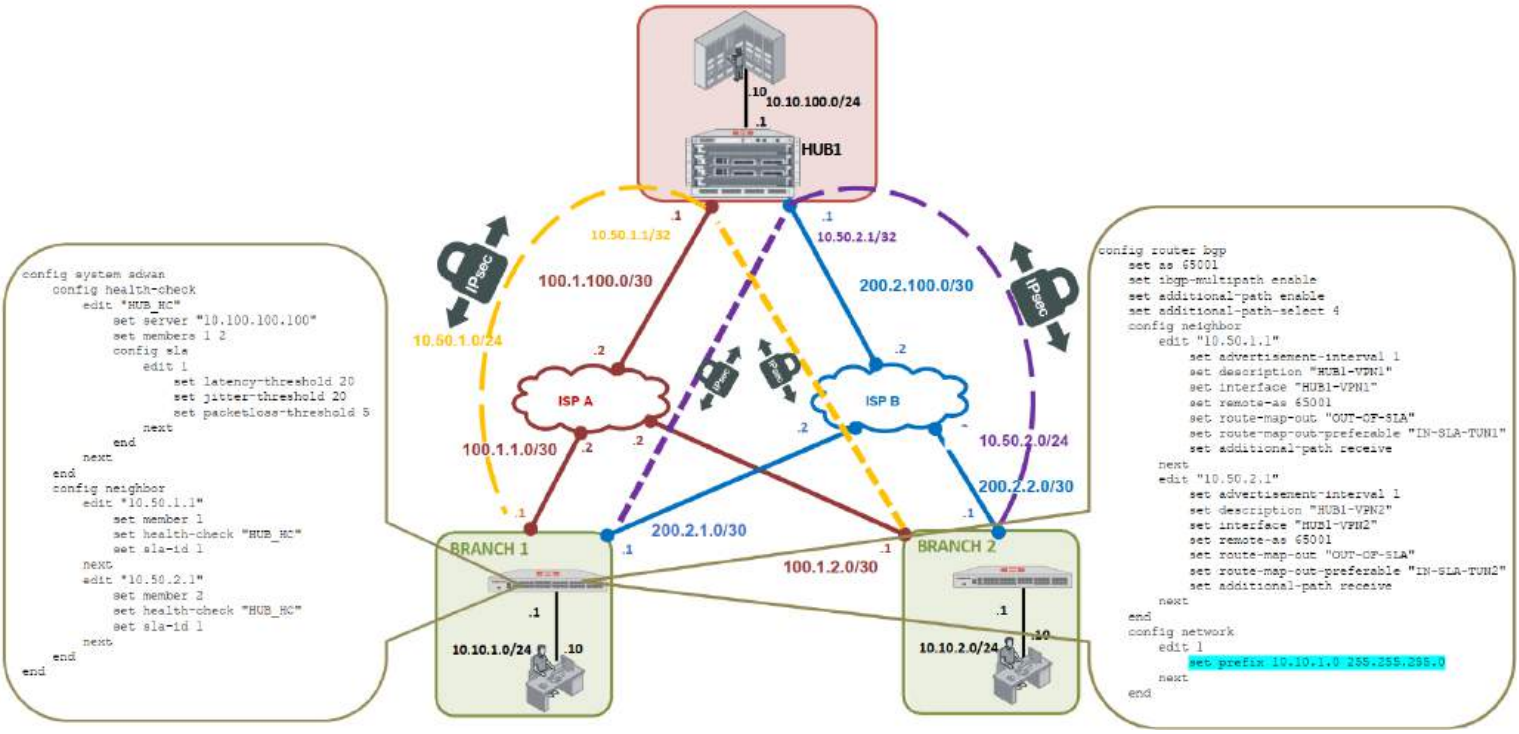
The **`neighbor-ranges`** and **`neighbor-group`** settings are configured for peering relationships to be established without defining individual peers. Branches have their tunnel interfaces configured within the range of the BGP peer and match the defined prefix under the neighbor-range defined by the command below:

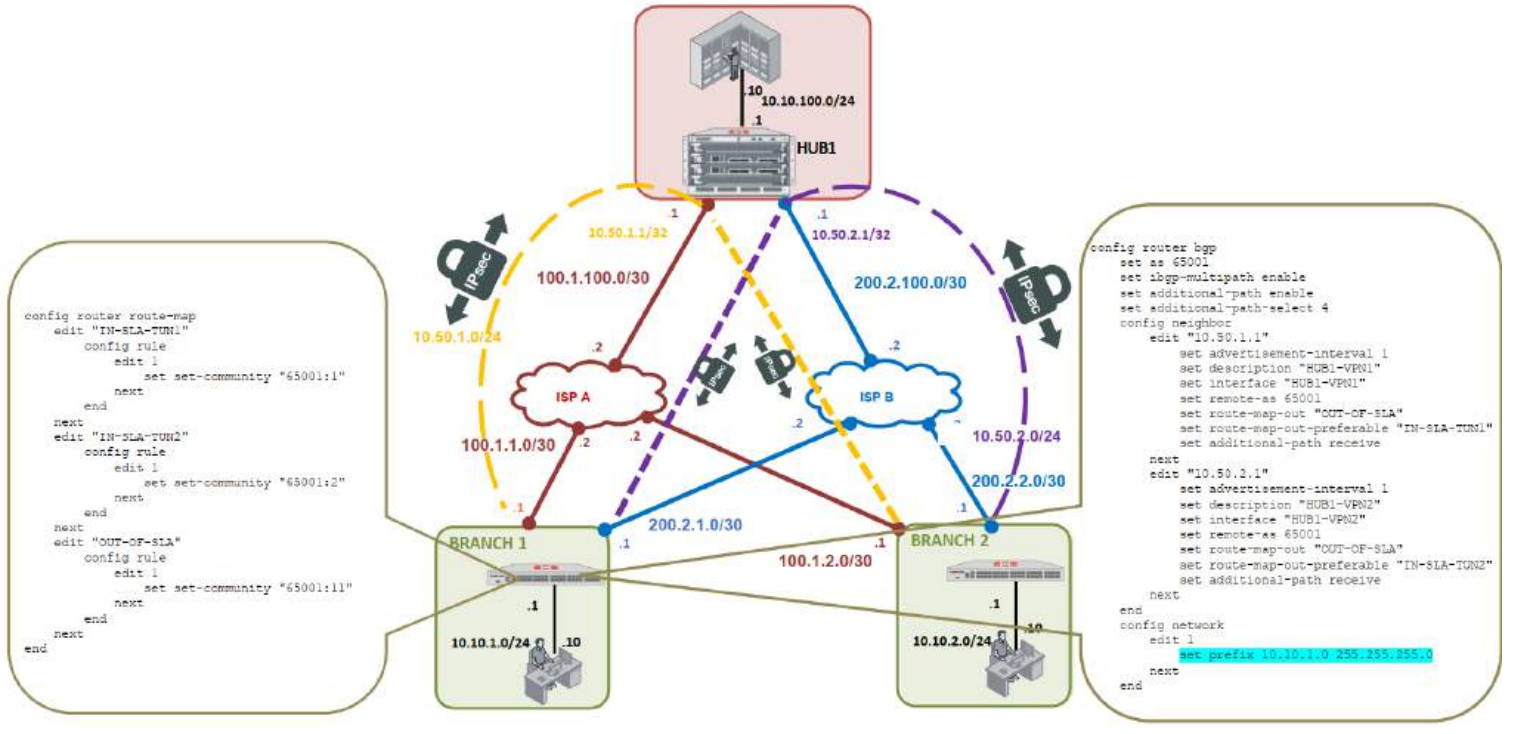
```
set prefix
```

At the branch level, prefixes are matched and advertised upstream to the HUB with the appropriate communities attached. The BGP Neighbor Group feature is used at the HUB for this peering. Branches advertises networks over each of the IBGP sessions running on the overlay networks. HUBs act as BGP Route Reflectors (RR), readvertising the prefixes to all other spokes in addition to local HUB networks.

BRANCH1 BGP Configuration

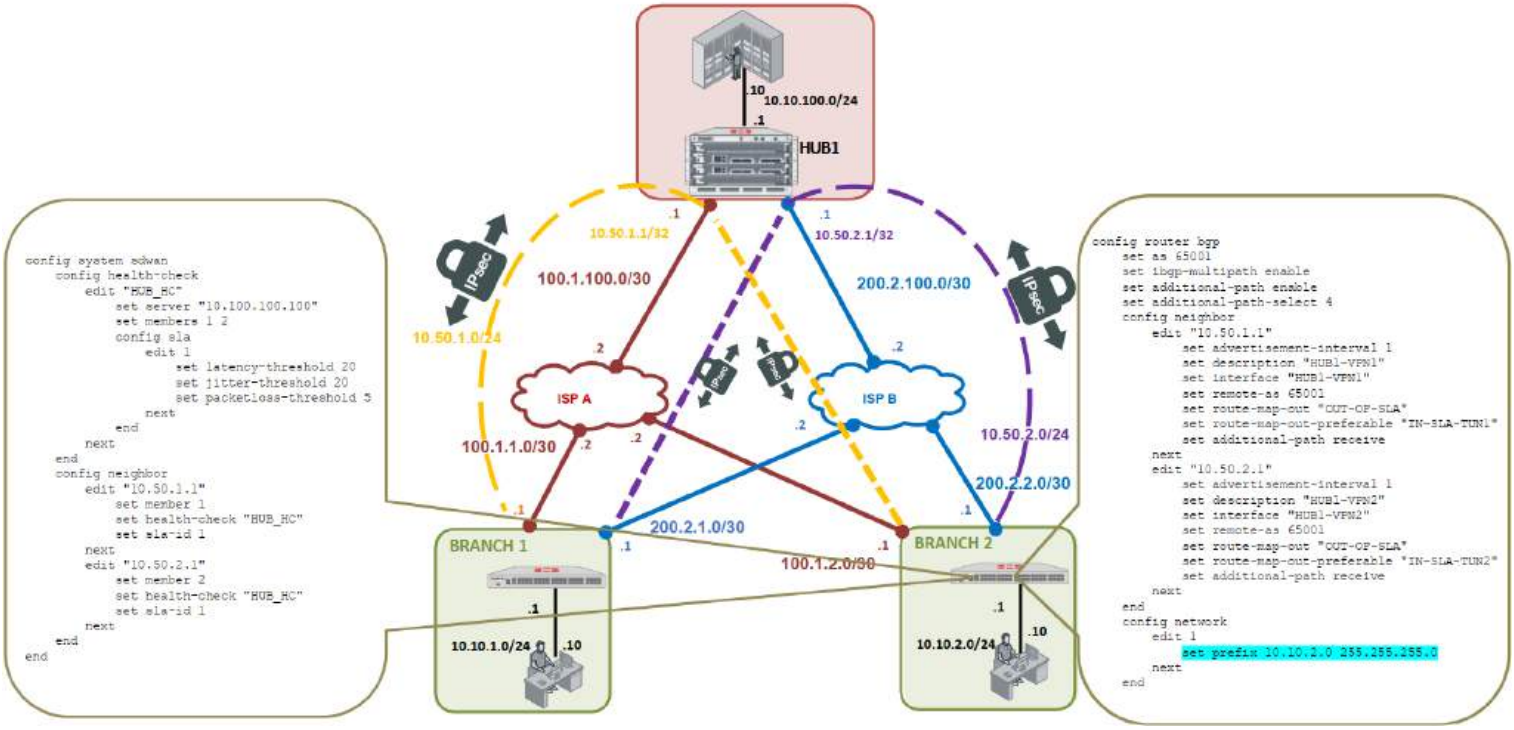
The diagrams below include the required commands to setup BGP at branch1. It includes neighbor definition, where the VPNx ip addresses are referenced as neighbor ip addresses, route-maps applied with required community settings and the SDWAN healthcheck for the HUB loopback configured.



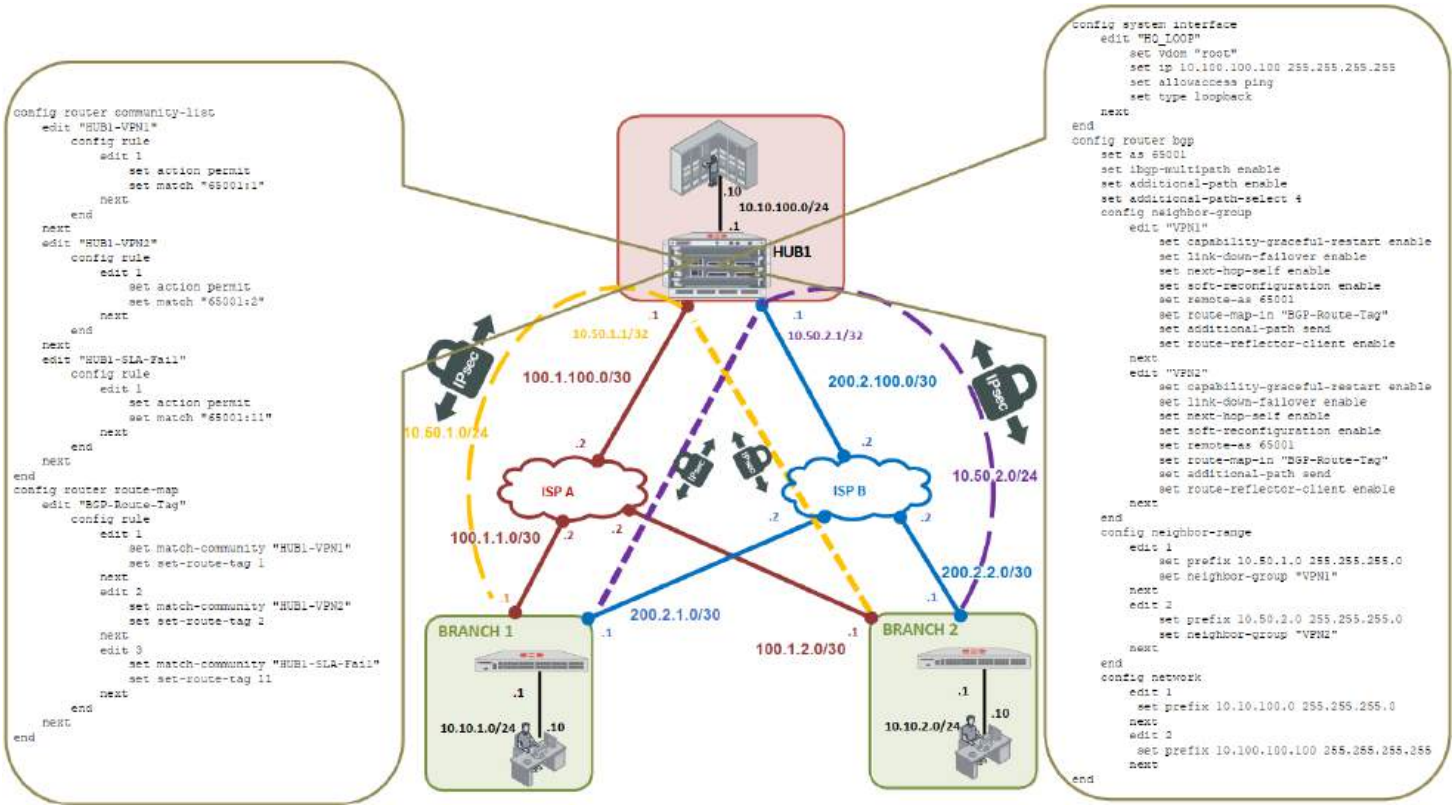


BRANCH2 BGP Configuration

This is the same configuration as branch1 or branchX.



HUB1 BGP Configuration



The HUB uses community-lists to track the communities configured at the respective branches. Once tracked, route-maps are used to match the inbound communities and set route-tags. The route-tags are then used to steer traffic using sdwan rules.

This offers greater efficiency since multiple SLAs at the HUB are not required to track all links for multiple branches since this is already shared within iBGP adverts for all subnets received on all tunnels.

A loopback interface is configured at the HUB for the branches to track link performance via ICMP. A firewall policy is required at the HUB to allow traffic inbound towards the loopback on overlay interfaces.

To enable the branch install more than one route option for a specific network, the options below are required within BGP globally.

```
set ibgp-multipath enable
set additional-path enable
set additional-path-select 4
```

To enable the HUB send multiple routes to the branches for networks received, for each neighbor group, configure:

```
set additional-path send
```

The configured route-map at the HUB is applied inbound for each neighbor group

```
set route-map-in
```

The neighbor-range option makes the configuration scalable since multiple branches match the respective tunnels for BGP based on the prefix configured.

Branches are configured as route-reflector clients within the neighbor group.

```
set route-reflector-client enable
```

At the branch level, a health check is configured to track the status of the loopback interface at the hub over HUB1-VPN1 and HUB1-VPN2. If either fails, BGP will advertise the failed community string 65001:11 to the hub. If they are within the defined SLA, the tunnel community will be advertised i.e. 65001:X where X is the tunnel number.

No health checks are needed at 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, additional settings are not needed for extra branches, making the design scalable.

BGP HUB Verification

Two paths are available for all branch networks because the multipath configuration.

```
HUB1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 10.50.1.11 (recursive is directly connected, VPN1), 03:11:41, [1/0]
                     [200/0] via 10.50.2.10 (recursive is directly connected, VPN2), 03:11:41, [1/0]
B       10.10.2.0/24 [200/0] via 10.50.1.10 (recursive is directly connected, VPN1), 02:28:35, [1/0]
                     [200/0] via 10.50.2.11 (recursive is directly connected, VPN2), 02:28:35, [1/0]
```

Route-tags are attached to all learnt networks.

```
HUB1 # get router info bgp network
VRF 0 BGP table version is 5, local router ID is 10.100.100.100
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.10.1.0/24     10.50.1.11      0              100      0        1 i <-/1>
*>i                 10.50.2.10      0              100      0        2 i <-/2>
*>i10.10.2.0/24     10.50.2.11      0              100      0        2 i <-/2>
*>i                 10.50.1.10      0              100      0        1 i <-/1>
*> 10.10.100.0/24   0.0.0.0                        100  32768        0 i <-/1>
*> 10.100.100.100/32
                    0.0.0.0                        100  32768        0 i <-/1>

Total number of prefixes 4
```

From the HUB perspective, a BGP neighbor is established for each tunnel. A total of four neighbors are established for this setup i.e. two for each branch.

```
HUB1 # get router info bgp summary

VRF 0 BGP router identifier 10.100.100.100, local AS number 65001
BGP table version is 5
1 BGP AS-PATH entries
2 BGP community entries
Next peer check timer due in 21 seconds

Neighbor   V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.50.1.10 4      65001     216     220        5    0    0 03:06:13        1
10.50.1.11 4      65001     233     239        5    0    0 03:22:02        1
10.50.2.10 4      65001     233     239        5    0    0 03:22:03        1
10.50.2.11 4      65001     215     219        5    0    0 03:06:14        1

Total number of neighbors 4
```

Since the branches are configured as RR-clients, next hops are retained in BGP adverts to other branches e.g. the network **10.10.2.0/24** is received from the neighbor **10.50.1.10** with a next hop of **10.50.1.10**. When advertised to the neighbor **10.50.1.11**, it retains the next hop of **10.50.1.10.**

```
HUB1 # get router info bgp neighbors 10.50.1.10 advertised-routes
VRF 0 BGP table version is 5, local router ID is 10.100.100.100
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.10.1.0/24     10.50.2.10                        100      0        2 i <-/2>
*>i10.10.1.0/24     10.50.1.11                        100      0        1 i <-/1>
*>i10.10.2.0/24     10.50.2.11                        100      0        2 i <-/2>
*>i10.10.100.0/24   10.50.1.1                         100  32768        0 i <-/1>
*>i10.100.100.100/32
                    10.50.1.1                         100  32768        0 i <-/1>

Total number of prefixes 5


HUB1 # get router info bgp neighbors 10.50.1.11 advertised-routes
VRF 0 BGP table version is 5, local router ID is 10.100.100.100
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.10.1.0/24     10.50.2.10                        100      0        2 i <-/2>
*>i10.10.2.0/24     10.50.2.11                        100      0        2 i <-/2>
*>i10.10.2.0/24     10.50.1.10                        100      0        1 i <-/1>
*>i10.10.100.0/24   10.50.1.1                         100  32768        0 i <-/1>
*>i10.100.100.100/32
                    10.50.1.1                         100  32768        0 i <-/1>

Total number of prefixes 5
```

```
HUB1 # get router info bgp neighbors 10.50.1.10 received-routes
VRF 0 BGP table version is 5, local router ID is 10.100.100.100
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.10.2.0/24     10.50.1.10                        100      0        0 i <-/->

Total number of prefixes 1


HUB1 # get router info bgp neighbors 10.50.1.11 received-routes
VRF 0 BGP table version is 5, local router ID is 10.100.100.100
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.10.1.0/24     10.50.1.11                        100      0        0 i <-/->

Total number of prefixes 1
```

The branches advertise the communities with each advertised network. Route-maps and community-lists are then used to match these and create route-tags. The branch neighbors are denoted as RR-clients.

```
HUB1 # get router info bgp network 10.10.1.0/24
VRF 0 BGP routing table entry for 10.10.1.0/24
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to peer-groups:
  VPN1 VPN2
  Original VRF 0
  Local, (Received from a RR-client)
    10.50.1.11 from 10.50.1.11 (200.2.1.1)
      Origin IGP metric 0, route tag 1, localpref 100, valid, internal, best
      Community: 65001:1
      Advertised Path ID: 1
       Last update: Thu Oct 24 05:22:28 2024

  Original VRF 0
  Local, (Received from a RR-client)
    10.50.2.10 from 10.50.2.10 (200.2.1.1)
      Origin IGP metric 0, route tag 2, localpref 100, valid, internal, best
      Community: 65001:2
      Advertised Path ID: 2
       Last update: Thu Oct 24 05:22:28 2024



HUB1 # get router info bgp network 10.10.2.0/24
VRF 0 BGP routing table entry for 10.10.2.0/24
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to peer-groups:
  VPN1 VPN2
  Original VRF 0
  Local, (Received from a RR-client)
    10.50.2.11 from 10.50.2.11 (200.2.2.1)
      Origin IGP metric 0, route tag 2, localpref 100, valid, internal, best
      Community: 65001:2
      Advertised Path ID: 2
       Last update: Thu Oct 24 06:05:34 2024

  Original VRF 0
  Local, (Received from a RR-client)
    10.50.1.10 from 10.50.1.10 (200.2.2.1)
      Origin IGP metric 0, route tag 1, localpref 100, valid, internal, best
      Community: 65001:1
      Advertised Path ID: 1
       Last update: Thu Oct 24 06:05:34 2024
```

BGP BRANCH VERIFICATION

Branches form neighborships with the static tunnel/overlay ip at the HUB.

```
BR1 # get router info bgp summary

VRF 0 BGP router identifier 200.2.1.1, local AS number 65001
BGP table version is 3
1 BGP AS-PATH entries
2 BGP community entries

Neighbor  V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.50.1.1 4     65001     235     229        2    0    0 03:18:35        4
10.50.2.1 4     65001     234     229        2    0    0 03:18:36        4

Total number of neighbors 2
```

```
BR2 # get router info bgp summary

VRF 0 BGP router identifier 200.2.2.1, local AS number 65001
BGP table version is 1
1 BGP AS-PATH entries
2 BGP community entries

Neighbor  V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.50.1.1 4     65001     216     212        1    0    0 03:03:19        4
10.50.2.1 4     65001     216     212        1    0    0 03:03:20        4

Total number of neighbors 2
```

Branches are configured as route reflectors and the next-hop information is retained. Two paths are available because multipath BGP is configured.

```
BR1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.2.0/24 [200/0] via 10.50.1.10 [2] (recursive is directly connected, HUB1-VPN1), 02:55:56, [1/0]
                     [200/0] via 10.50.2.11 [2] (recursive is directly connected, HUB1-VPN2), 02:55:56, [1/0]
B       10.10.100.0/24 [200/0] via 10.50.1.1 (recursive via HUB1-VPN1 tunnel 100.1.100.1), 03:15:53, [1/0]
                       [200/0] via 10.50.2.1 (recursive via HUB1-VPN2 tunnel 200.2.100.1), 03:15:53, [1/0]
B       10.100.100.100/32 [200/0] via 10.50.1.1 (recursive via HUB1-VPN1 tunnel 100.1.100.1), 03:15:53, [1/0]
                          [200/0] via 10.50.2.1 (recursive via HUB1-VPN2 tunnel 200.2.100.1), 03:15:53, [1/0]
```

```
BR2 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 10.50.1.11 [2] (recursive is directly connected, HUB1-VPN1), 03:00:19, [1/0]
                     [200/0] via 10.50.2.10 [2] (recursive is directly connected, HUB1-VPN2), 03:00:19, [1/0]
B       10.10.100.0/24 [200/0] via 10.50.1.1 (recursive via HUB1-VPN1 tunnel 100.1.100.1), 03:00:19, [1/0]
                       [200/0] via 10.50.2.1 (recursive via HUB1-VPN2 tunnel 200.2.100.1), 03:00:19, [1/0]
B       10.100.100.100/32 [200/0] via 10.50.1.1 (recursive via HUB1-VPN1 tunnel 100.1.100.1), 03:00:19, [1/0]
                          [200/0] via 10.50.2.1 (recursive via HUB1-VPN2 tunnel 200.2.100.1), 03:00:19, [1/0]
```

ADVPN

Auto-Discovery VPN (ADVPN) allows the central hub to dynamically inform spokes about a better path for traffic between two spokes. It is a Fortinet proprietary solution that achieves on-demand spoke-to-spoke circuits based on IKE & IPSec

It requires **underlay reachability between spokes** since new tunnels should form independent of the parent IPsec circuits through the hub. Reachability is natively available if internet links with static ip addresses are used at the spokes but is missing in DHCP networks e.g. 4G/5G, starlink and must be verified in MPLS networks.

Requirements for SDWAN e.g. Static addressing at the Hub, primary IPSec, iBGP etc. are still required. In summary, ADVPN is an addon feature on an existing SDWAN setup. Most of the configurations done so far are ADVPN ready but the critical commands will be reviewed with additional features.

For the correct operation of ADVPN, branches must preserve prefixes unchanged, including their original BGP next-hop values. Hence, it is impossible to replace the specific routes with summarization. To achieve this, the BGP RR function is mandatory: the gateway must reflect the original routes between the spokes without altering the original next-hop received.

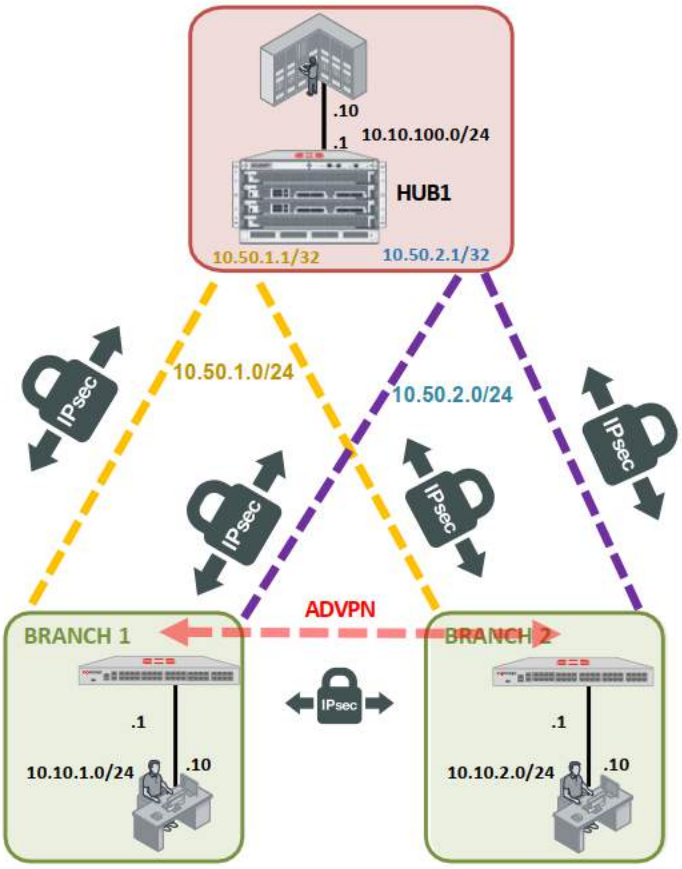
The initial communication goes through the Hub and if properly configured, the hub offers the spoke a shortcut. The spoke then negotiate a new tunnel between them on their overlay network. Once formed a shortcut tunnel would remain up even if the parent IPsec tunnel through the hub went down. This is trackable using the command below in **phase1-interface.** With the command below, the tunnel is torn down automatically when the parent tunnel goes down.

```
set auto-discovery-shortcuts dependent
```

Other ADVPN specific commands are defined in the SPOKE and HUB command section i.e. Hub IPsec details and Branch IPsec details. Traffic processing is done using firewall policies already defined for traffic to/from the Hub i.e. no policies are required for traffic to/from spokes.

ADVPN SLA monitoring of shortcuts starts 10 seconds after the shortcut is created. Shortcuts are not automatically brought down when their parent tunnel goes down. However, they can be torn down when idle. This is configured in phase1 with the command below.

```
set idle-timeoutinterval <minutes>
```



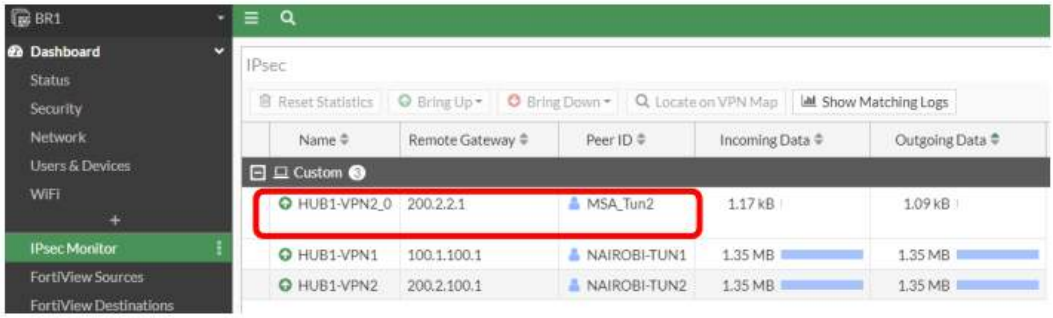
ADVPN Verification

A new connected interface of the connected overlay is created and points to the ADVPN interface. A new static route is formed with the overlay ip for the remote networks with the next hop of the peer overlay IP (Shortcut).

```
BR1 # get router info routing-table connected
Routing table for VRF=0
C       10.10.1.0/24 is directly connected, port3
C       10.50.1.0/24 is directly connected, HUB1-VPN1
C       10.50.1.11/32 is directly connected, HUB1-VPN1
C       10.50.2.0/24 is directly connected, HUB1-VPN2
C       10.50.2.10/32 is directly connected, HUB1-VPN2
                      is directly connected, HUB1-VPN2_0
C       10.50.2.11/32 is directly connected, HUB1-VPN2_0
C       100.1.1.0/30 is directly connected, port1
C       192.168.11.0/24 is directly connected, port10
C       200.2.1.0/30 is directly connected, port2
```

```
BR2 # get router info routing-table connected
Routing table for VRF=0
C       10.10.2.0/24 is directly connected, port3
C       10.50.1.0/24 is directly connected, HUB1-VPN1
C       10.50.1.10/32 is directly connected, HUB1-VPN1
C       10.50.2.0/24 is directly connected, HUB1-VPN2
C       10.50.2.10/32 is directly connected, HUB1-VPN2_0
C       10.50.2.11/32 is directly connected, HUB1-VPN2
                      is directly connected, HUB1-VPN2_0
C       100.1.2.0/30 is directly connected, port1
C       192.168.11.0/24 is directly connected, port10
C       200.2.2.0/30 is directly connected, port2
```

The new IPsec tunnel formed (ADVPN) uses the underlay network of the other branch as the peer IP. The ADVPN Shortcut tunnel & interface names nave the following syntax **<phase1name>_<index>** e.g**. HUB1_VPN2_0**




```
BR1 # get vpn ipsec tunnel summary
'HUB1-VPN2_0' 200.2.2.1:0  selectors(total,up): 2/2  rx(pkt,err): 8/0  tx(pkt,err): 4/8
'HUB1-VPN1' 100.1.100.1:0  selectors(total,up): 1/1  rx(pkt,err): 35341/0  tx(pkt,err): 35342/2
'HUB1-VPN2' 200.2.100.1:0  selectors(total,up): 1/1  rx(pkt,err): 35371/0  tx(pkt,err): 35370/17

BR2 # get vpn ipsec tunnel summary
'HUB1-VPN2_0' 200.2.1.1:0  selectors(total,up): 2/2  rx(pkt,err): 13/0  tx(pkt,err): 16/2
'HUB1-VPN1' 100.1.100.1:0  selectors(total,up): 1/1  rx(pkt,err): 33324/0  tx(pkt,err): 33325/0
'HUB1-VPN2' 200.2.100.1:0  selectors(total,up): 1/1  rx(pkt,err): 33340/0  tx(pkt,err): 33343/2
```

SLA monitoring is done with icmp traffic between the overlay IP addresses of the two branches. ICMP must be allowed on the overlay ip address for the SLA to work.

```
BR1 # diagnose sys sdwan health-check
Health Check(HUB_HC):
Seq(1 HUB1-VPN1): state(alive), packet-loss(0.000%) latency(4.761), jitter(1.343), mos(4.401), bandwidth-up(999999), ba
ndwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(2 HUB1-VPN2): state(alive), packet-loss(0.000%) latency(4.171), jitter(1.083), mos(4.401), bandwidth-up(999999), ba
ndwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(2 HUB1-VPN2_0): state(alive), packet-loss(0.000%) latency(4.835), jitter(1.839), mos(4.400), bandwidth-up(1000000),
 bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
```

```
BR2 # diagnose sys sdwan health-check
Health Check(HUB_HC):
Seq(1 HUB1-VPN1): state(alive), packet-loss(0.000%) latency(5.351), jitter(2.101), mos(4.400), bandwidth-up(999999), ba
ndwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(2 HUB1-VPN2): state(alive), packet-loss(0.000%) latency(4.738), jitter(1.994), mos(4.400), bandwidth-up(999999), ba
ndwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(2 HUB1-VPN2_0): state(alive), packet-loss(0.000%) latency(4.513), jitter(2.772), mos(4.399), bandwidth-up(1000000),
 bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
```

After configuring IPsec and BGP, SDWAN rules and policies are configured at the HUB and branches GUI. This option is only feasible for installations with few branches. The next chapters will focus on deployment using FortiManager.

Chapter 2 questions

1. Which is the recommended order to configure the key components of SDWAN.
   a) IPsec> BGP> SDWAN
   b) IPsec> SDWAN>BGP
   c) BGP> IPsec> SDWAN

2. Which component must match between the HUB and a spoke/branch.
   a) Local ID
   b) Network ID
   c) All the above

3. What identifier is set at the branch with the advertised prefixes.
   a) Route map
   b) Community
   c) Route tag

4. Which routing protocol runs between the HUB and branches.
   a) iBGP
   b) EBGP
   c) Any of the above

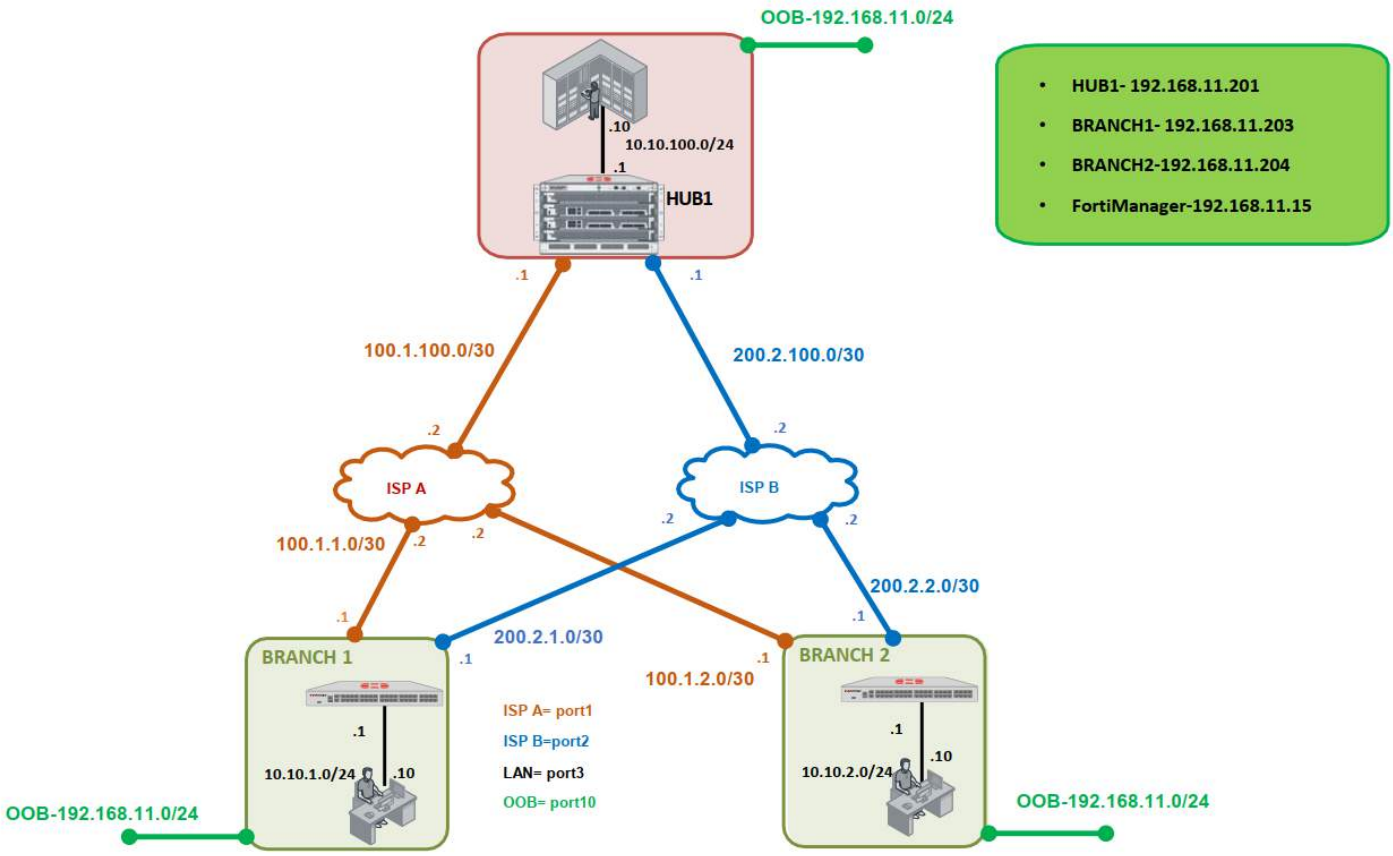
5. Which IP address do the branches reference as the bgp neighbor at the HUB.
   a) Overlay static interface ip
   b) Underlay static IP
   c) Loopback at the HUB

6. What attribute is used at the HUB as the destination when defining SDWAN rules to access branch networks.
   a) Route-tag
   b) Community
   c) Destination network

7. What SDWAN steering strategy is used to direct traffic to the branch with SDWAN rules.
   a) Best quality
   b) Manual
   c) Lowest cost

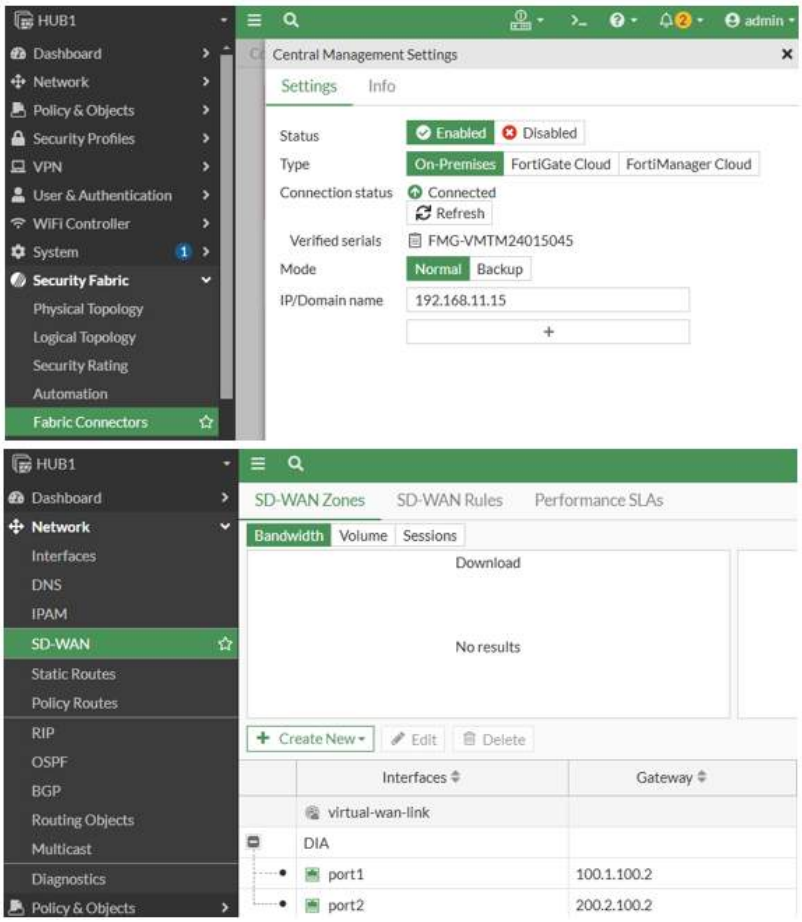
Chapter 3: FortiManager Single Hub with Dual ISP Underlays

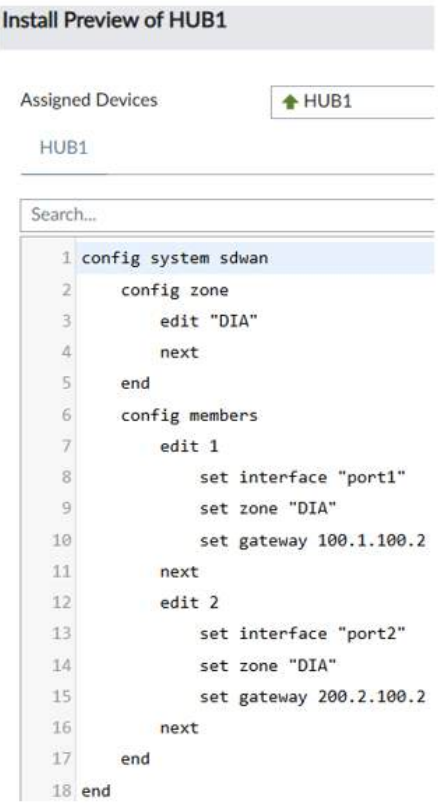
The topology used for this setup is the same as the one used in the GUI & CLI section with the addition of FortiManager VM as for centralized management. FortiManager is deployed at HUB1.



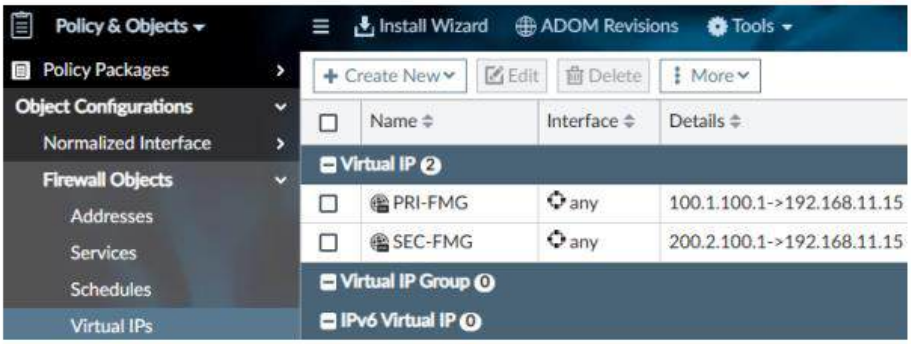
Provision the HUB for FortiManager Access.

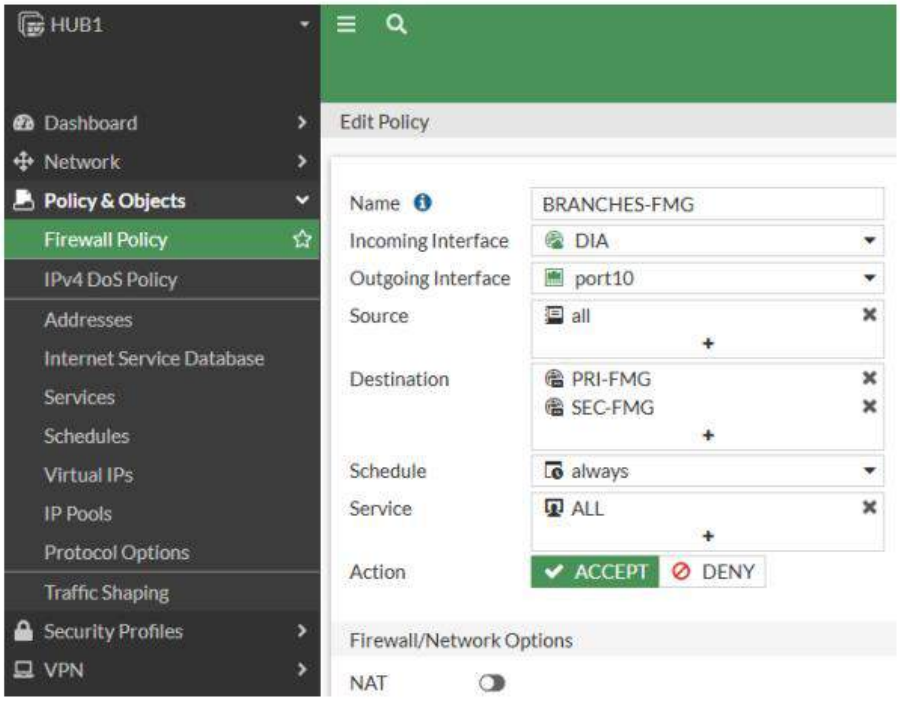
The HUB connects to FortiManager via the OOB network. Configure the underlay SDWAN settings to be used for DIA. The SDWAN configuration is done from FMG but could be done from the HUB GUI as long as the zone name is the same as what would be later used in FortiManager templates to avoid provisioning errors.



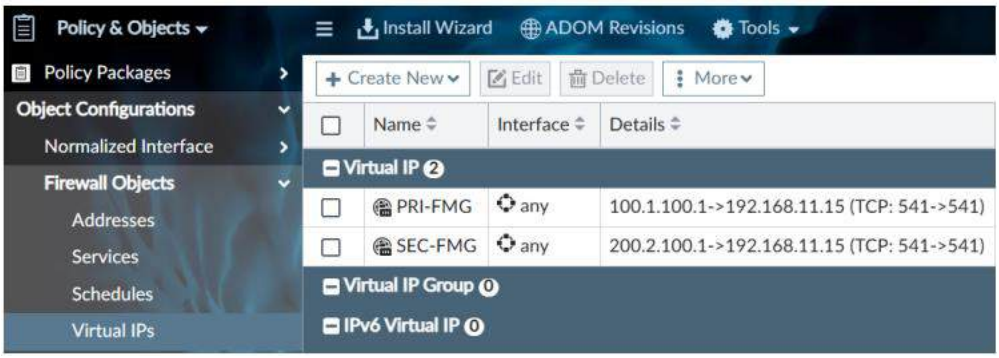


The branches will access FMG via the public IPs at the HUB. Two NAT rules and a firewall policy are required to allow the inbound connections. The inbound rule is applied on the underlay network to allow branch devices to access FortiManager



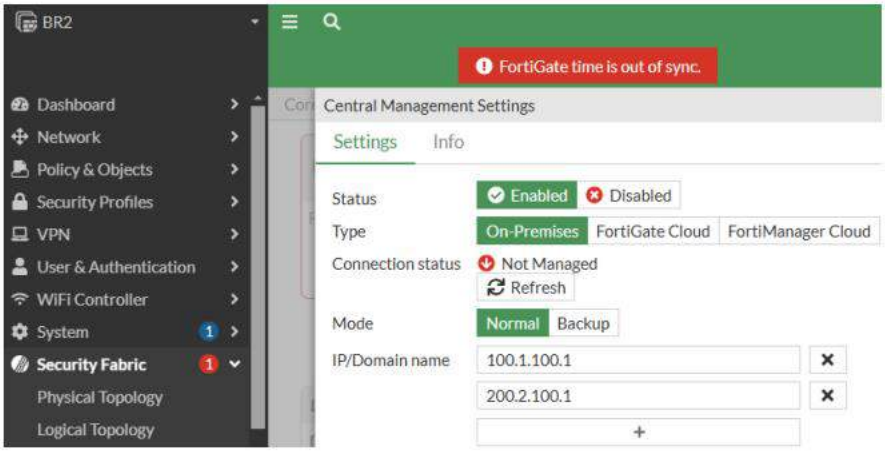


If a single public IP is available at the primary and secondary links, port address translation (PAT) must be used to avoid NAT directing all inbound traffic to FortiManager, blocking all branch devices dialing in over IPsec since all traffic would be directed to FortiManager including IPsec terminations. For PAT, use TCP port 541 as shown below to allow other connections on the IP inbound i.e. IPsec terminations.

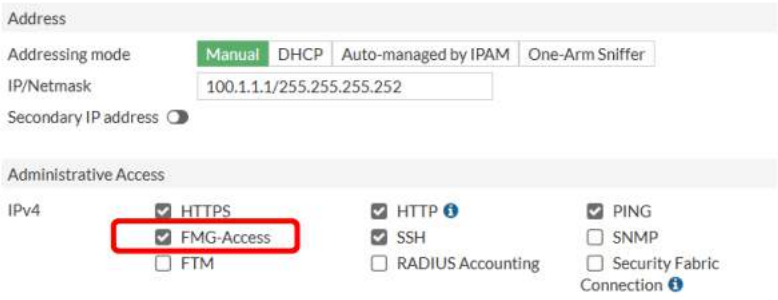


Since there are two ISP links at the HUB, the branches are configured to access both for redundancy. When multiple addresses are listed, the FortiGate attempts to establish the FGFM tunnel using the first IP listed, and if it is unreachable, it tries each subsequent IP until the tunnel is established.

```
BR1 (central-management) # set fmg
<fortimanager    FQDN/IP>       Up to 10 entries


BR1 (central-management) # set fmg 100.1.100.1 200.2.100.1
```

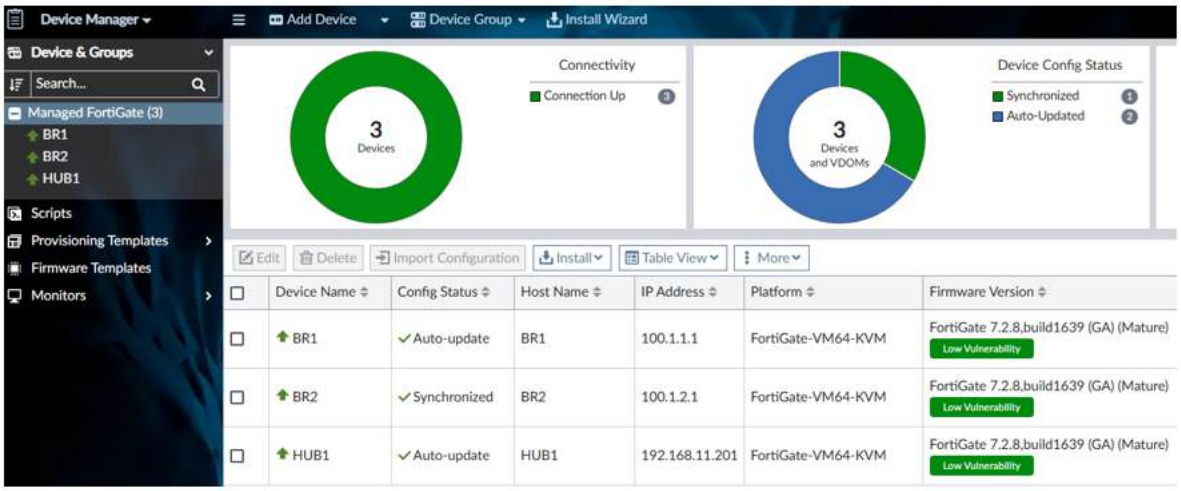


Only one address is used to establish the FGFM tunnel at a time. In FortiManager-HA, when listing multiple management addresses, the first address defines the Primary device and the second address is the Secondary device. This design supports a maximum of 10 FortiManager instances that is adequate for most designs. FMG-Access should be enabled on the WAN interfaces at the branches to allow registration to FortiManager.



To check registration status, check the device manager section under Managed FortiGate. This is also visible on FortiManager CLI using the command:

```
Diagnose fgfm session-list
```




```
FMG # diagnose fgfm session-list
        HUB1(163) sn(FGVM01TM24004958) ip(192.168.11.201)
                state(tunnel) tunnel(169.254.0.2) uptime:Thu Oct 24 20:27:08 2024
        BR1(175) sn(FGVM01TM24004960) ip(100.1.1.1)
                state(tunnel) tunnel(169.254.0.3) uptime:Thu Oct 24 20:43:57 2024
        BR2(187) sn(FGVM01TM24004961) ip(100.1.2.1)
                state(tunnel) tunnel(169.254.0.4) uptime:Thu Oct 24 21:01:57 2024
        Session count = 3 (tunnel 3)
```

```
BR1 # diagnose fdsm central-mgmt-status
Connection status: Up
Registration status: Registered
Serial: FMG-VMTM24015045
```

```
BR2 # diagnose fdsm central-mgmt-status
Connection status: Up
Registration status: Registered
Serial: FMG-VMTM24015045
```

The branches register to the first FortiManager IP to be configured but failover to the next available IP when it is unreachable.

The image below shows an outage of FortiManager on primary ip 100.1.100.1 and failover to the second ip address 200.2.100.1. For this lab, both IP addresses lead to the same FortiManager instance providing link redundancy at the HUB.

Branch devices are grouped in one device group for centralized management of template Groups and Firewall policies.

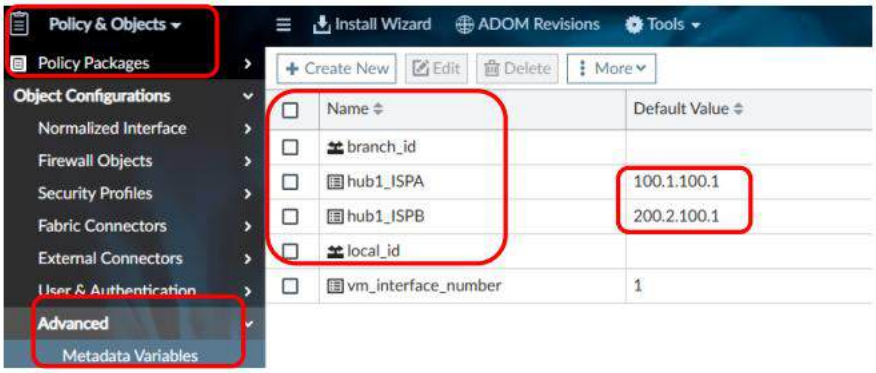
Metadata Variables

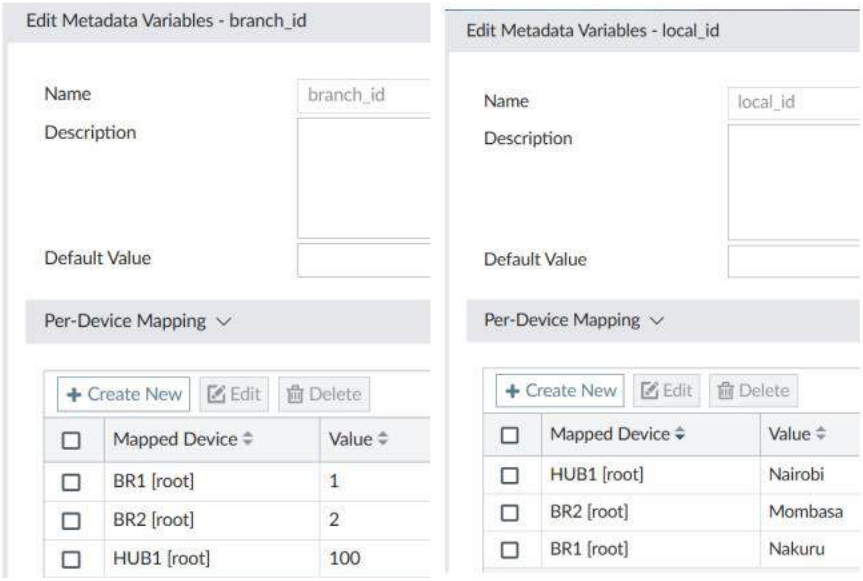
Metadata Variables are defined under Policy & Objects in the advanced section. The local-id Metadata Variable is created to define local identifiers. For this lab, we will use **Nakuru** and **Mombasa** as the branches and **Nairobi** as HUB1.

Once created using per device mapping, it is available for use in templates by prepending the **$** symbol**.**

Branch1 and Branch2 are defined with branch_ids of **1&2** respectively while HUB1 has **100**. This will be used in BGP & IPSec Templates.

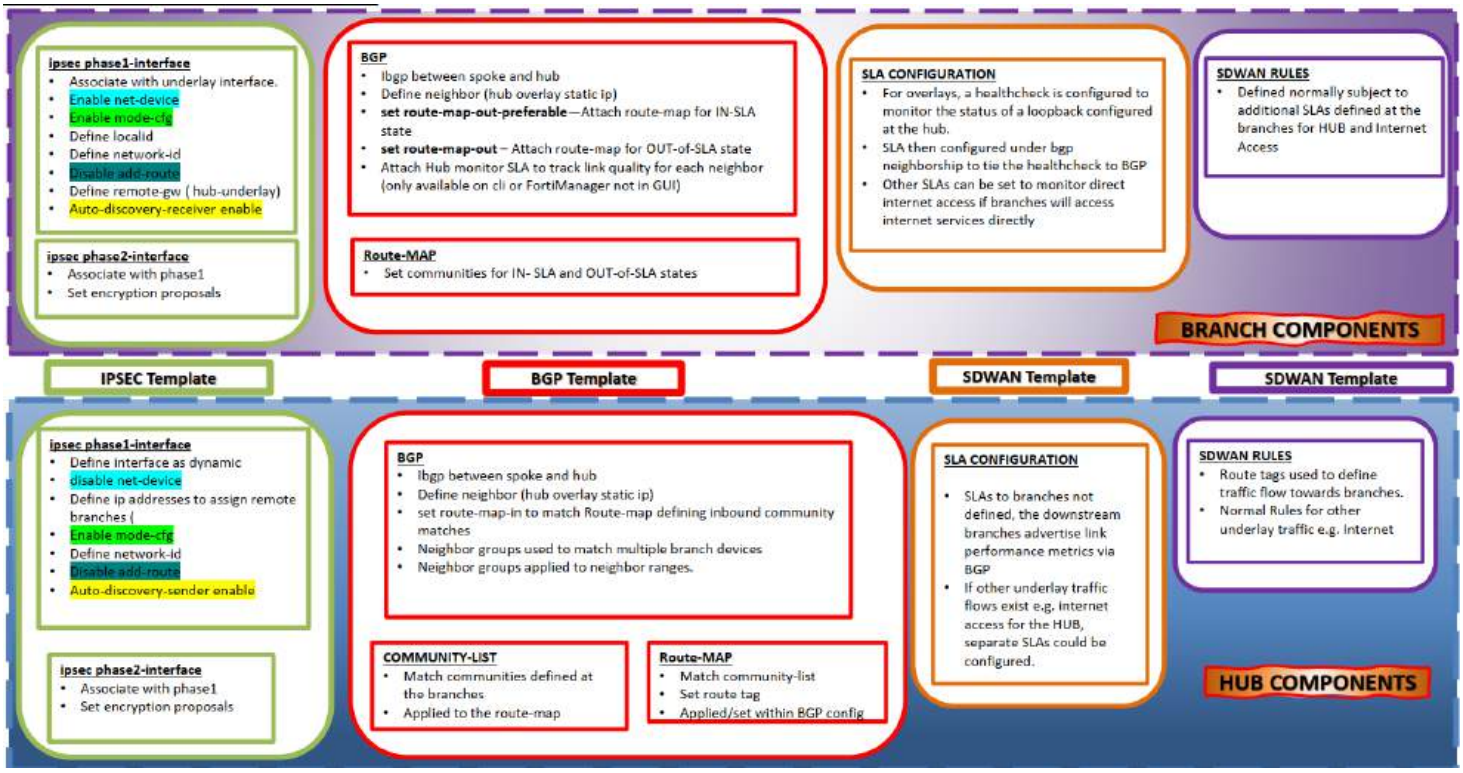
hub1_ISPA and hub2_ISPB underlay IPs are also defined for use by branches to reference remote gateways at the HUB. These are defined as default values.





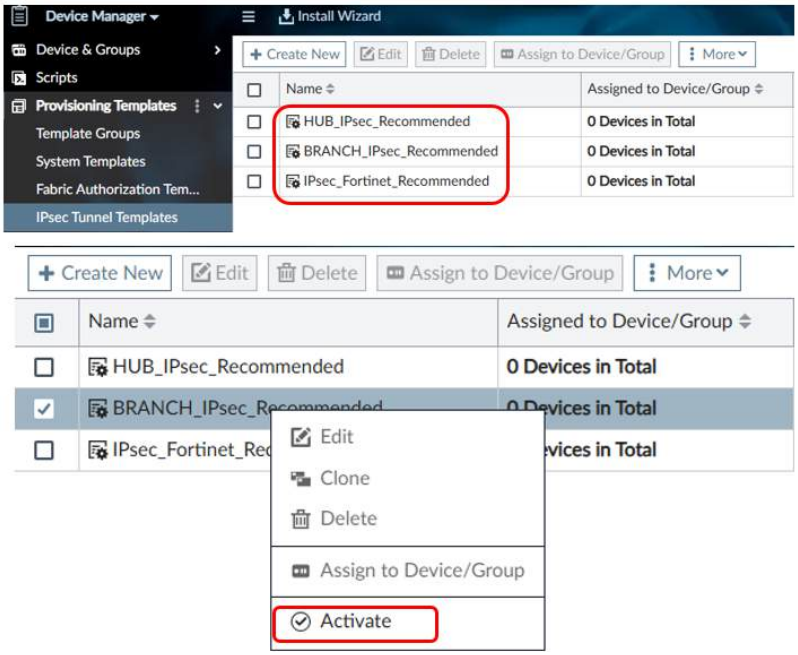
SD-WAN FortiManager role mapping.

The configurations are done using three main templates i.e. **IPSEC, BGP and SDWAN**.



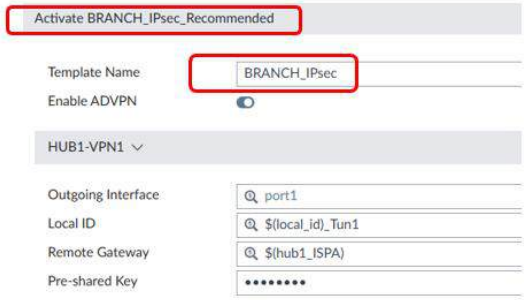
IPSec Templates

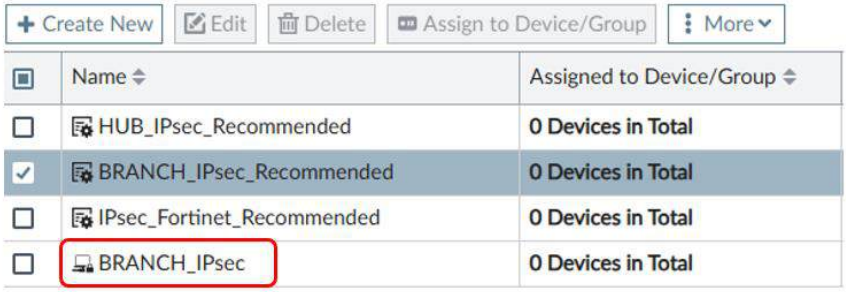
The first template to be created is the IPSec template. This is the same logic we used in the CLI/GUI section. Three default templates are available and those relevant to SD-WAN are the Branch and Hub recommended templates.



You right click, activate and rename a template. Once activated, define the outgoing local interface, local-id, remote-gateway (undelay at hub) and preshared key.

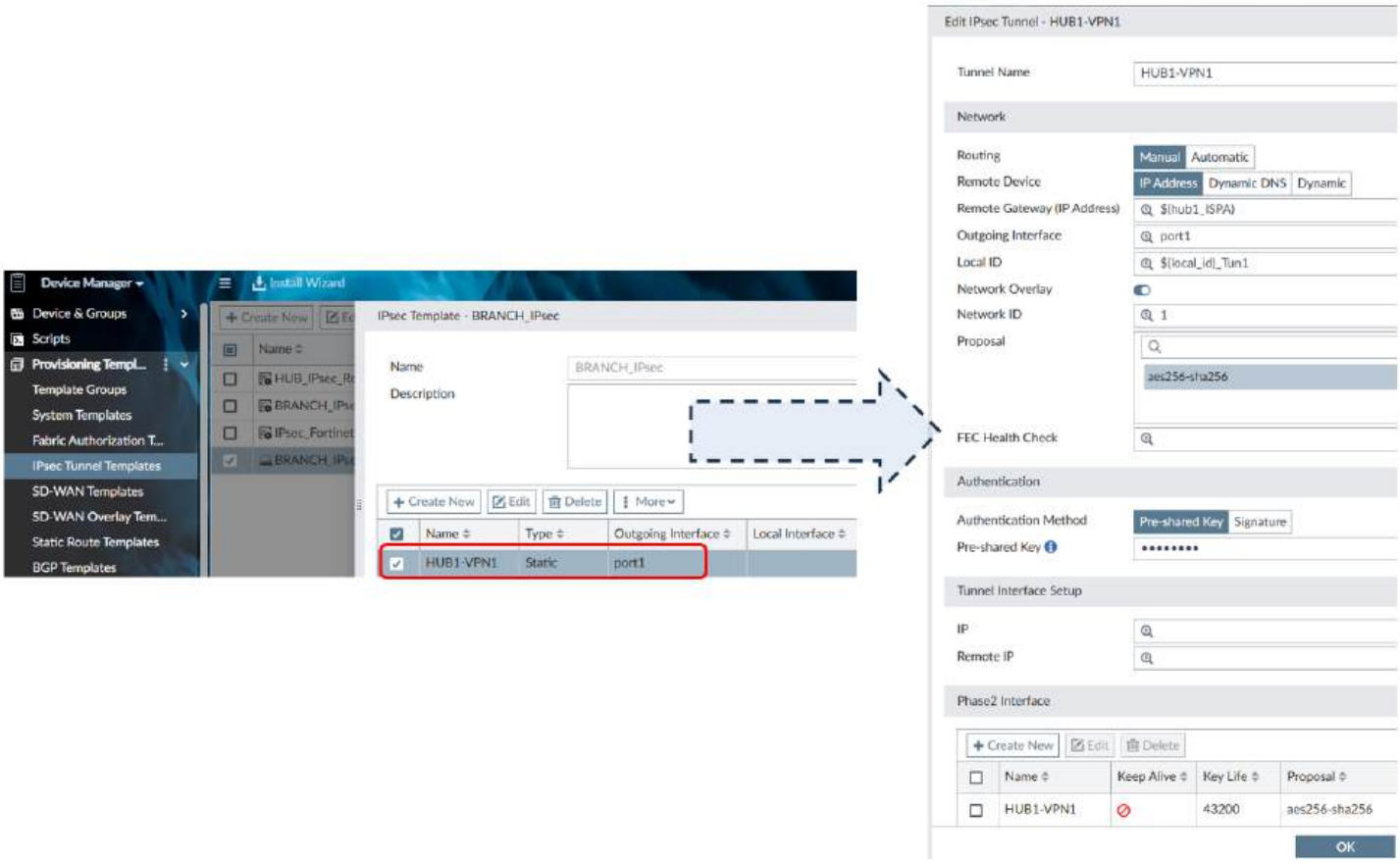
It is recommended to define the remote gateway as a metadata variable for easy configuration and greater accuracy.





The activated Branch_IPsec template is renamed, the local primary gateway interface (port1), Local ID (Metadata Variable), remote Gateway (Metavariable) and a PSK. At this point, ADVPN at the branch level is also activated.

The IPsec template creates all required configurations for phase1 and 2 .To be more descriptive, each tunnel will have a unique Local_ID e.g. Sitename_Tun1 for the first tunnel.

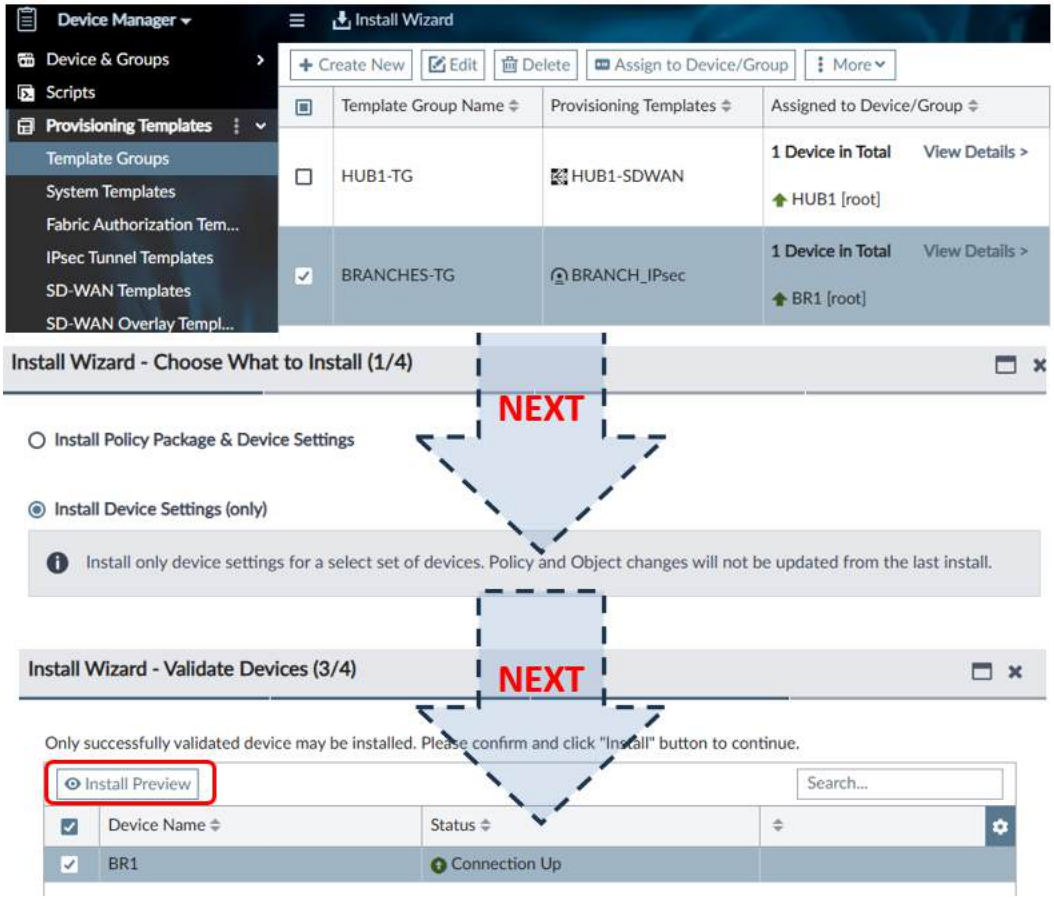


Branch Template Group

IPSec templates are attached to **Template groups** which are assigned to the devices. For initial modelling, we assign the template to one device to optimize all applicable settings before adding all branch units. The installation Wizard is used to install the configured IPSec template. These are considered **device settings** (interface, routing etc.).

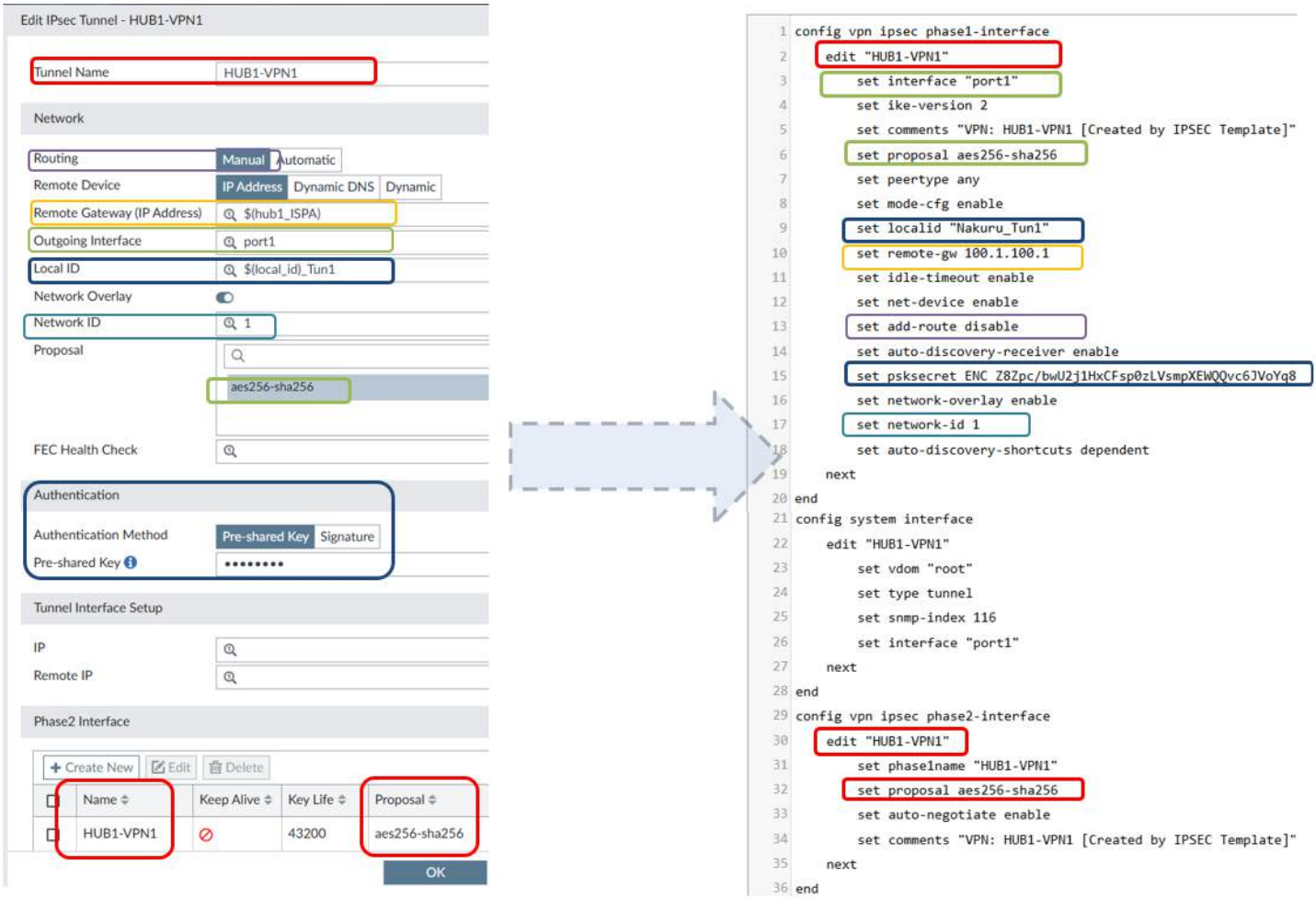
The target device should be available (connection up). During the initial modelling, use the install preview to review the configuration and check compatibility with the target device. Any incompatibility will be flagged at this point.

When running the install preview, you might get the error **invalid preview file** or **No preview** when sending device settings only. To get a detailed error report, select **install Policy Package & Device settings** which has detailed error reporting compared to **install device setting (only).** If no policy package is attached to the device, create an empty package with implicit deny for testing only, this should be detached once the errors are identified to avoid overwriting firewall policies on the remote unit.



Branch1 IPSec HUB1-VPN1 Config Preview.

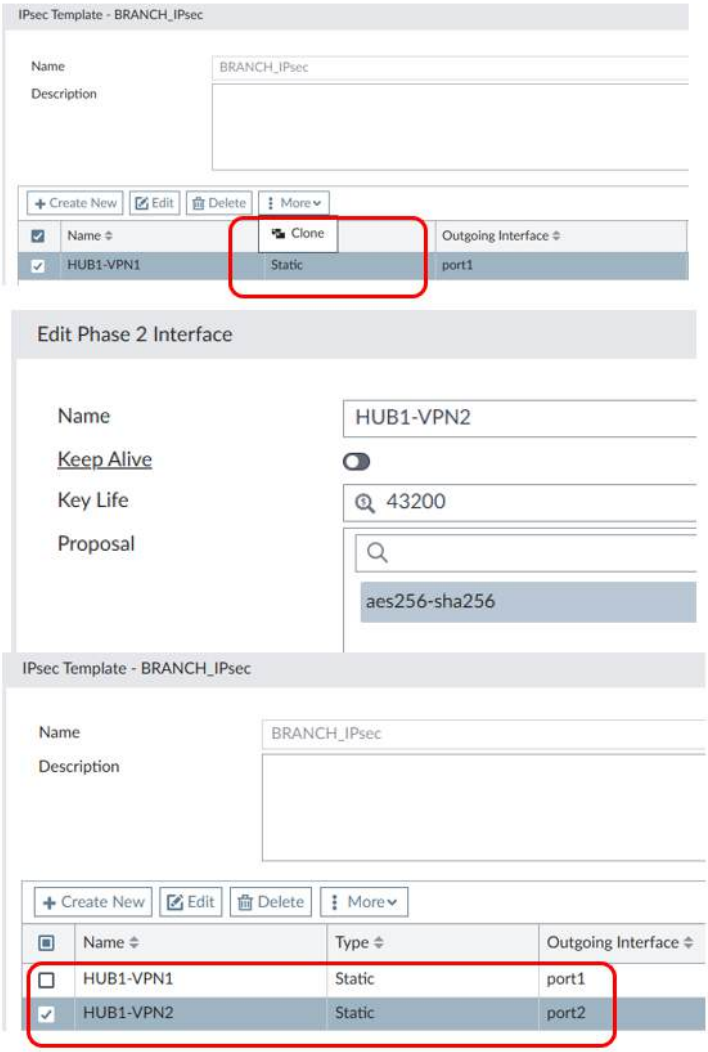
The IPSec template creates the commands shown to be installed on the branch firewall. It creates all required phase1 and phase2 configuration with the required settings. When pushed to branch1, the tunnel is configured but it will be down because the required setup at the Hub and the inbound and outbound firewall policies or done SDWAN zone grouping are not yet configured

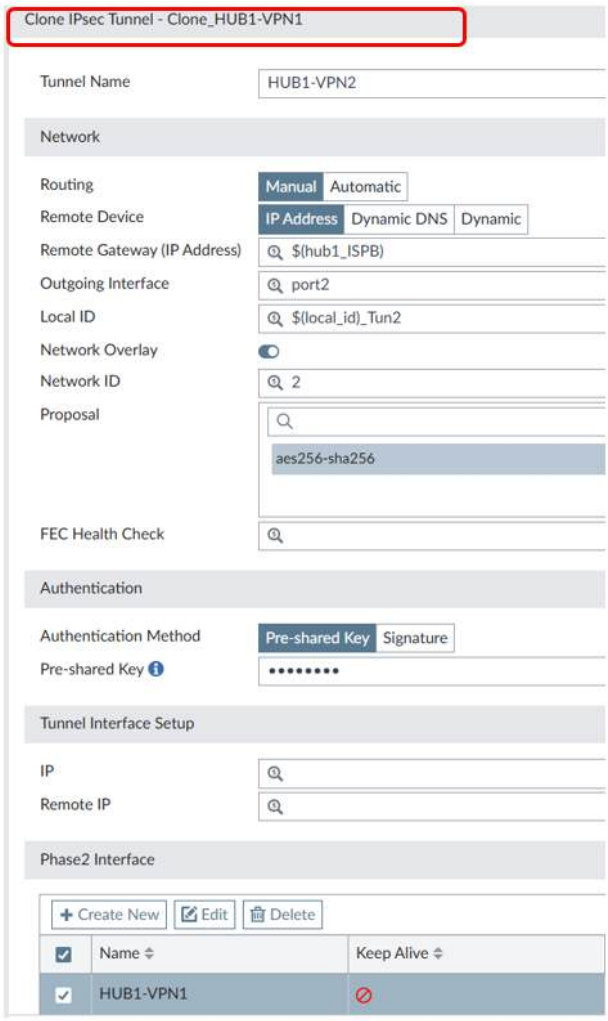


Branch1 IPSec HUB1-VPN2 Configuration

The second tunnel is configured by cloning the one created from the IPsec Template. Once cloned, settings are customized to match the second tunnel.

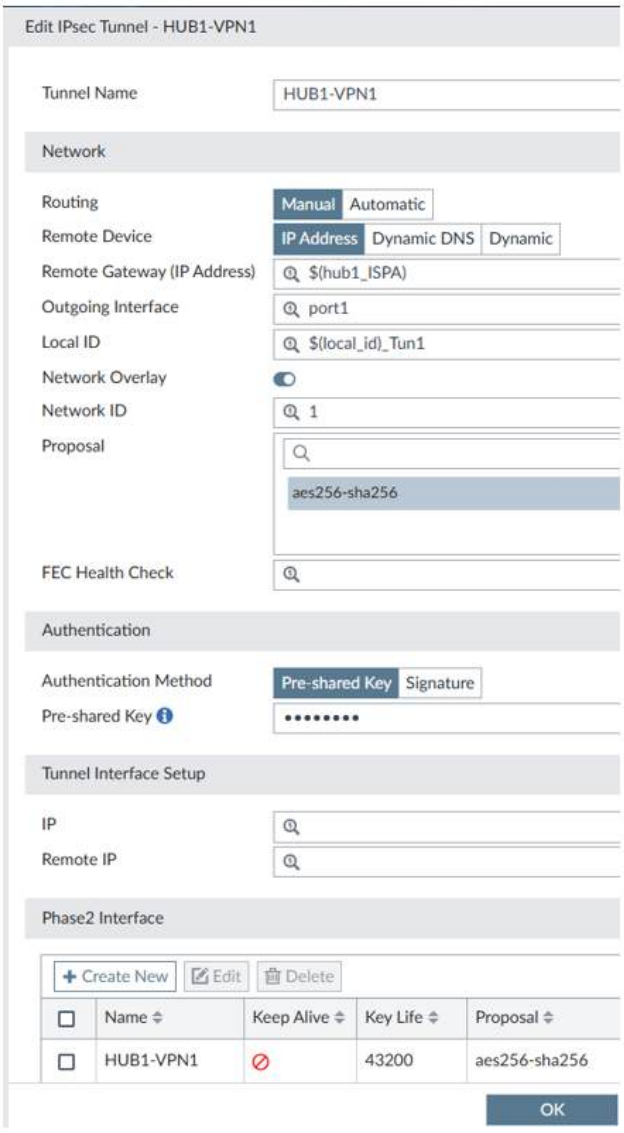
The tunnel name, Gateway, outgoing interface, local ID, network ID and phase2 name are customized to match the second tunnel. All other settings remain the same. Any additional tunnels (3,4..) are made in the same way by creating a clone of the existing tunnels.

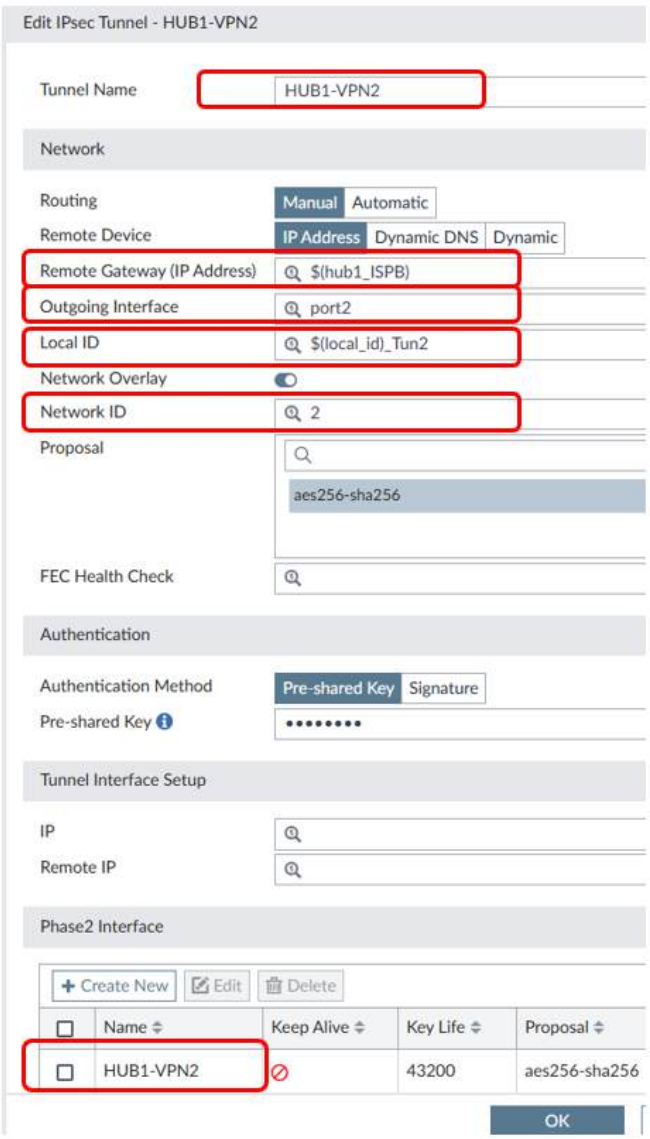




Branch1 HUB1-VPN1 & HUB1-VPN2 Comparison

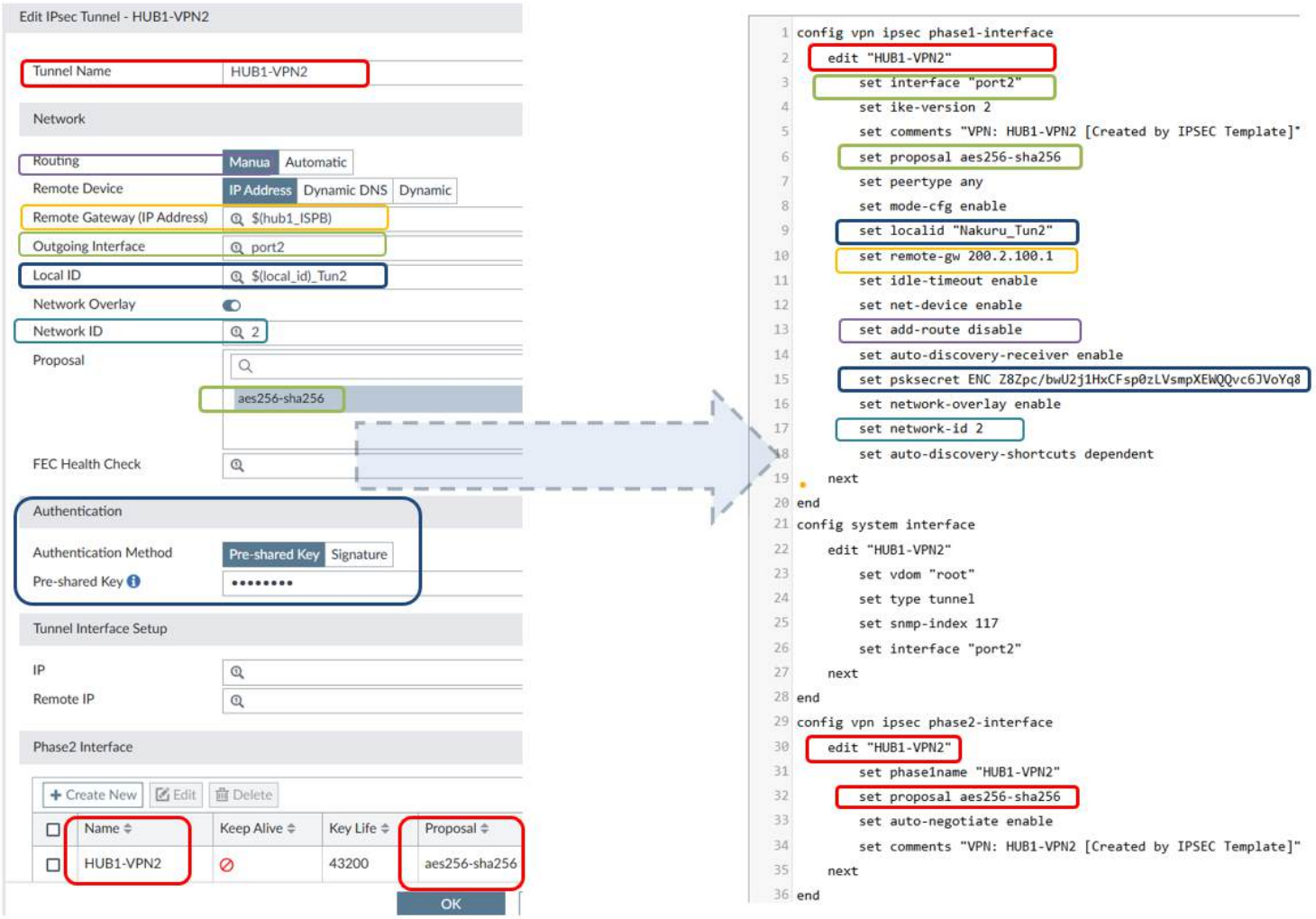
The variations between the VPNs are highlighted.





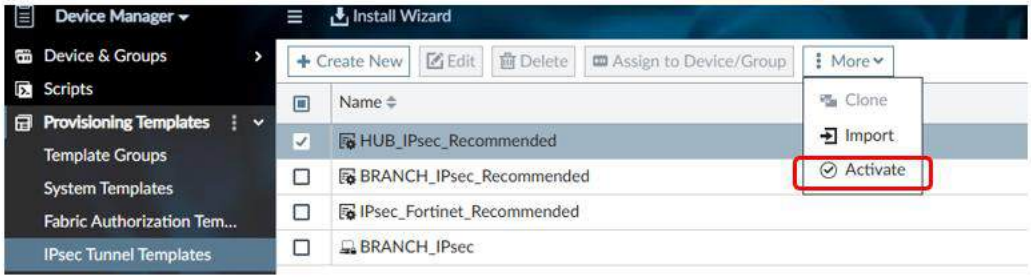
Branch1 IPSec HUB1-VPN2 Config Preview

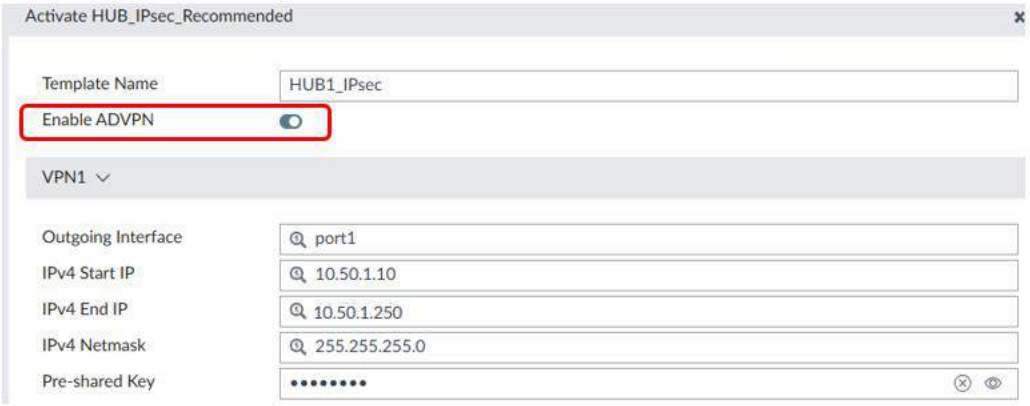
The second tunnel settings are then pushed to the branch firewall with the custom settings as shown below.

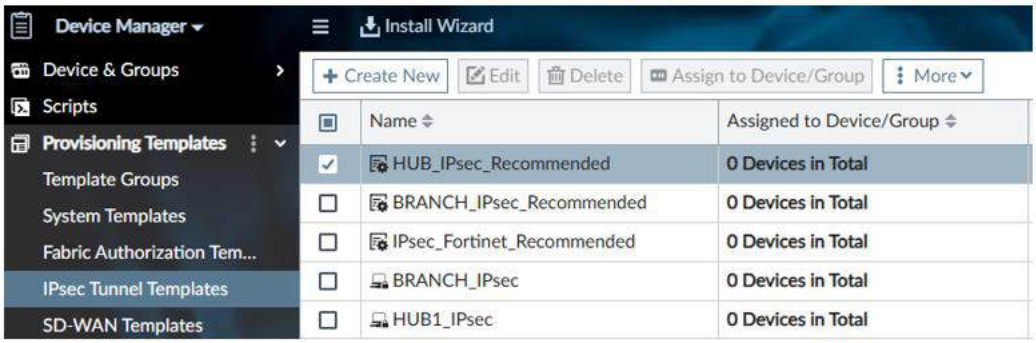


HUB IPSec Template

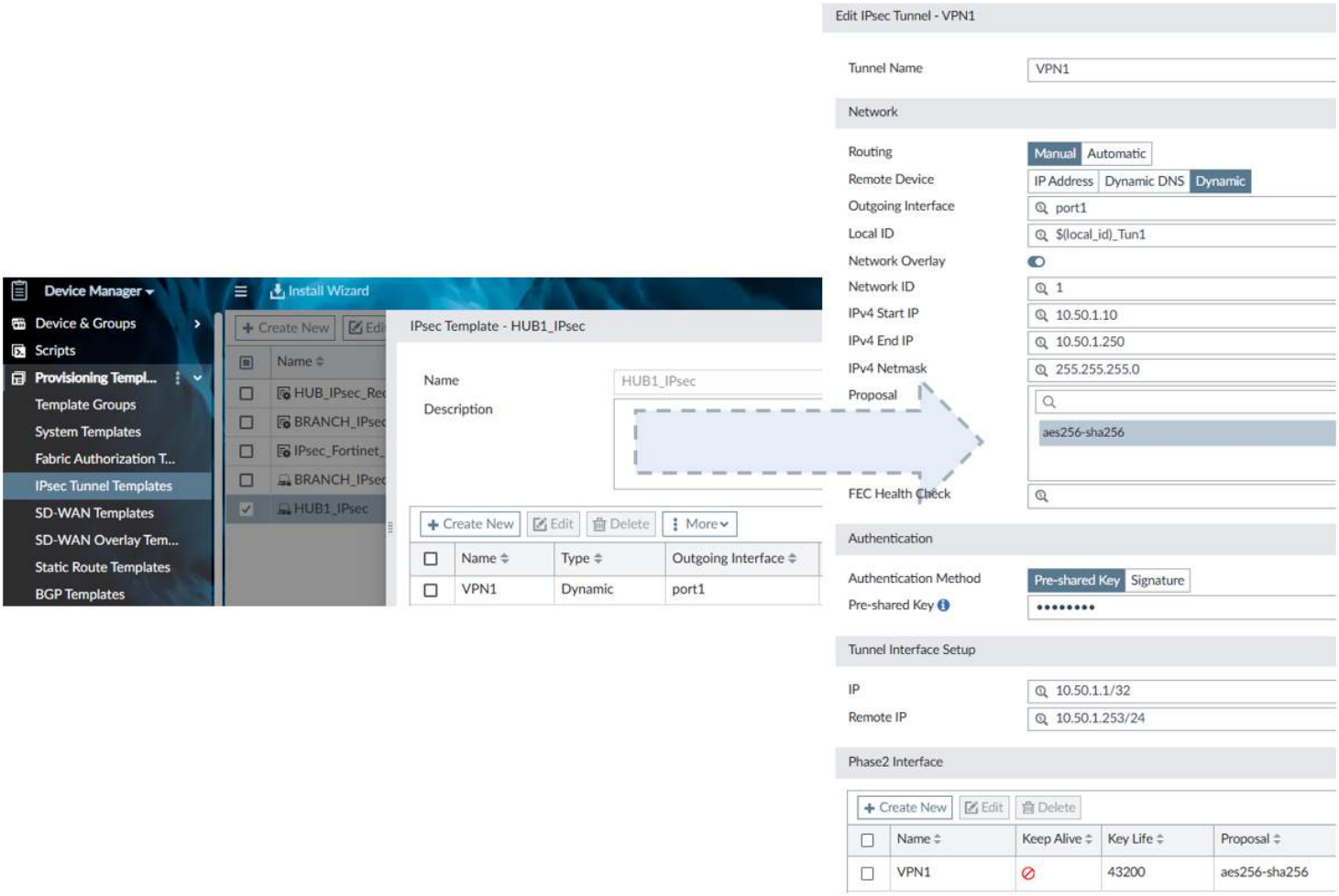
It is activated from the Hub_IPSec _Recommended template. Define the Underlay port and the scope to be used in assigning addresses for Tunnel1 (VPN1) and ensure HUB ADVPN setting are also activated at this point if required.



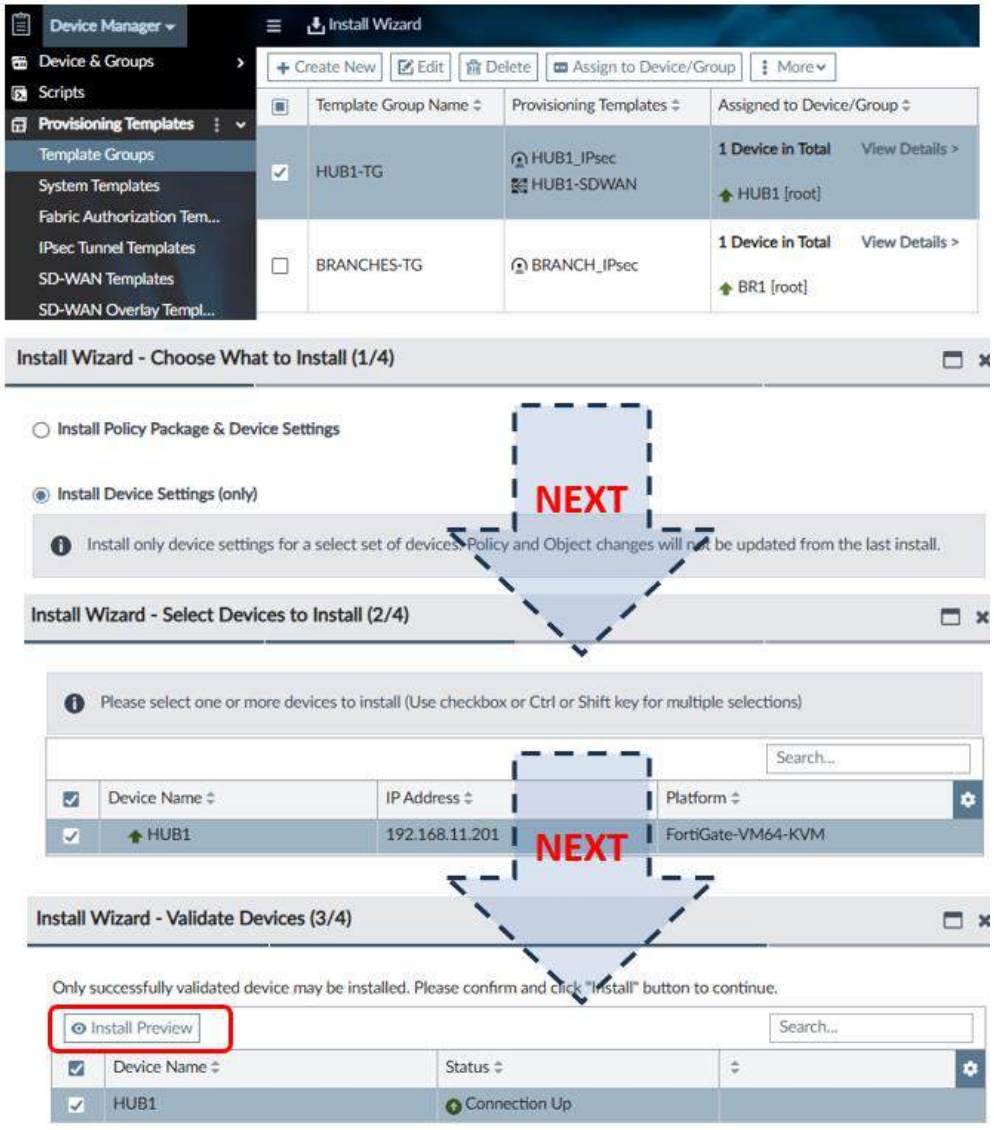




The HUB template creates all required settings for the HUB IPsec. The HUB acts as the DHCP server for all remote sites dialing on respective tunnels. For each tunnel, the interface ip address is defines as a **/32** host and the remote IP configured with the /24 network or other appropriate mask to define the network scope.

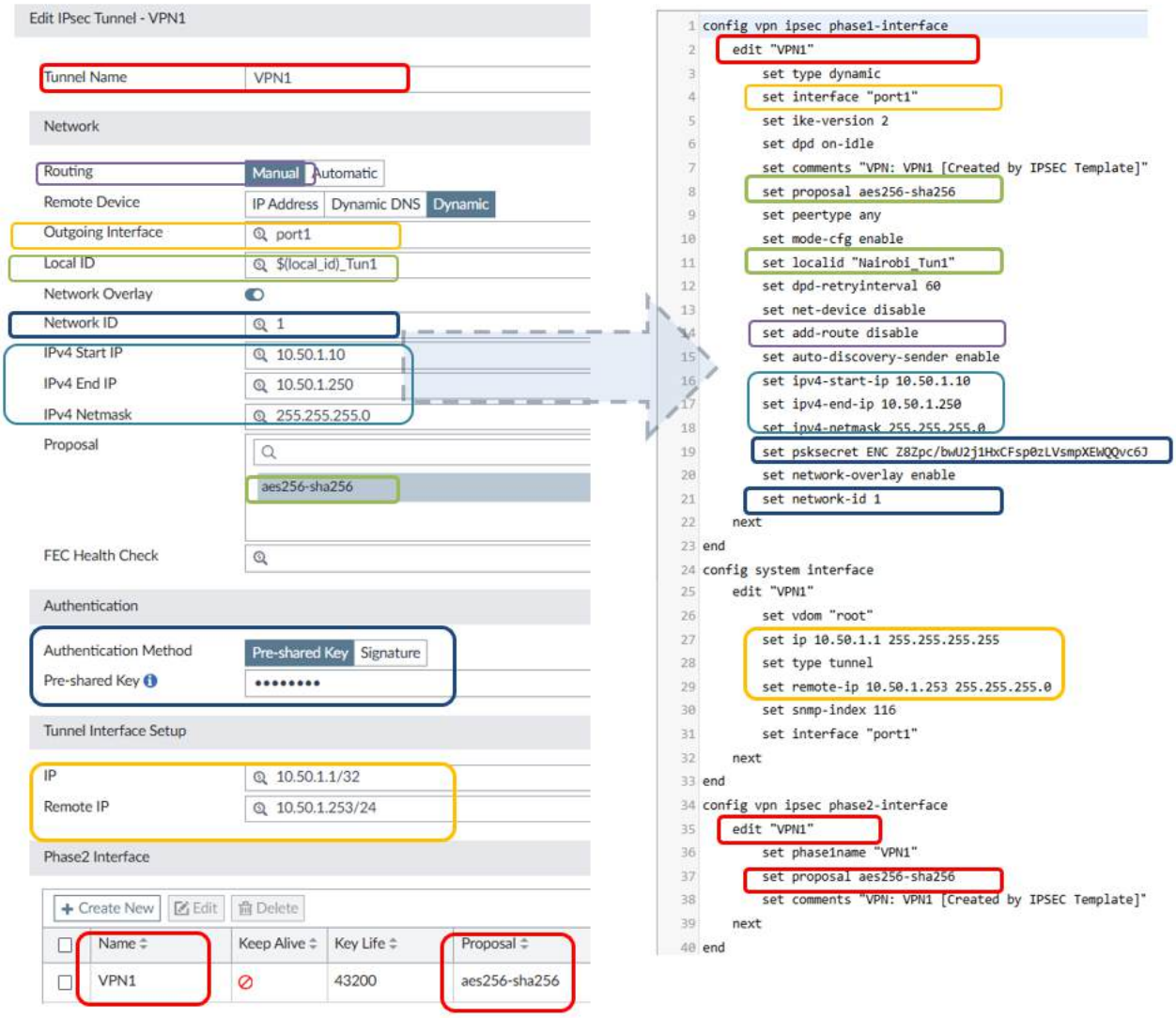


HUB1_IPSec templates are attached to **Template groups** which are assigned to the devices. The installation Wizard is used to install the configured IPSec template. These are also considered **device setting** (interface, routing etc.). The target device should be available (connection up). During the initial modelling, use the install preview to review the configuration and check compatibility with the target device to track any mismatches.



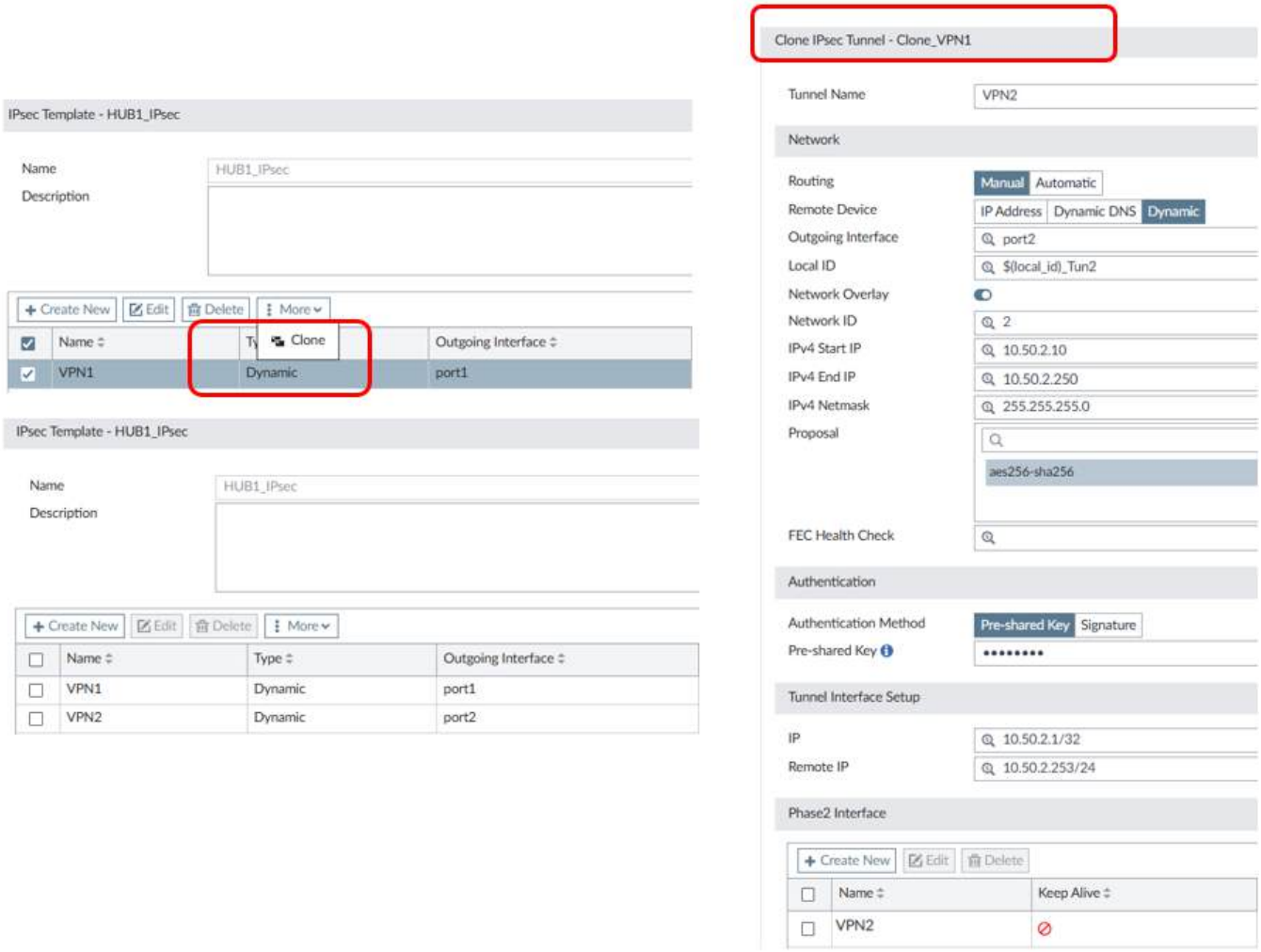
HUB1 IPSec VPN1 Config Preview

The template is pushed to HUB1 and creates the required interfaces and IPsec settings as shown.



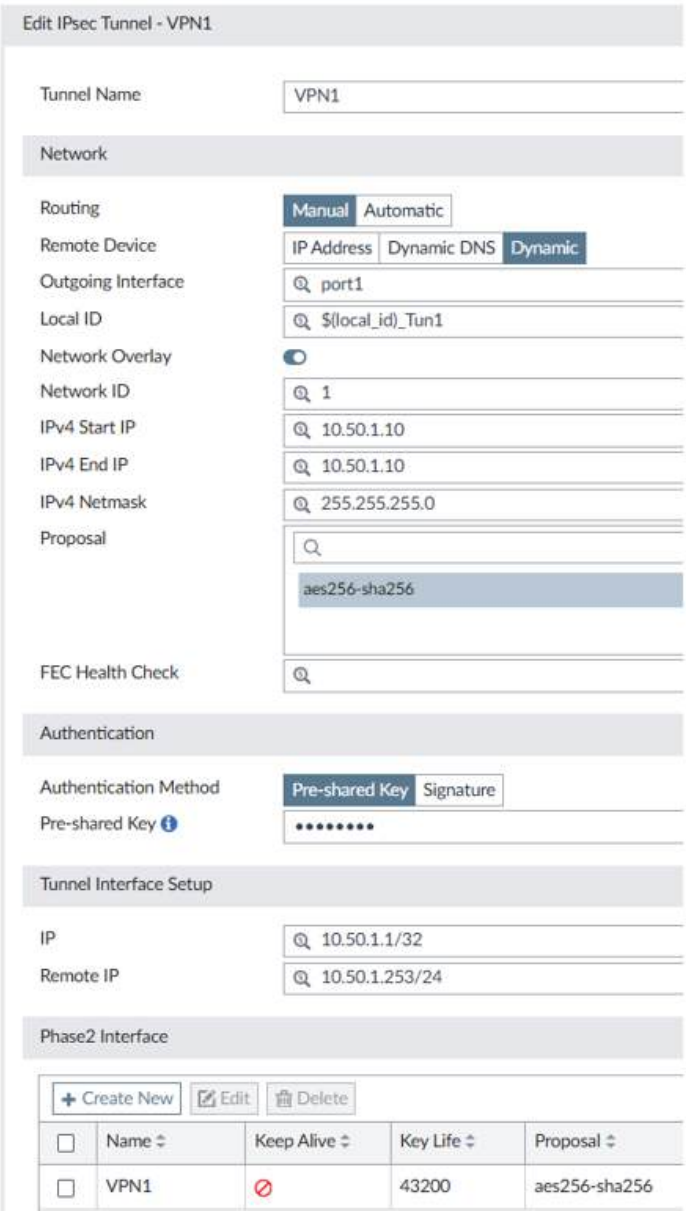
HUB1 IPSec VPN2

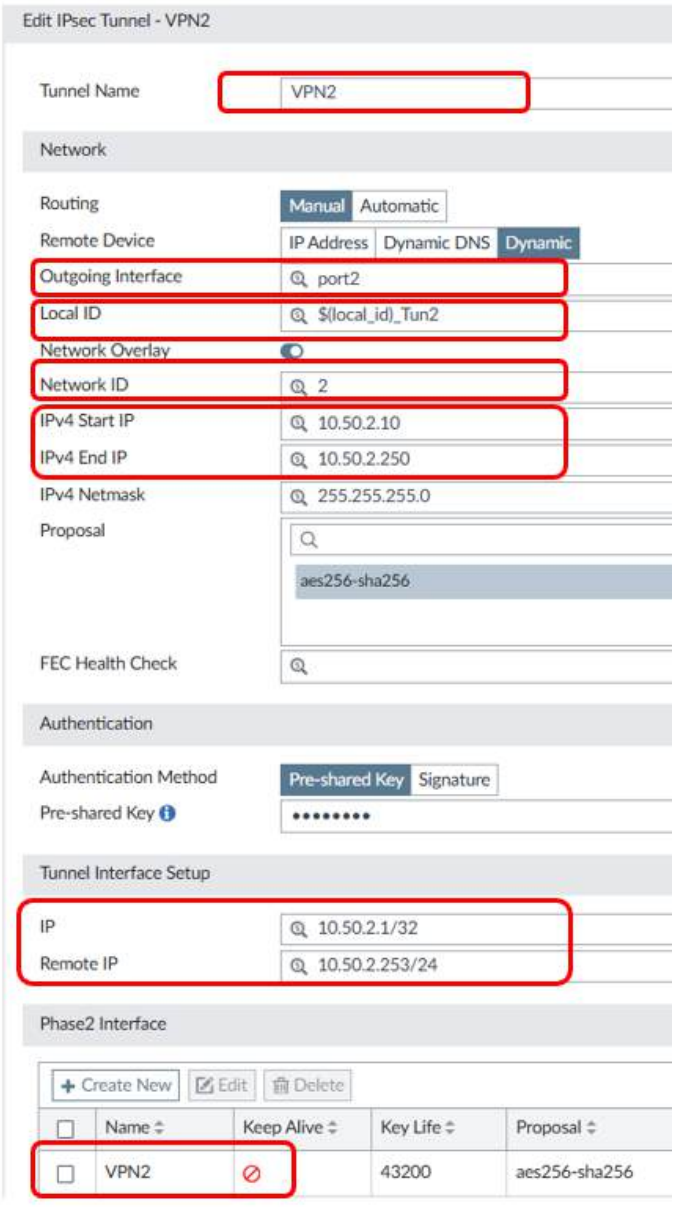
The second tunnel is configured by cloning VPN1 created from the IPsec Template. Once cloned, settings are customized to match the second tunnel.



HUB IPSec VPN1 & VPN2 Templates Comparison

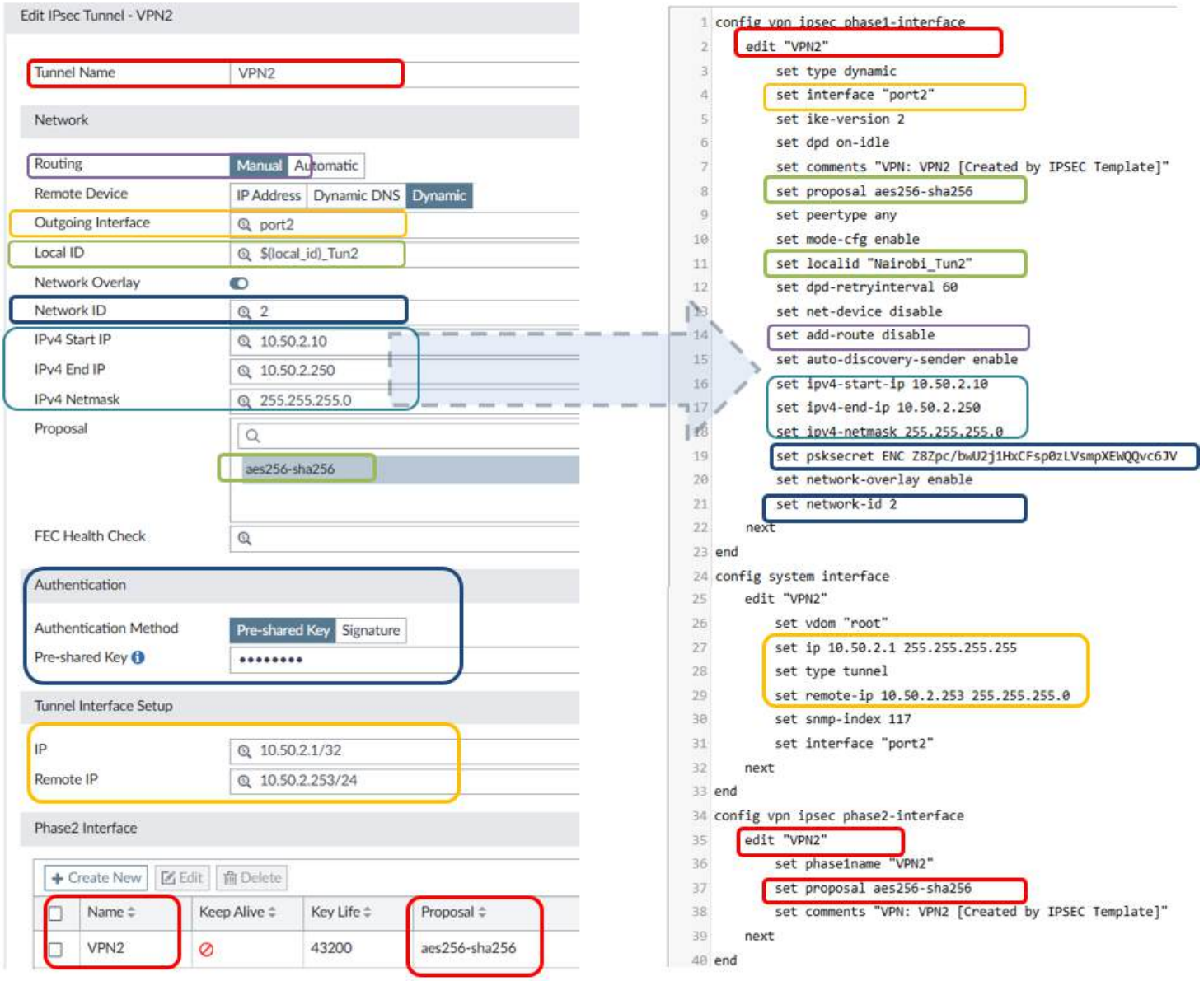
The tunnel name, outgoing interface, local ID, network ID, Start and End Ips, Tunnel Interface Setup and phase2 name are customized to match the second tunnel. All other settings remain the same. Any additional VPNx tunnels (3,4..) are made in the same way by creating a clone of the existing tunnels.





HUB1 IPSec VPN2 Config Preview

The template is pushed to HUB1 and creates the required interfaces and IPsec settings as shown for VPN2.

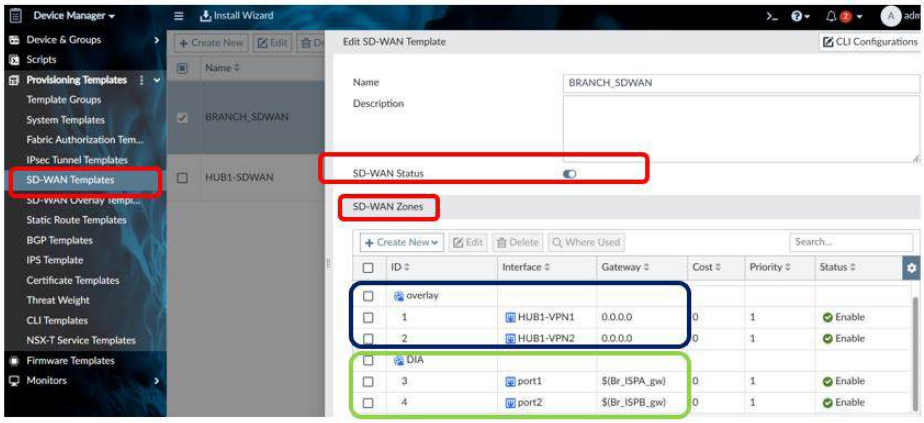


Branch SD-WAN Template

While SD-WAN Template configurations are done at any stage on FortiManager, they can only be pushed to devices after IPsec configuration is done and the referenced interfaces are configured. This is critical during initial modelling but can be sent at once when all configurations are done for additional branches.

The SD-WAN template requires the interface names to be typed manually and must match the interfaces created by the IPsec Template. Great caution is required to avoid mismatches. For the underlay network, metadata variables are used to define the gateways to be used for underlay default routing.

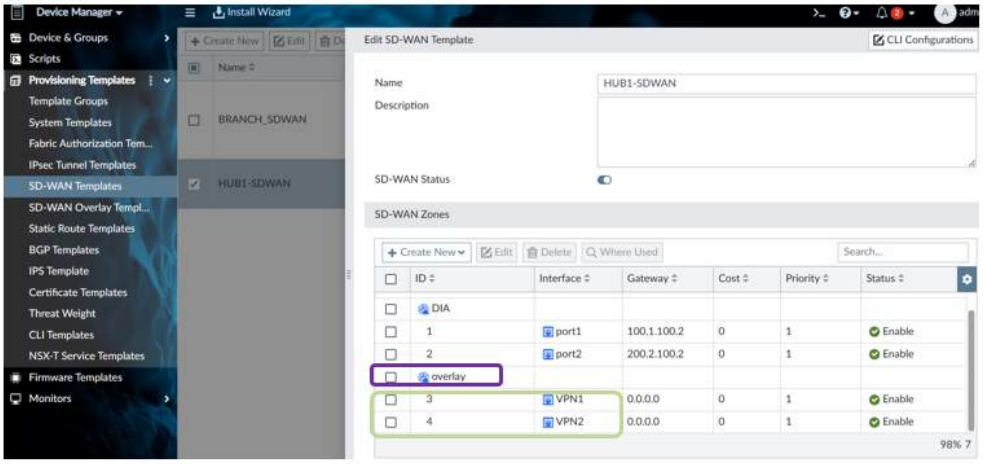
When adding the underlay network as SD-WAN interfaces, you must add the gateway ip addresses when defining them as sdwan member interfaces. Without gateways explicitly defined, the remote devices will not have default routes, any static default routes already configured will not be present in the routing table although available in the running configuration. This results in the remote unit losing access to FortiManager and remote access (GUI or CLI) won't be possible because of missing default routes.



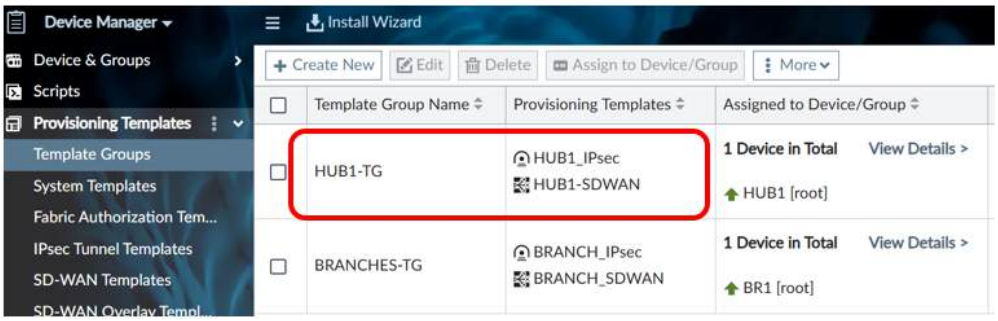

```
config system sdwan
    config zone
        edit "overlay"
        next
        edit "DIA"
        next
    end
    config members
        edit 1
            set interface "HUB1-VPN1"
            set zone "overlay"
        next
        edit 2
            set interface "HUB1-VPN2"
            set zone "overlay"
        next
        edit 3
            set interface "port1"
            set zone "DIA"
            set gateway 100.1.1.2
        next
        edit 4
            set interface "port2"
            set zone "DIA"
            set gateway 200.2.1.2
        next
    end
end
```

HUB SDWAN Template

HUB1_SDWAN template is attached to **HUB1_TG** template group and applied. This provisions the overlay zone since the DIA zone was provisioned when setting up FMG access over the internet. The zone and member interfaces are then used where required since they are now grouped as members. VPN1 & VPN2 are typed manually.

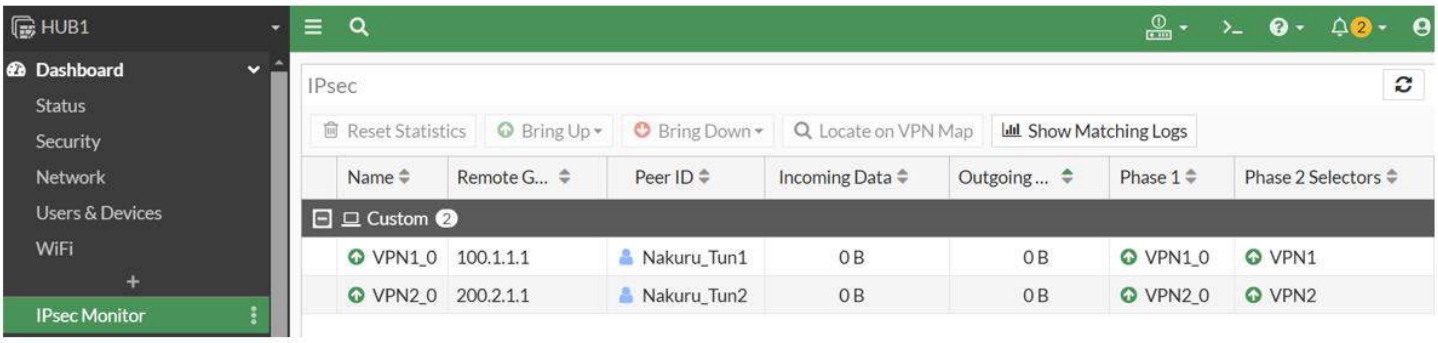


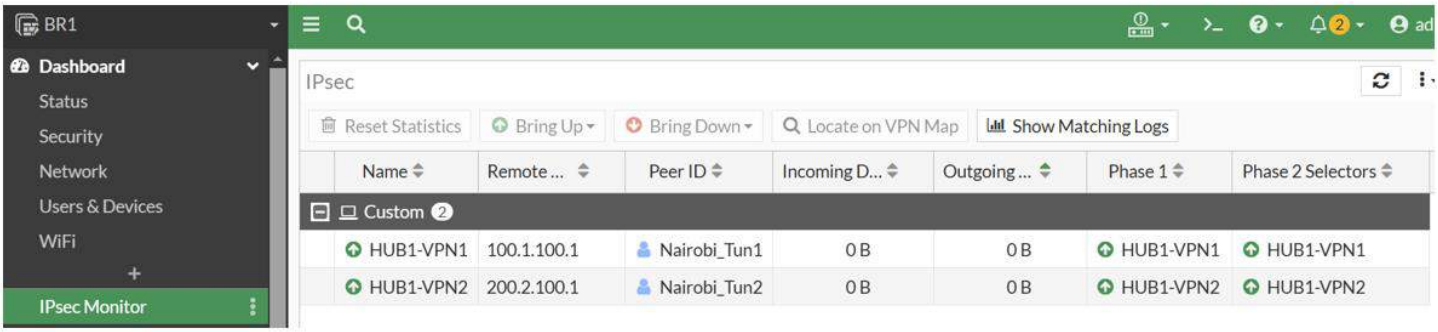

```
config system sdwan
    config zone
        edit "overlay"
        next
    end
    config members
        edit 3
            set interface "VPN1"
            set zone "overlay"
        next
        edit 4
            set interface "VPN2"
            set zone "overlay"
        next
    end
end
```



Requirement for firewall policies at the HUB and Branches

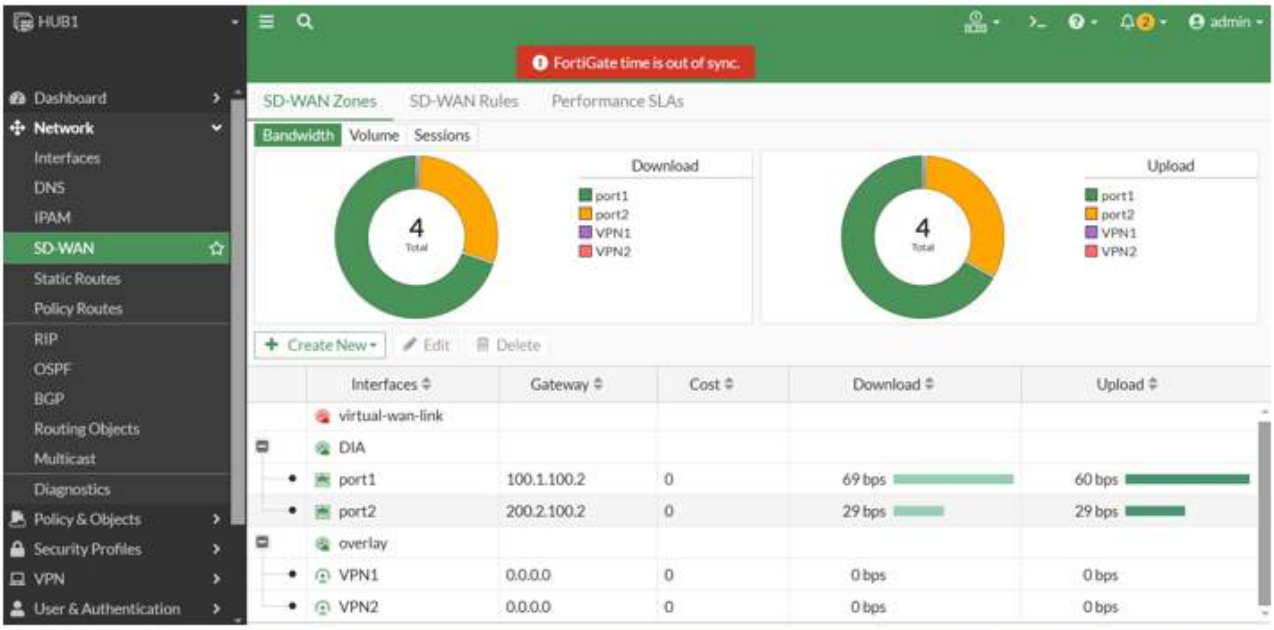
The tunnels come up without security policies configured when the tunnel interfaces at the HUB and branches are configured as SDWAN zones. The alternative is to configure firewall policies inbound at the hub and outbound at the branches referencing overlay interfaces. Since the SDWAN zones are already configured, the tunnels will be up as shown below.

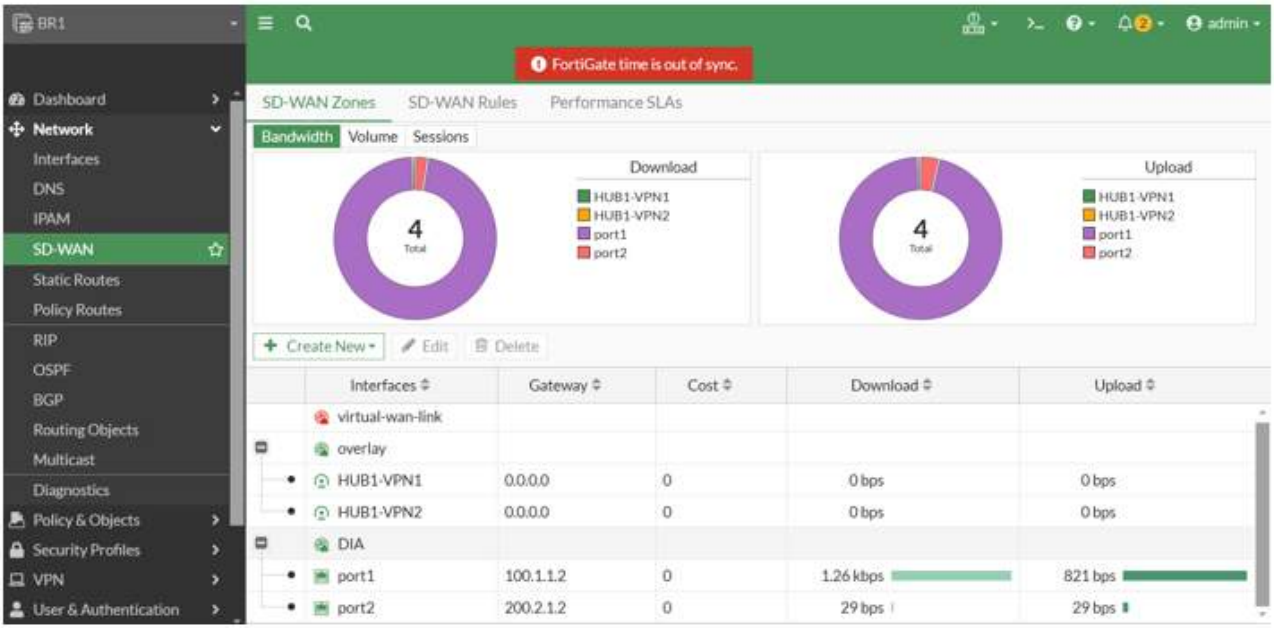




SDWAN FortiGate GUI at HUB1 and Branch1 Verification

At this point, HUB1 and BR1 have the underlay and overlay interfaces grouped in SDWAN zones. Since HUB1 and BR1 overlay interfaces are now grouped in zones, the tunnels will come up without configuring firewall policies to allow IPsec traffic. If Zones are not configured, it is mandatory to configure the firewall policies otherwise, the tunnels remain down/disabled.

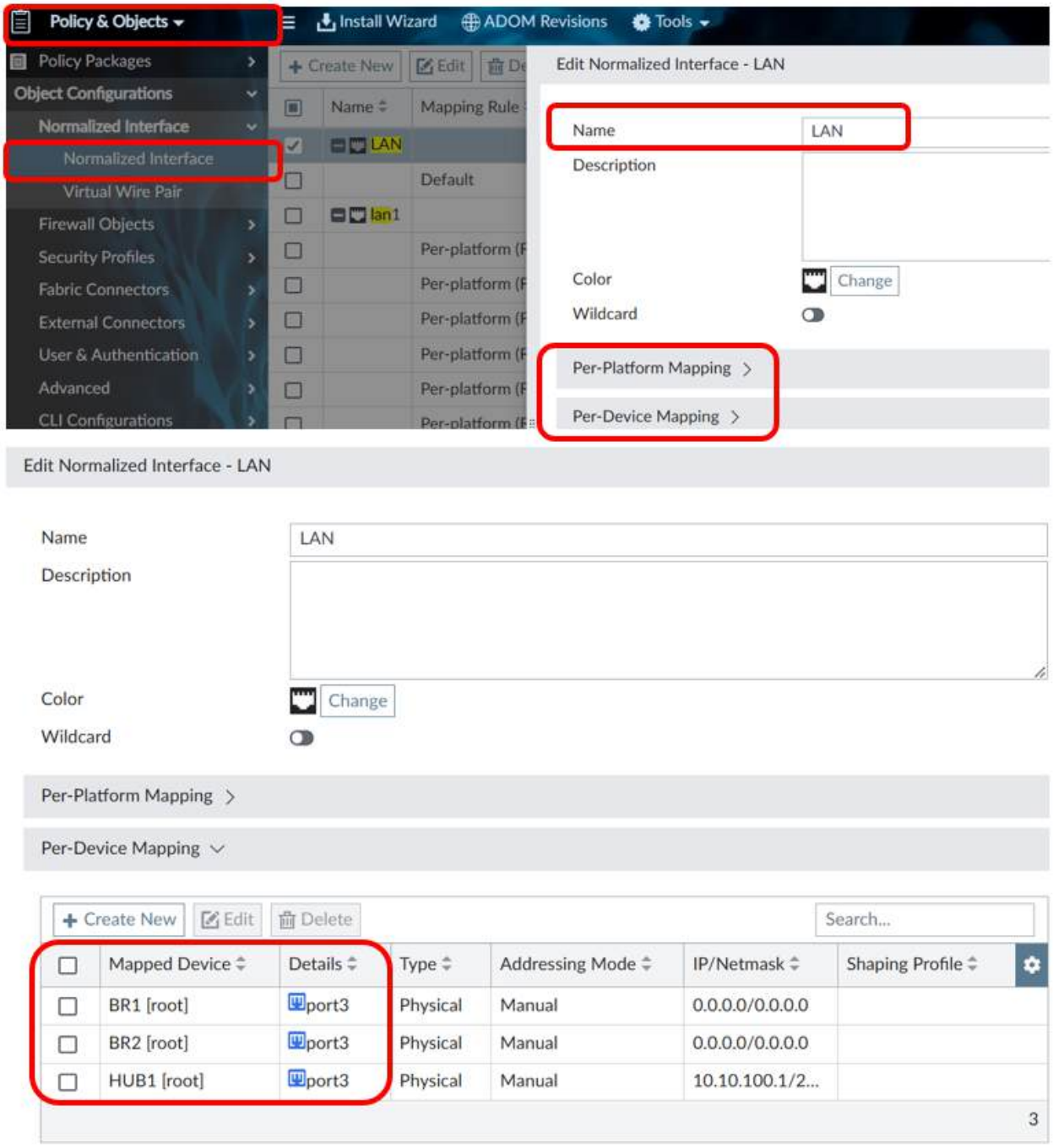




Normalized Interface

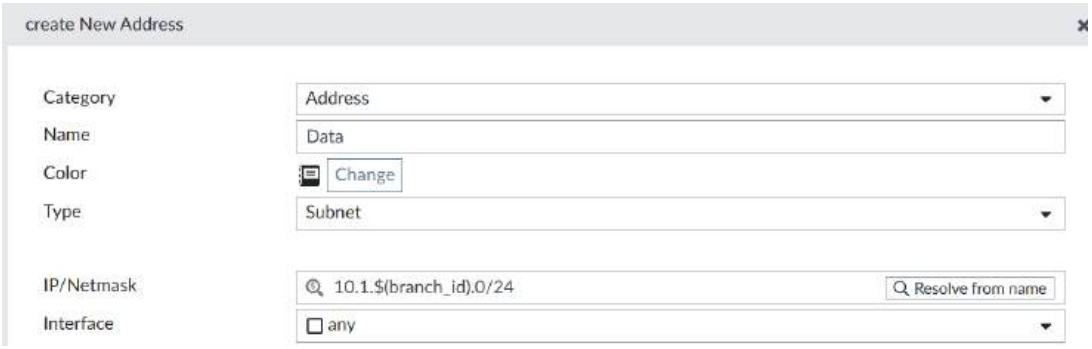
This is a logical interface used to map/refence different physical interfaces. This makes designs where the LAN/WAN/DMZ etc. have different physical interfaces per site share policies and other settings since the normalized interface is the one referenced. It defines mapping rules per-device or per platform.

Typical in most SDWAN deployments, the LAN and ISP links (underlays) are represented by normalized interfaces. For this demo, only the LAN interface is abstracted using a normalized interface. The configurations are done in the policy & objects section. Many default mappings exist for different models.

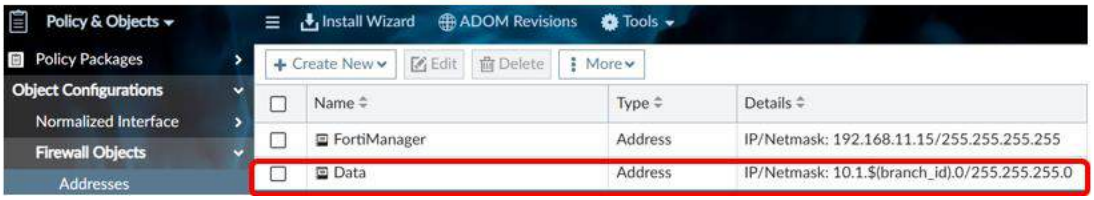


Address Objects

Address Objects support per device mapping by leveraging the branch_id to offer great flexibility for well-designed networks. For this setup, the branch_id is used to reference the 3rd octet when defining data networks/subnets.

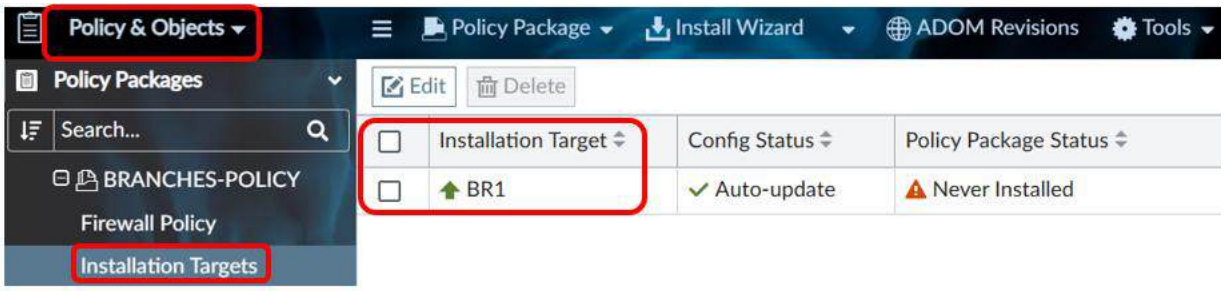


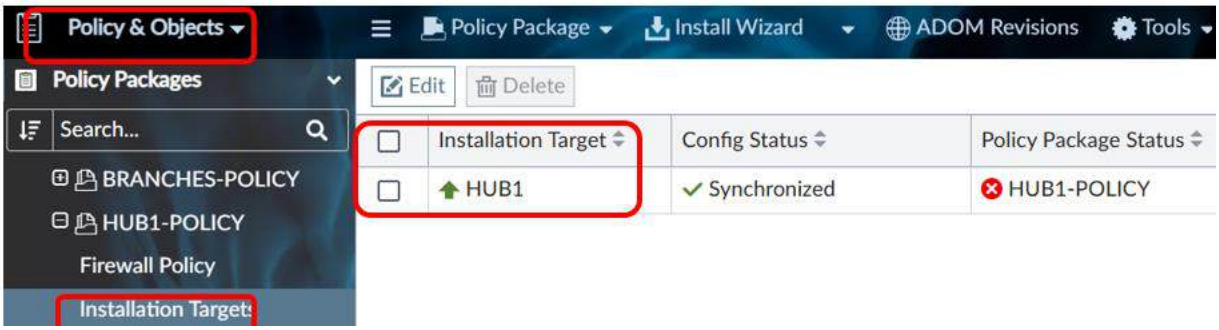
To use the metadata variable in address objects you **must** define a value in **Default Value** field when defining the Metadata Variable to avoid getting the error below:



Firewall Policies

Firewall policy packages are configured under policies and objects, one policy is required for all branches and one for the hub. Once defined, they are attached to the appropriate devices under installation targets, the branch policy is mapped to the branches and HUB1 policy to HUB1. Once modelling of the branches is complete, attach the group instead of attaching individual branches.

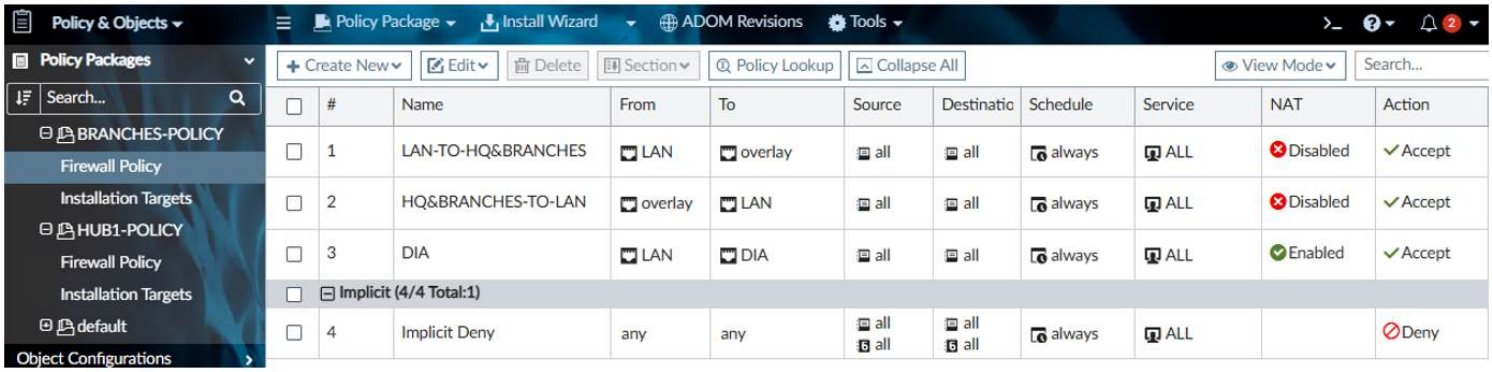




Branch Firewall Policies

Three key policies are required at each branch.

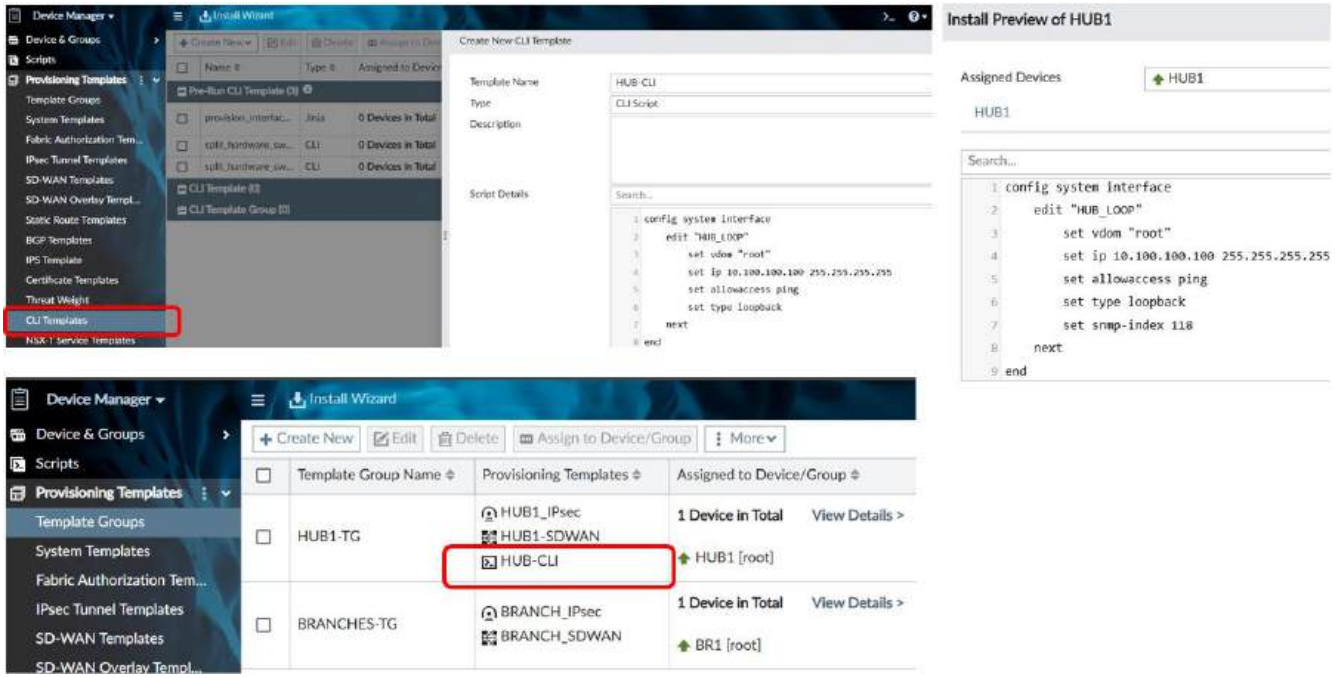
- Branch LAN to HQ and other remote sites outbound
- Inbound rule to allow HQ and Branches to access the LAN
- Direct internet access (DIA) using the underlay network. This will have NAT enabled



Hub Loopback Setup

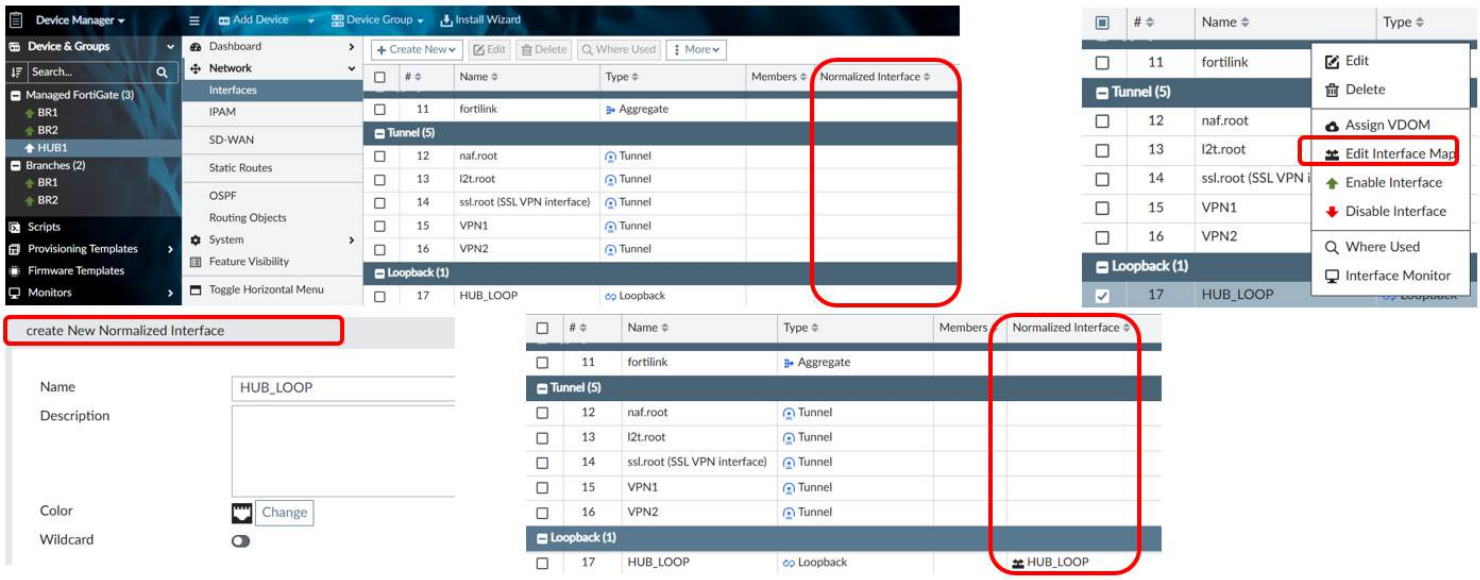
To probe the link performance, branches probe a loopback at the HUB. A loopback is preferred for stability and guaranteed uptime compared to a device at the HUB. It is also directly attached to the firewall and not subject to any degraded performance at the HUB LAN network.

A Loopback interface is configured at the HUB with **10.100.100.100/32** A firewall policy is required at the HUB to allow traffic to the loopback on overlay interfaces. The loopback is configured using the CLI template but could be configured using other ways. Once configured, the loopback interface will not be available on FortiManager until it is normalized using interface Mapping



HUB LOOPBACK Interface normalization

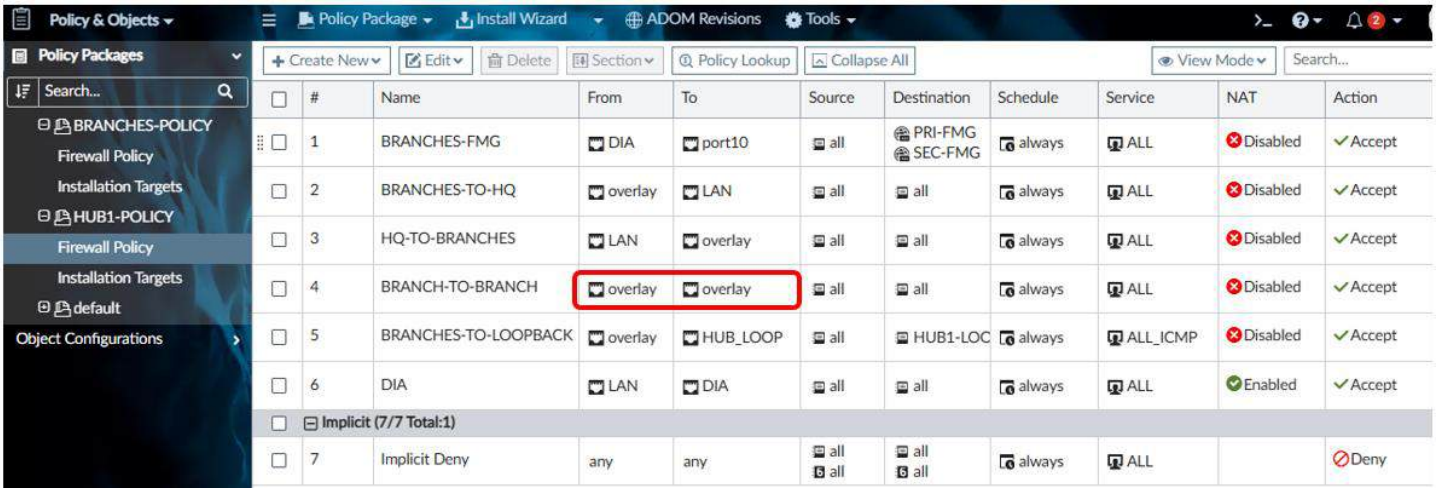
The loopback interfaces will not be available to configure firewall policies until interface normalization is done using interface mapping. Edit the loopback interface, select Edit interface Map, create the new Normalized interface and map it to the loopback interface. You can use the same name used when defining the interface.



HUB Firewall Policies

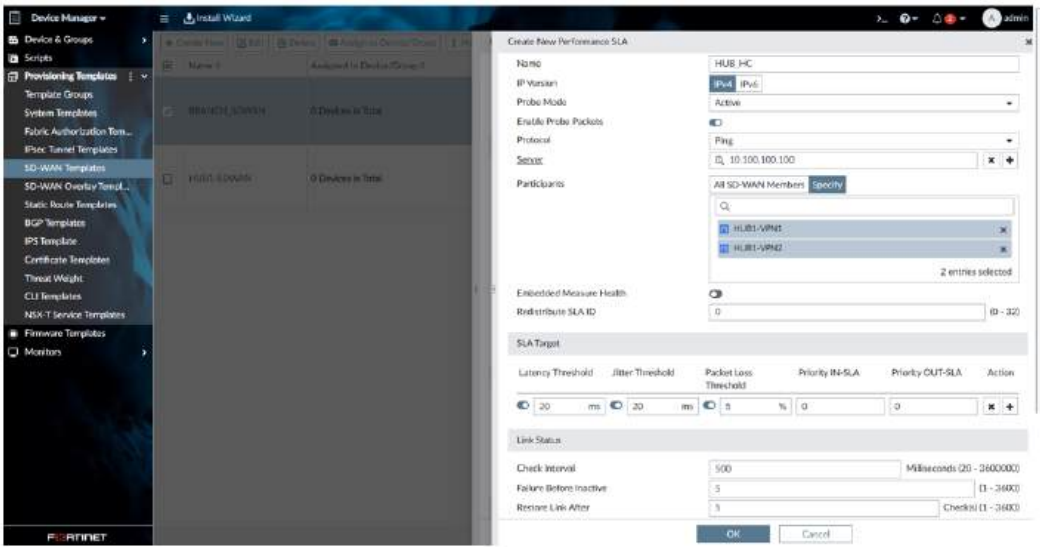
Six key policies are required at HUB1.

- FortiManager access from the Branches
- Branches to HQ LAN inbound on the overlay network
- Outbound rule to allow HQ to access branches on the overlay network
- Branch to Branch traffic hair pinned at the HUB **i.e. source and destination interfaces are the overlay network.**
- Branches to LOOPBACK for Healthcheck on the overlay network
- Direct internet access (DIA) using the underlay network. This will have NAT enabled



HUB LOOPBACK Health Check

A Healthcheck is configured using the SDWAN template at the branches. This probes the loopback at the HUB using the overlay networks according to defined SLA metrics.

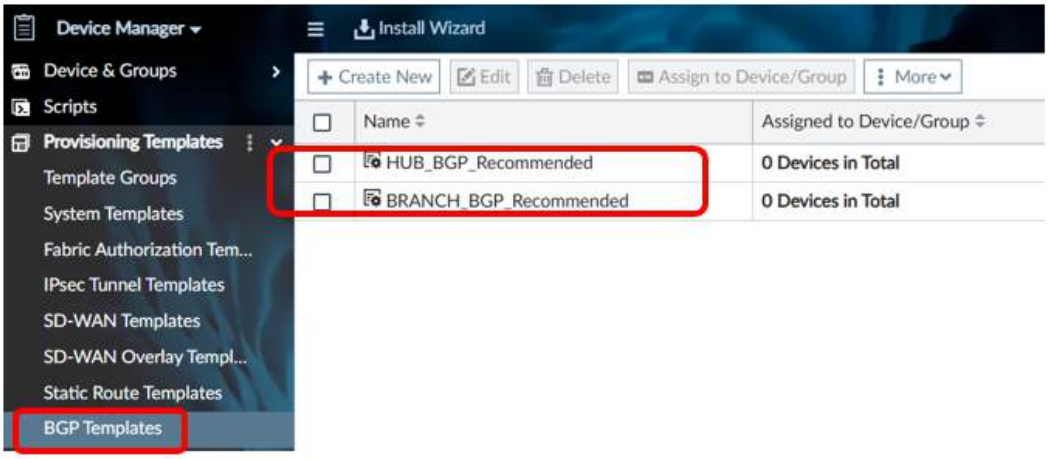


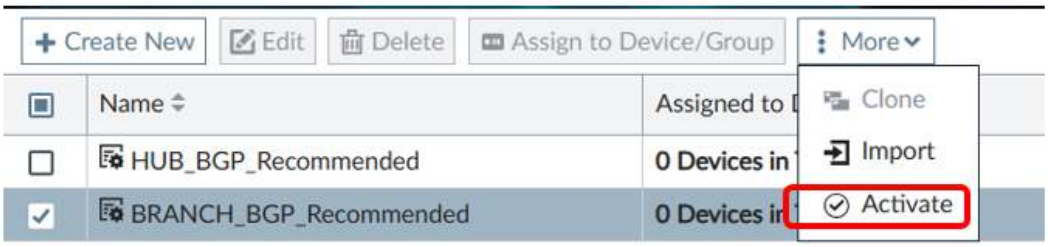

```
config system sdwan
    config health-check
        edit "HUB_HC"
            set server 10.100.100.100
            set members 2 1
            config sla
                edit 1
                    set latency-threshold 20
                    set jitter-threshold 20
                    set packetloss-threshold 5
                next
            end
        next
    end
end
```

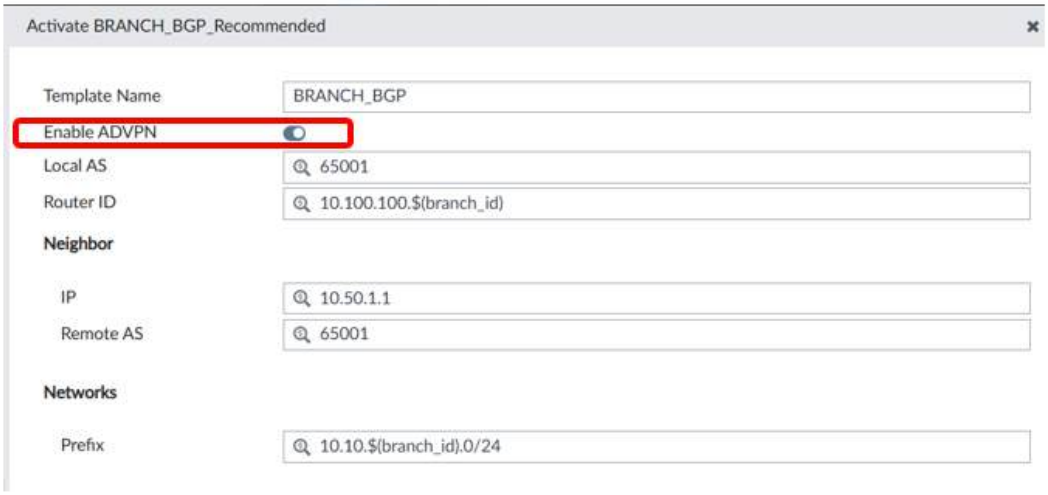
BGP Branch Configuration

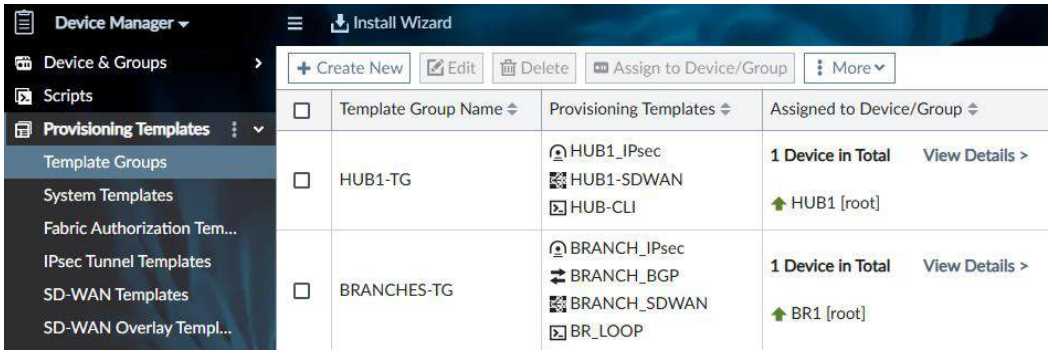
Two templates are predefined for BGP i.e. Hub and Branch Recommended that are activated to configure custom templates. The remote neighbor referenced is the tunnel static ip. For each tunnel configured, a new neighbor is defined.

BGP is also used to define the networks to be advertised to the peers. iBGP is configured between the HUB and branches. At the point of activation, ADVPN is configured for BGP. The template is attached to the Branches template Group.

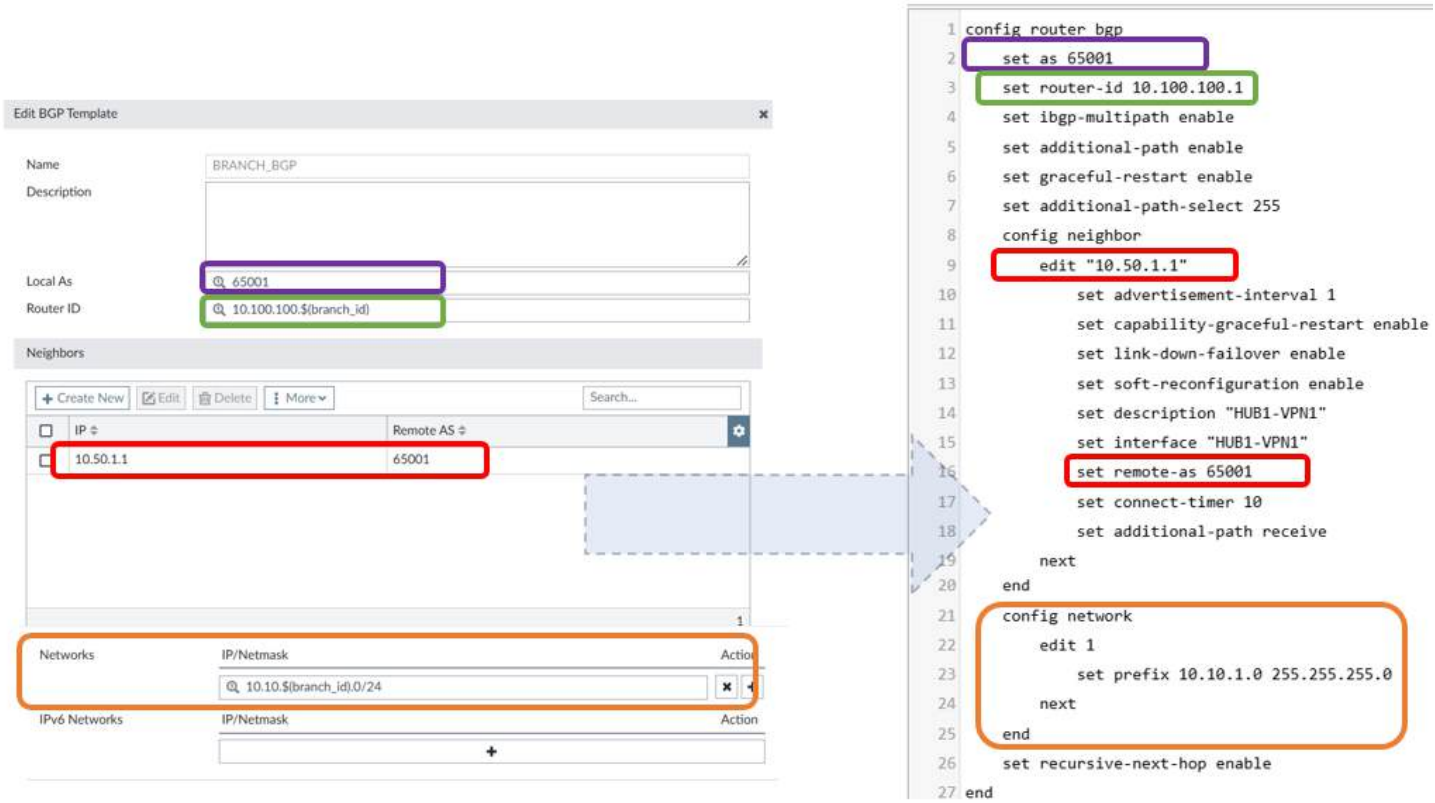








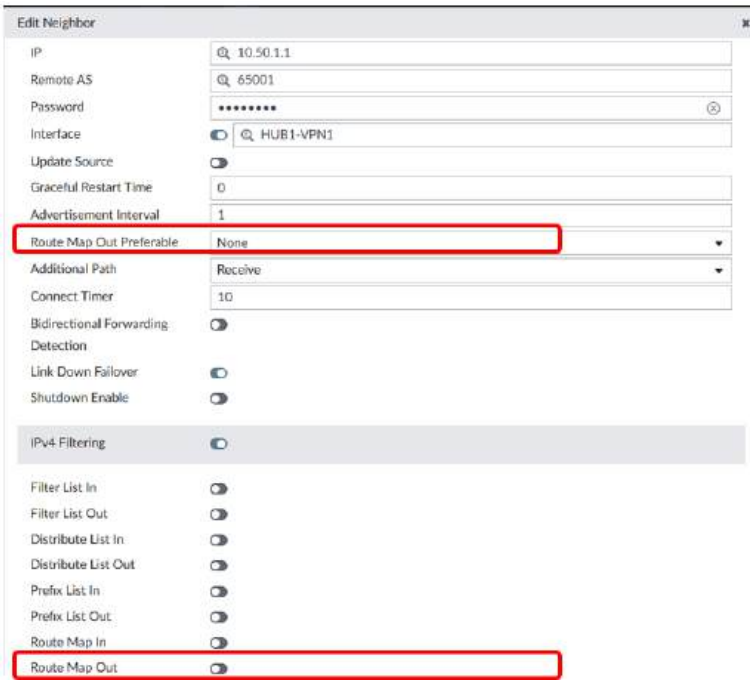
Branch1 BGP Neighbor1 Configuration Preview

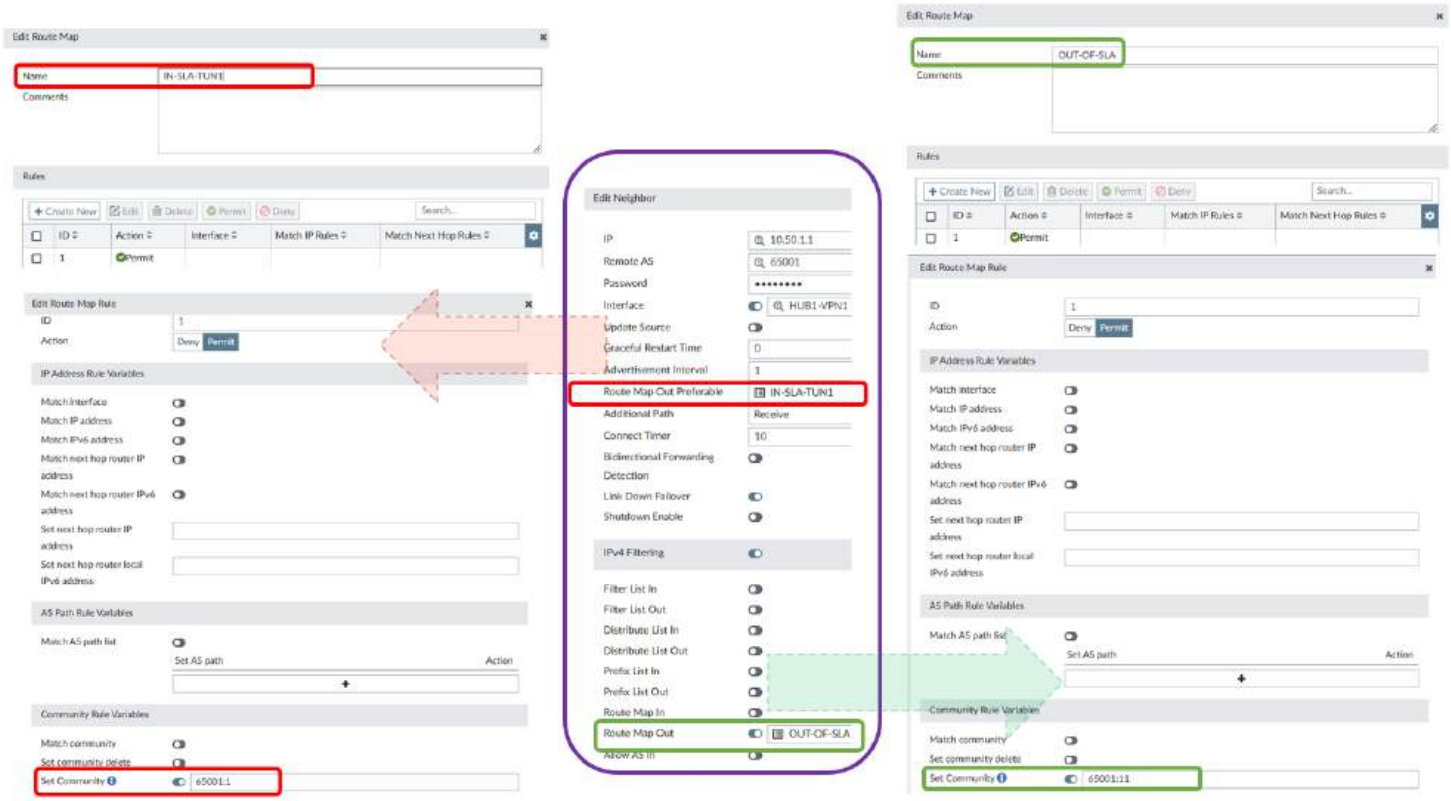


Branch1 BGP Route-Map Configuration

Two route-maps are configured. For the IN-SLA state, a route-map will be configured for each tunnel while one route-map is configured for the OUT-OF-SLA state.

When the link is within the SLA as defined by reachability to the loopback at HUB1, the **Route Map Out Preferable** is applied, when out of SLA, **Route Map Out** is applied.



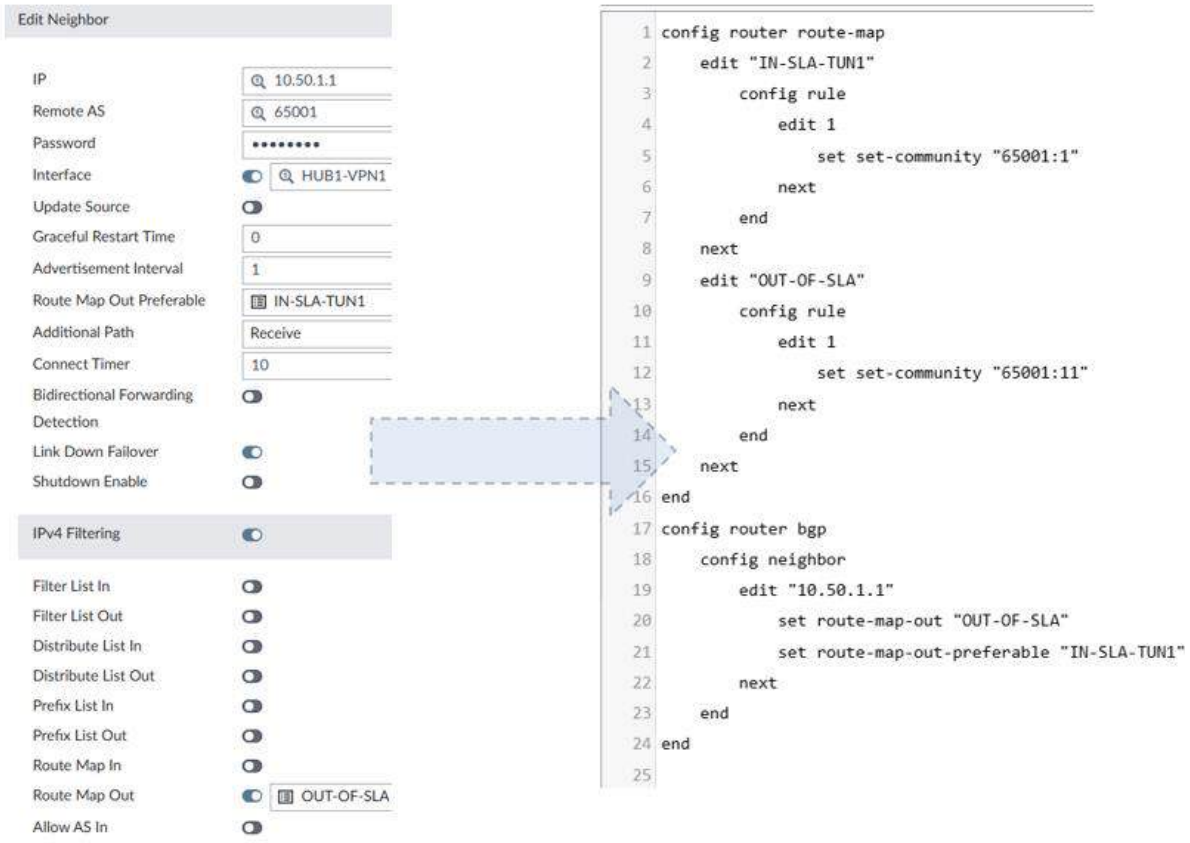


The route-map IN-SLA-TUN1 sets a community of 65001:1 for advertised BGP routes when the links are within SLA. The route-map OUT-OF-SLA sets a community of 65001:11 for advertised BGP routes when the links are out of defined SLA thresholds.

Although it is possible to use a single SLA when all available VPN tunnels are within SLA (IN-SLA), using different route-maps/communities makes visibility and troubleshooting at the HUB easier since we can easily map communities to Tags and routes.

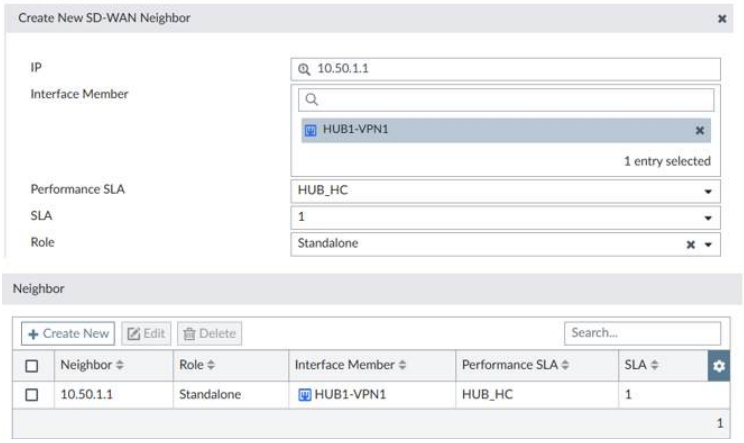
A single out of SLA route map caters for all available overlays when out of compliance.

Branch1 BGP Neighbor1 Configuration Preview



BRANCH1 SDWAN-BGP Configuration

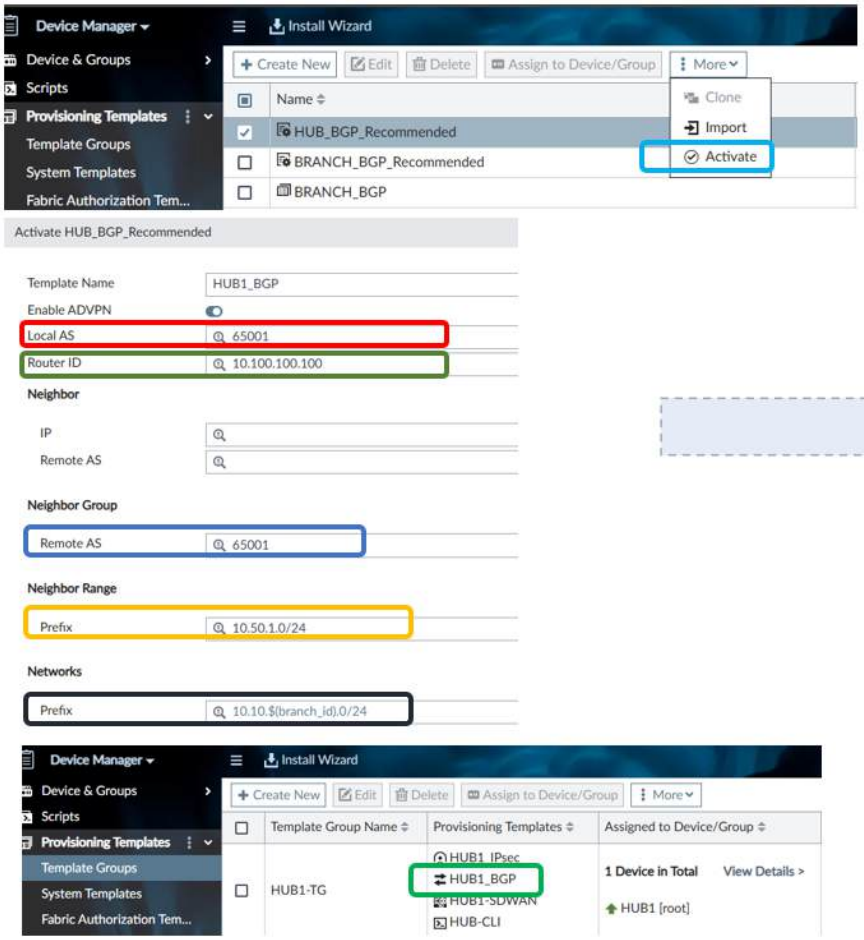
The defined performance SLA to track reachability of the HUB is used within the SDWAN template to associate link quality to specific BGP neighbors. All available neighbors are defined in the SDWAN template and associated with the same Loopback Health Check. This is an important step without which all available routes are marked as out of SLA from a iBGP perspective at the HUB.




```
config system sdwan
    config neighbor
        edit "10.50.1.1"
            set health-check "HUB_HC"
            set sla-id 1
            set member 1
        next
    end
end
```

HUB1 VPN1 BGP Configuration

Neighbor configurations are defined using neighbor-groups and neighbor-ranges to support multiple branches without configuration changes. The HUB BGP template is configured and attached to the HUB Template Group.

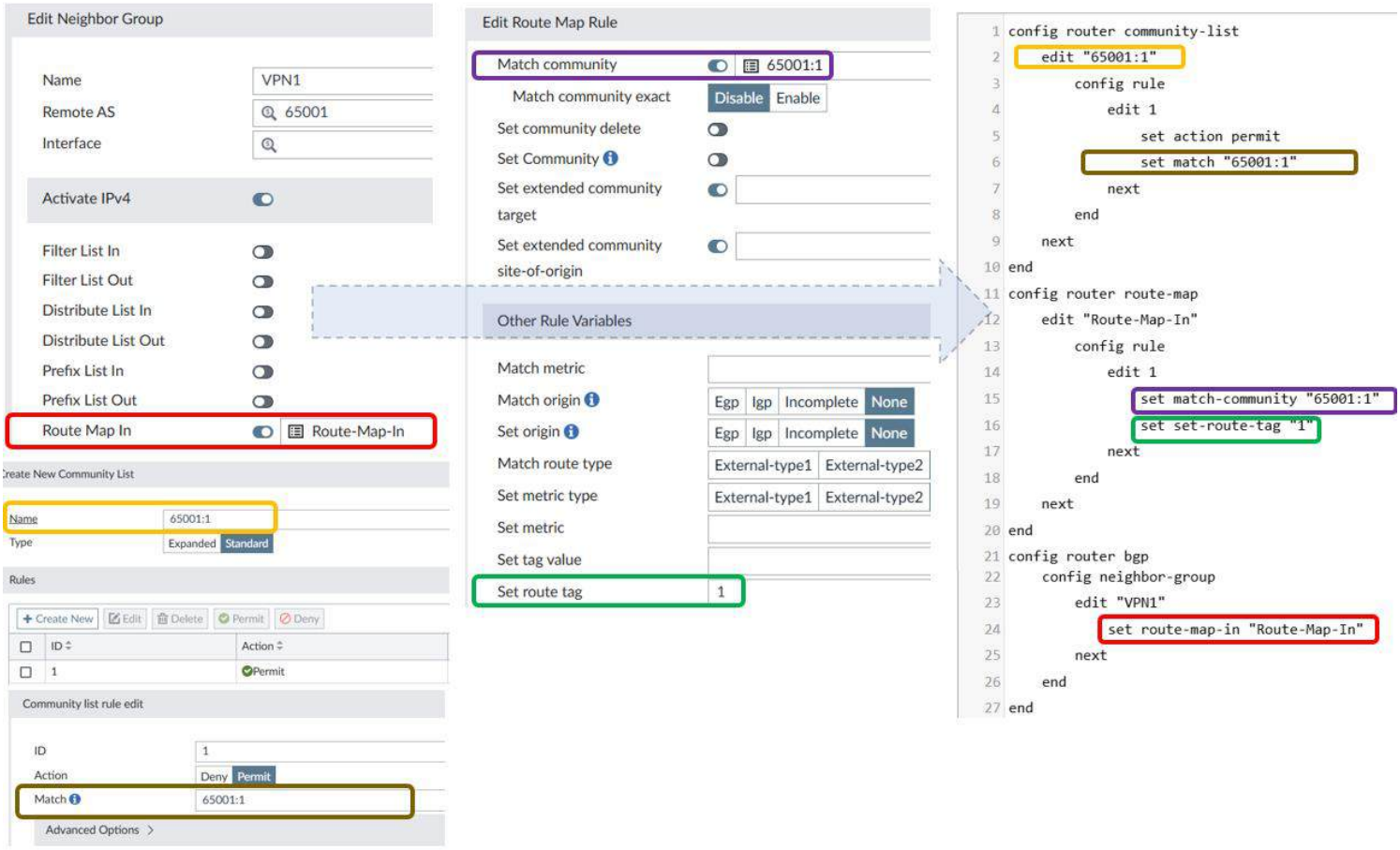



```
config router bgp
    set as 65001
    set router-id 10.100.100.100
    set ibgp-multipath enable
    set additional-path enable
    set graceful-restart enable
    set additional-path-select 255
    config neighbor-group
        edit "VPN1"
            set capability-graceful-restart enable
            set link-down-failover enable
            set next-hop-self enable
            set soft-reconfiguration enable
            set remote-as 65001
            set additional-path send
            set route-reflector-client enable
        next
    end
    config neighbor-range
        edit 1
            set prefix 10.50.1.0 255.255.255.0
            set neighbor-group "VPN1"
        next
    end
    config network
        edit 1
            set prefix 10.10.100.0 255.255.255.0
        next
    end
end
```

HUB1 BGP Route-Map-In Neighbor1

The route-map and community-lists are defined within BGP Neighbor Groups. The route-map is applied inbound to match the communities set at the branches. It essentially matches the communities defined at the branch and converts them to route tags to be used in sdwan rules.

Use one route-map inbound with multiple rules to match different tunnels and communities defined at the branches.



HUB1 BGP Verification

The branches are seen as a route reflector. The IP and router-id of the branch are shared for each subnet shared. Route-Tags/communities are attached for each network when within SLA and when out of conformity.

```
HUB1 # get router info bgp network
VRF 0 BGP table version is 2, local router ID is 10.100.100.100
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.10.1.0/24     10.50.1.10      0              100      0        1 i <-/1>
*> 10.10.100.0/24   0.0.0.0                        100  32768        0 i <-/1>

Total number of prefixes 2
```

```
HUB1 # get router info bgp network 10.10.1.0/24
VRF 0 BGP routing table entry for 10.10.1.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    10.50.1.10 from 10.50.1.10 (10.100.100.1)
      Origin IGP metric 0, route tag 1, localpref 100, valid, internal, best
      Community: 65001:1
      Advertised Path ID: 1
       Last update: Tue Oct 29 23:00:07 2024
```

```
HUB1 # get router  info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 10.50.1.10 (recursive is directly connected, VPN1), 00:04:16, [1/0]
```

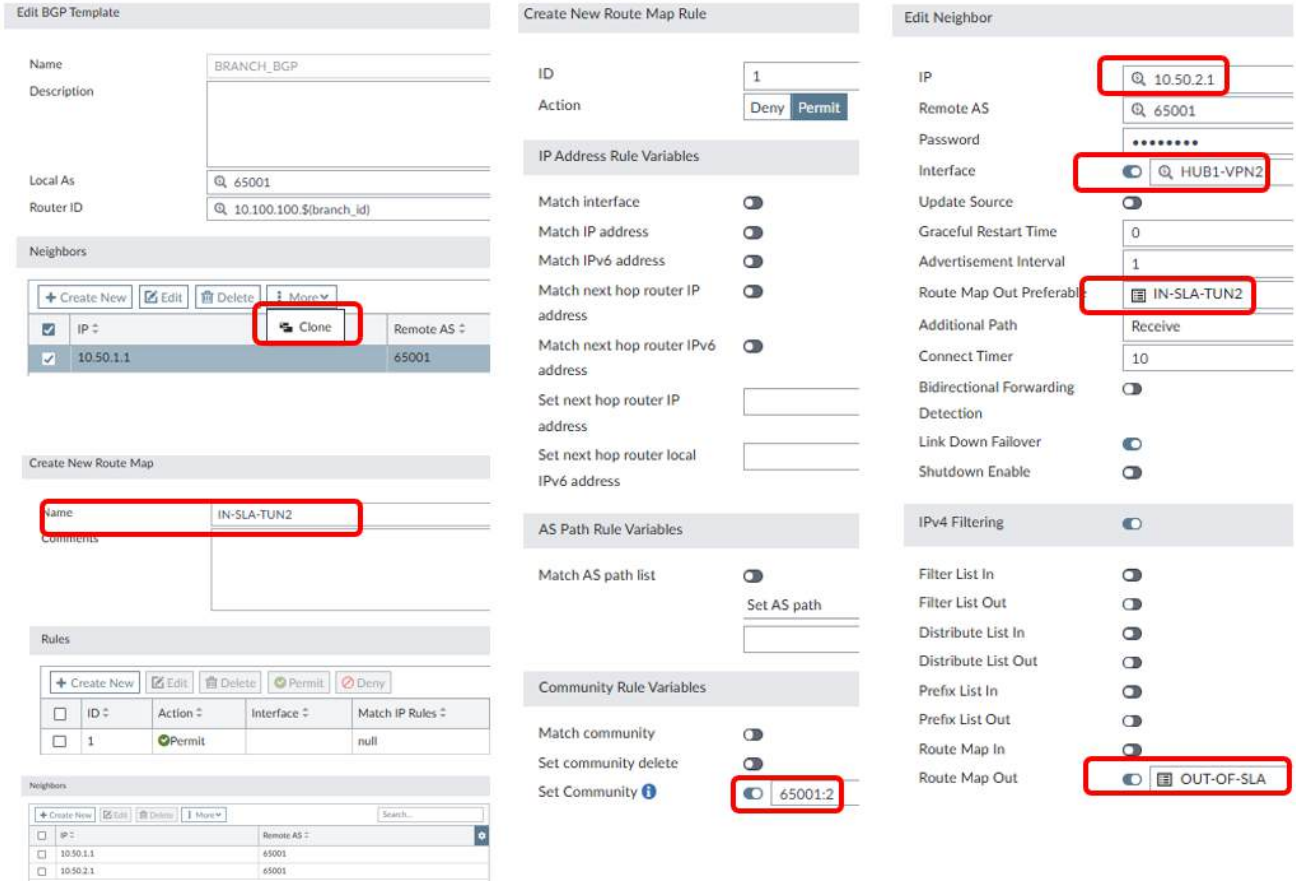
Branch1 BGP Verification

Neighborships are formed with the static ip configured on the overlay tunnel at the HUB. There are no route-tags since these are advertised upstream and not from the HUB down.

```
BR1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.100.0/24 [200/0] via 10.50.1.1 (recursive via HUB1-VPN1 tunnel 100.1.100.1), 06:02:47, [1/0]


BR1 # get router info bgp network 10.10.100.0/24
VRF 0 BGP routing table entry for 10.10.100.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local
    10.50.1.1 from 10.50.1.1 (10.100.100.100)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Receive Path ID: 1
      Advertised Path ID: 1
       Last update: Tue Oct 29 23:00:29 2024



BR1 # get router info bgp network
VRF 0 BGP table version is 5, local router ID is 10.100.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.1.0/24     0.0.0.0                       100  32768        0 i <-/1>
*>i10.10.100.0/24   10.50.1.1        0            100      0        0 i <1/1>

Total number of prefixes 2


BR1 # get router info bgp summary

VRF 0 BGP router identifier 10.100.100.1, local AS number 65001
BGP table version is 5
1 BGP AS-PATH entries
1 BGP community entries

Neighbor  V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.50.1.1 4      65001     491     494        5    0    0 06:03:58        1

Total number of neighbors 1
```

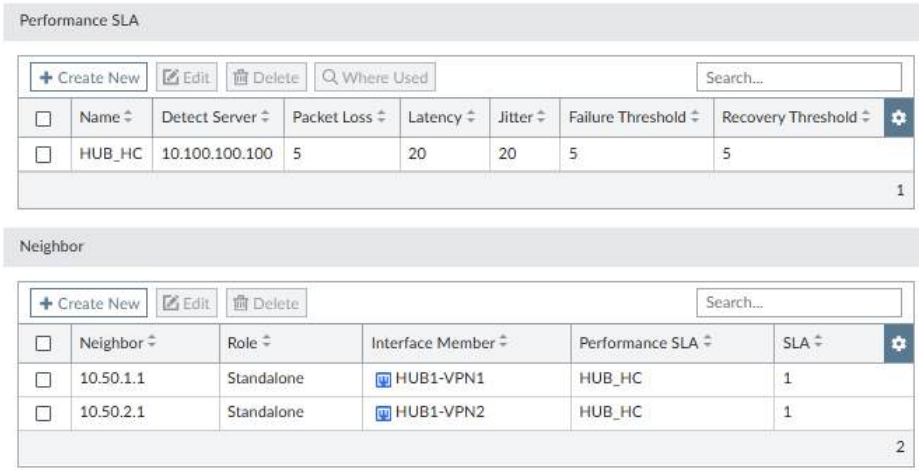
Branch1 BGP Neighbor2 Configuration.

For the second tunnel, the first neighbor is cloned and then define the neighbor as the underlay ip of ISPB at the HUB and change other parameters accordingly. As noted earlier, a second Route-map is created for the IN-SLA routes while the same route-map is attached when out of SLA.

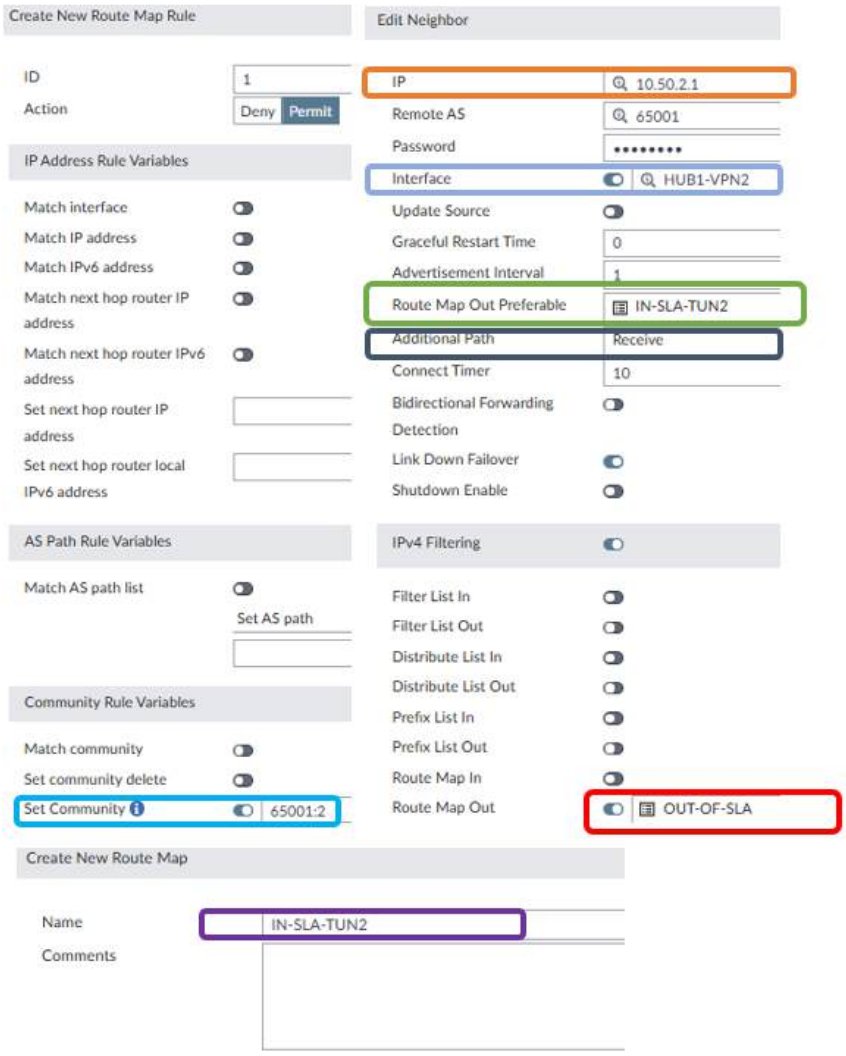


SDWAN Branch1 HUB1-VPN2

The SDWAN Template is modified by attaching the second bgp neighbor, second tunnel and Hub Loopback Healthcheck settings.



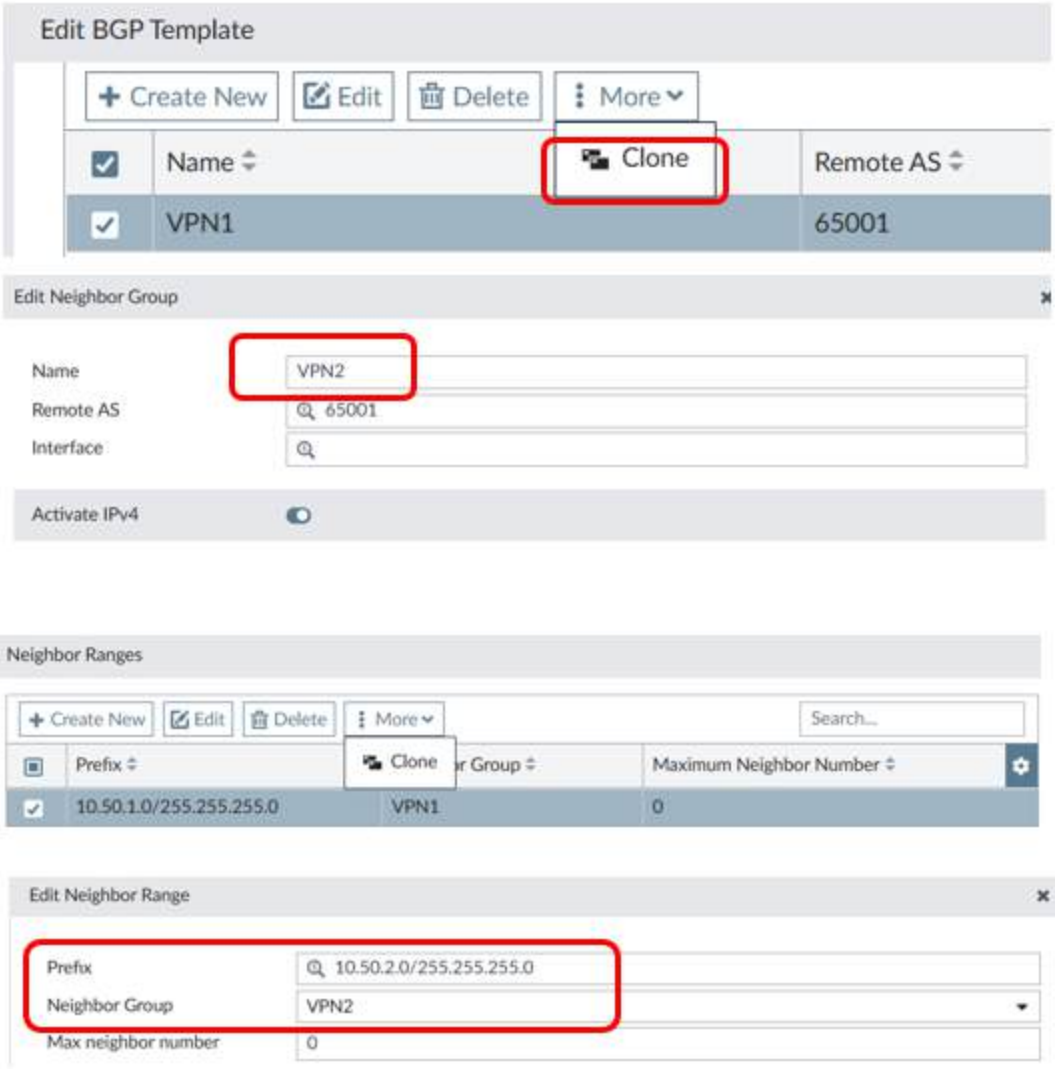
Branch1 BGP Neighbor2 Configuration Preview




```
config router route-map
    edit "IN-SLA-TUN2"
        config rule
            edit 1
                set set-community "65001:2"
            next
        end
    next
end
config router bgp
    config neighbor
        edit "10.50.2.1"
            set advertisement-interval 1
            set capability-graceful-restart enable
            set link-down-failover enable
            set soft-reconfiguration enable
            set description "HUB1-VPN1"
            set interface "HUB1-VPN2"
            set remote-as 65001
            set route-map-out "OUT-OF-SLA"
            set connect-timer 10
            set additional-path receive
            set route-map-out-preferable "IN-SLA-TUN2"
        next
    end
end
config system sdwan
    config neighbor
        edit "10.50.2.1"
            set health-check "HUB_HC"
            set sla-id 1
            set member 2
        next
    end
end
```

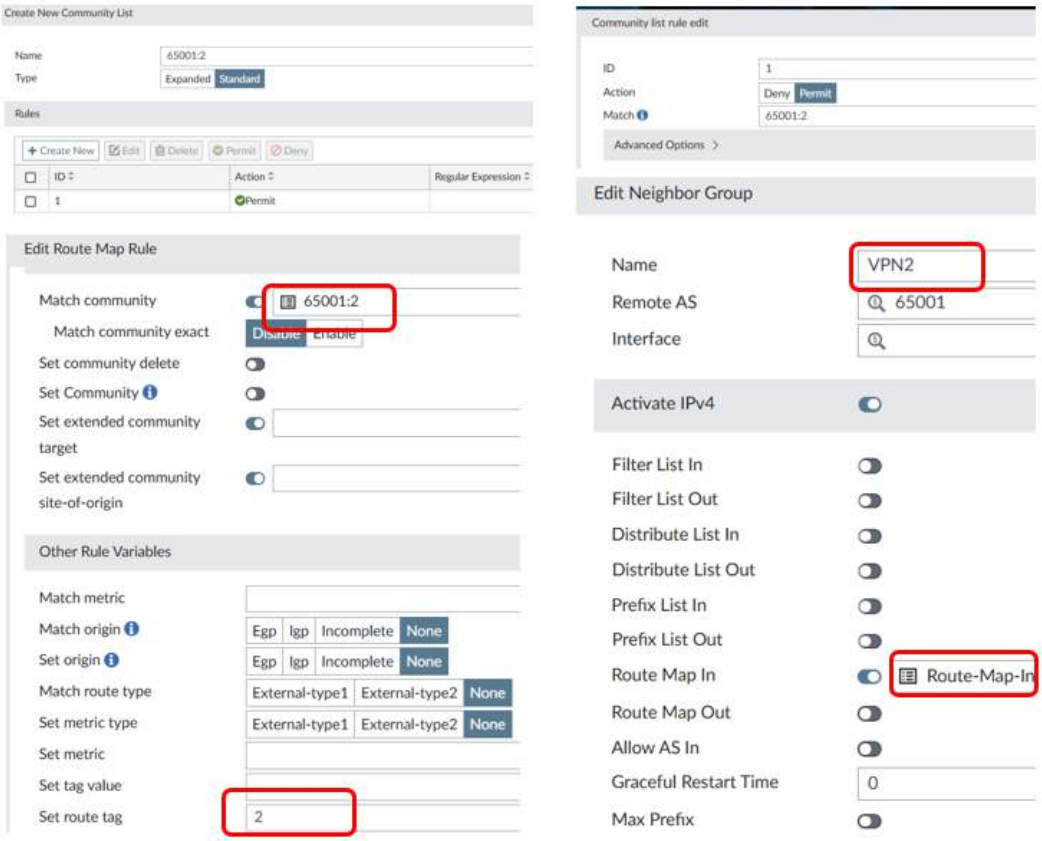
HUB1 VPN2 BGP Configuration

For the second VPN, create clones of both the Neighbor group and the neighbor range. For the new VPN2 neighbor range, define the new prefix and associate with the new neighbor group created i.e. VPN2. This is the logic used to provision additional tunnels at HUB1 and HUB2 for the dual HUB setup in later chapters.



HUB1 BGP Route-Map-In Neighbor2

For the second VPN, a clone of VPN1 is used and while the same route-map is referenced, new communities (65001:2) and a new route tag (2) are created. The same route map is still used to match out of SLA routes with additional match and set options for 65001:11 and route tag 11.



HUB1 VPN2 Configuration Preview.

```
config router community-list
    edit "65001:11"
        config rule
            edit 1
                set action permit
                set match "65001:11"
            next
        end
    next
    edit "65001:2"
        config rule
            edit 1
                set action permit
                set match "65001:2"
            next
        end
    next
end
config router route-map
    edit "Route-Map-In"
        config rule
            edit 2
                set match-community "65001:2"
                set set-route-tag "2"
            next
            edit 11
                set match-community "65001:11"
                set set-route-tag "11"
            next
        end
    next
end
config router bgp
    config neighbor-group
        edit "VPN2"
            set capability-graceful-restart enable
            set link-down-failover enable
            set next-hop-self enable
            set soft-reconfiguration enable
            set remote-as 65001
            set route-map-in "Route-Map-In"
            set additional-path send
            set route-reflector-client enable
        next
    end
    config neighbor-range
        edit 2
            set prefix 10.50.2.0 255.255.255.0
            set neighbor-group "VPN2"
        next
    end
end
```

These are all the commands pushed to the HUB for the second neighbor to be established. The community-list to match the OUT-OF-SLA state is configured here too i.e. 65001:11

HUB1 BGP Routing Verification after two Tunnels are Established.

Two neighborships are established between HUB1 and Branch1 over the two tunnels. Redundant routes are shared for each network learnt from the branch. Route Tags are mapped based on the Tunnel/neighbor advertising the network from the branch.

```
HUB1 # get router info bgp summary

VRF 0 BGP router identifier 10.100.100.100, local AS number 65001
BGP table version is 2
1 BGP AS-PATH entries
2 BGP community entries
Next peer check timer due in 27 seconds

Neighbor   V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.50.1.10 4      65001       7       7        2    0    0 00:02:21        1
10.50.2.10 4      65001       7       7        2    0    0 00:02:21        1

Total number of neighbors 2


HUB1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 10.50.1.10 (recursive is directly connected, VPN1), 00:01:38, [1/0]
                     [200/0] via 10.50.2.10 (recursive is directly connected, VPN2), 00:01:38, [1/0]
```

```
HUB1 # get router info bgp network
VRF 0 BGP table version is 2, local router ID is 10.100.100.100
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.10.1.0/24     10.50.1.10       0             100      0        1 i <-/1>
*>i                 10.50.2.10       0             100      0        2 i <-/2>
*> 10.10.100.0/24   0.0.0.0                        100  32768        0 i <-/1>

Total number of prefixes 2


HUB1 # get router info bgp network 10.10.1.0/24
VRF 0 BGP routing table entry for 10.10.1.0/24
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to peer-groups:
  VPN1 VPN2
  Original VRF 0
  Local, (Received from a RR-client)
    10.50.1.10 from 10.50.1.10 (10.100.100.1)
      Origin IGP metric 0, route tag 1, localpref 100, valid, internal, best
      Community: 65001:1
      Advertised Path ID: 1
       Last update: Wed Oct 30 12:57:48 2024

  Original VRF 0
  Local, (Received from a RR-client)
    10.50.2.10 from 10.50.2.10 (10.100.100.1)
      Origin IGP metric 0, route tag 2, localpref 100, valid, internal, best
      Community: 65001:2
      Advertised Path ID: 2
       Last update: Wed Oct 30 12:57:51 2024
```

Branch1 BGP Routing Verification after two Tunnels are Established

Two neighborships are established between HUB1 and Branch1 over the two tunnels with the upstream tunnel static IP addresses as the neighbor. Redundant routes are received for each network learnt from HUB1.

```
BR1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.100.0/24 [200/0] via 10.50.1.1 (recursive via HUB1-VPN1 tunnel 100.1.100.1), 00:14:42, [1/0]
                       [200/0] via 10.50.2.1 (recursive via HUB1-VPN2 tunnel 200.2.100.1), 00:14:42, [1/0]


BR1 # get router info bgp summary

VRF 0 BGP router identifier 10.100.100.1, local AS number 65001
BGP table version is 2
1 BGP AS-PATH entries
2 BGP community entries

Neighbor  V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.50.1.1 4      65001      21      23        2    0    0 00:15:24        1
10.50.2.1 4      65001      22      22        2    0    0 00:15:24        1

Total number of neighbors 2
```

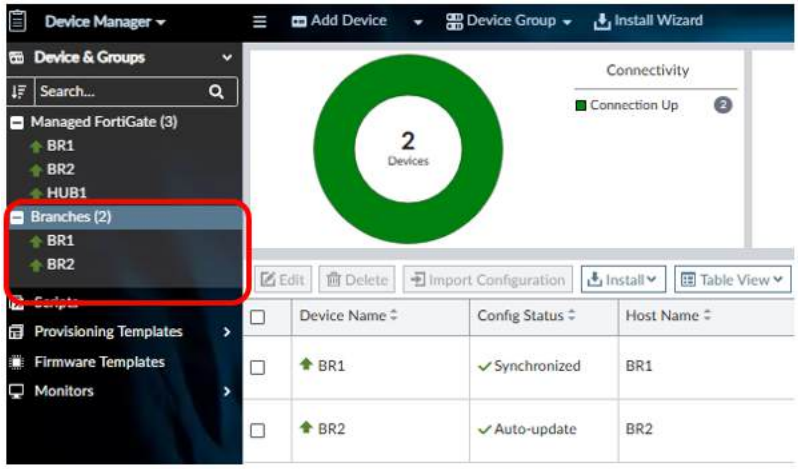
```
BR1 # get router info bgp network
VRF 0 BGP table version is 2, local router ID is 10.100.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.1.0/24     0.0.0.0                       100  32768        0 i <-/1>
*>i10.10.100.0/24   10.50.2.1       0             100      0        0 i <1/2>
*>i                 10.50.1.1       0             100      0        0 i <1/1>

Total number of prefixes 2


BR1 # get router info bgp network 10.10.100.0/24
VRF 0 BGP routing table entry for 10.10.100.0/24
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local
    10.50.2.1 from 10.50.2.1 (10.100.100.100)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Receive Path ID: 1
      Advertised Path ID: 2
       Last update: Wed Oct 30 12:57:29 2024

  Original VRF 0
  Local
    10.50.1.1 from 10.50.1.1 (10.100.100.100)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Receive Path ID: 1
      Advertised Path ID: 1
       Last update: Wed Oct 30 12:57:26 2024
```

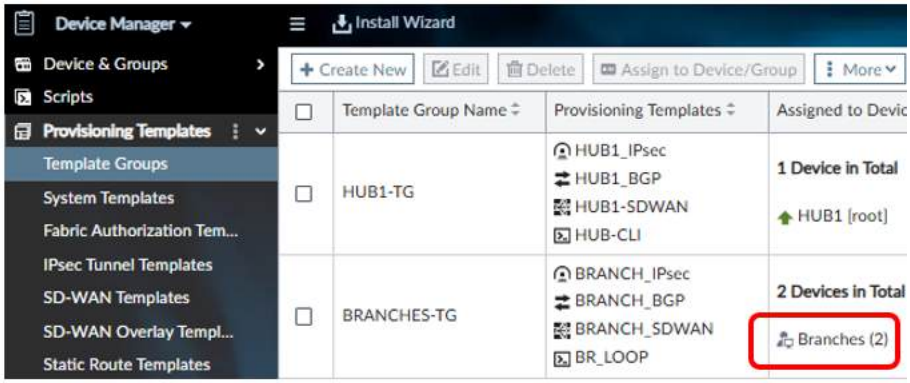
Configuring Additional Branches.

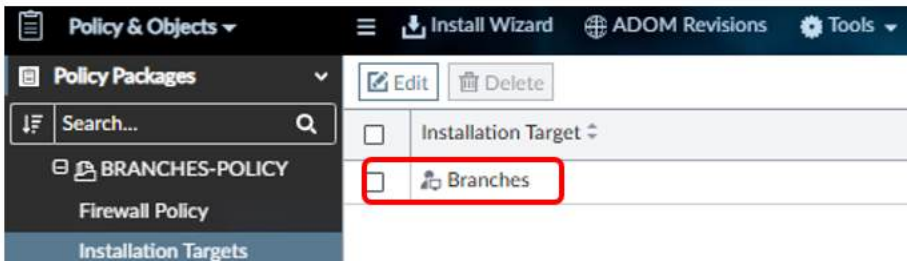
The modelling part is complete after bringing up the additional tunnels. **No configuration is required at the HUB to add additional branches**. At this point, additional branches are configured in the order listed below:

- Register additional branches to FortiManager. (Branch2 already registered for this demo).
- Define Metadata variables for the new branch i.e. Local-ID, branch-ID etc.
- Push Device settings first to the new branch to avoid conflicts when you send the Firewall policy.
- Push centralized Firewall policy.

The additional Branches are now grouped in the Device group and the appropriate mapping done to the Template Group and Firewall policy.





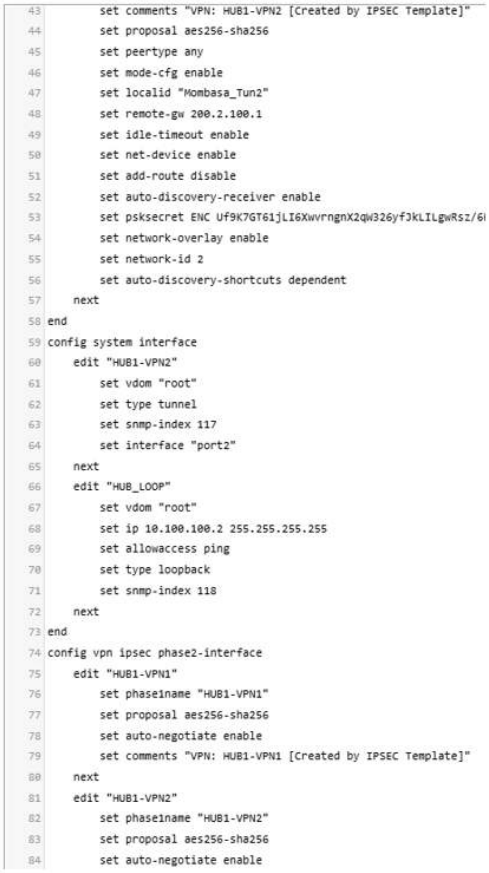


Device Settings Installation to Branch2

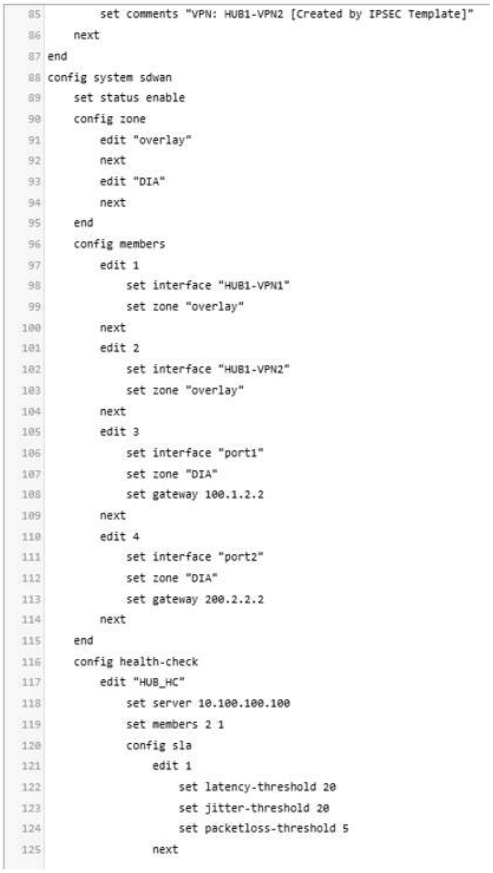
After the device settings are sent to Branch 2, push the firewall policies. Additional branches are provisioned in a similar way if they have the same interface mapping. Customization is required if there are variations with the WAN links, different model of firewalls etc. Unlike the initial modelling of branch1 where we pushed IPSEC, SDWAN and then BGP separately, additional branches receive all the configurations at once.

For deployments with multiple sites, it is recommended to model multiple sites since variations likely exist per site e.g. for a deployment with 100 sites, 5 -10 sites could account for all possible variations. Most of the work in most projects is in the initial modelling.

```
config system sdwan
    config health-check
        delete "Default_Office_365"
        delete "Default_Google Search"
        delete "Default_Gmail"
        delete "Default_FortiGuard"
        delete "Default_DNS"
        delete "Default_AWS"
    end
end
config vpn ipsec phase1-interface
    edit "HUB1-VPN1"
        set interface "port1"
        set ike-version 2
        set comments "VPN: HUB1-VPN1 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set localid "Mombasa_Tun1"
        set remote-gw 100.1.100.1
        set idle-timeout enable
        set net-device enable
        set add-route disable
        set auto-discovery-receiver enable
        set psksecret ENC Uf9K7GT61jLI6XwvrngnX2qW326yfJkLILgwRsz/6Hs
        set network-overlay enable
        set network-id 1
        set auto-discovery-shortcuts dependent
    next
end
config system interface
    edit "HUB1-VPN1"
        set vdom "root"
        set type tunnel
        set snmp-index 116
        set interface "port1"
    next
end
config vpn ipsec phase1-interface
    edit "HUB1-VPN2"
        set interface "port2"
        set ike-version 2
```



```
        set comments "VPN: HUB1-VPN2 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set localid "Mombasa_Tun2"
        set remote-gw 200.2.100.1
        set idle-timeout enable
        set net-device enable
        set add-route disable
        set auto-discovery-receiver enable
        set psksecret ENC Uf9K7GT61jLI6XwvrngnX2qW326yfJkLILgwRsz/6I
        set network-overlay enable
        set network-id 2
        set auto-discovery-shortcuts dependent
    next
end
config system interface
    edit "HUB1-VPN2"
        set vdom "root"
        set type tunnel
        set snmp-index 117
        set interface "port2"
    next
    edit "HUB_LOOP"
        set vdom "root"
        set ip 10.100.100.2 255.255.255.255
        set allowaccess ping
        set type loopback
        set snmp-index 118
    next
end
config vpn ipsec phase2-interface
    edit "HUB1-VPN1"
        set phase1name "HUB1-VPN1"
        set proposal aes256-sha256
        set auto-negotiate enable
        set comments "VPN: HUB1-VPN1 [Created by IPSEC Template]"
    next
    edit "HUB1-VPN2"
        set phase1name "HUB1-VPN2"
        set proposal aes256-sha256
        set auto-negotiate enable
```



```
        set comments "VPN: HUB1-VPN2 [Created by IPSEC Template]"
    next
end
config system sdwan
    set status enable
    config zone
        edit "overlay"
        next
        edit "DIA"
        next
    end
    config members
        edit 1
            set interface "HUB1-VPN1"
            set zone "overlay"
        next
        edit 2
            set interface "HUB1-VPN2"
            set zone "overlay"
        next
        edit 3
            set interface "port1"
            set zone "DIA"
            set gateway 100.1.2.2
        next
        edit 4
            set interface "port2"
            set zone "DIA"
            set gateway 200.2.2.2
        next
    end
    config health-check
        edit "HUB_HC"
            set server 10.100.100.100
            set members 2 1
            config sla
                edit 1
                    set latency-threshold 20
                    set jitter-threshold 20
                    set packetloss-threshold 5
                next
```

```
            end
        next
    end
end
config router route-map
    edit "OUT-OF-SLA"
        config rule
            edit 1
                set set-community "65001:11"
            next
        end
    next
    edit "IN-SLA-TUN1"
        config rule
            edit 1
                set set-community "65001:1"
            next
        end
    next
end
config router bgp
    config neighbor
        edit "10.50.1.1"
            set advertisement-interval 1
            set capability-graceful-restart enable
            set link-down-failover enable
            set soft-reconfiguration enable
            set description "HUB1-VPN1"
            set interface "HUB1-VPN1"
            set remote-as 65001
            set route-map-out "OUT-OF-SLA"
            set connect-timer 10
            set additional-path receive
            set route-map-out-preferable "IN-SLA-TUN1"
        next
    end
end
config system sdwan
    config neighbor
        edit "10.50.1.1"
            set health-check "HUB_HC"
            set sla-id 1
```

```
            set sla-id 1
            set member 1
        next
    end
end
config router route-map
    edit "IN-SLA-TUN2"
        config rule
            edit 1
                set set-community "65001:2"
            next
        end
    next
end
config router bgp
    config neighbor
        edit "10.50.2.1"
            set advertisement-interval 1
            set capability-graceful-restart enable
            set link-down-failover enable
            set soft-reconfiguration enable
            set description "HUB1-VPN1"
            set interface "HUB1-VPN2"
            set remote-as 65001
            set route-map-out "OUT-OF-SLA"
            set connect-timer 10
            set additional-path receive
            set route-map-out-preferable "IN-SLA-TUN2"
        next
    end
end
config system sdwan
    config neighbor
        edit "10.50.2.1"
            set health-check "HUB_HC"
            set sla-id 1
            set member 2
        next
    end
end
config router bgp
```

```
    set as 65001
    set router-id 10.100.100.2
    set ibgp-multipath enable
    set additional-path enable
    set graceful-restart enable
    set additional-path-select 255
    config network
        edit 1
            set prefix 10.10.2.0 255.255.255.0
        next
    end
    set recursive-next-hop enable
end
```

Routing Review at Branch1 & Branch 2

Each branch will have redundant links to the remote sites. The next-hop for the remote peer is not changed as it traverses HUB1 because the branches are configured as route-reflector clients at the HUB.

```
BR1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.2.0/24 [200/0] via 10.50.1.11 [2] (recursive is directly connected, HUB1-VPN1), 00:04:14, [1/0]
                     [200/0] via 10.50.2.11 [2] (recursive is directly connected, HUB1-VPN2), 00:04:14, [1/0]
B       10.10.100.0/24 [200/0] via 10.50.1.1 (recursive via HUB1-VPN1 tunnel 100.1.100.1), 00:09:34, [1/0]
                       [200/0] via 10.50.2.1 (recursive via HUB1-VPN2 tunnel 200.2.100.1), 00:09:34, [1/0]

BR1 # get router info bgp summary

VRF 0 BGP router identifier 10.100.100.1, local AS number 65001
BGP table version is 7
1 BGP AS-PATH entries
2 BGP community entries

Neighbor  V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.50.1.1 4      65001      34      24        7    0    0 00:10:11        3
10.50.2.1 4      65001      35      22        6    0    0 00:10:11        3

Total number of neighbors 2
```

```
BR2 # get router info routing-table  bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 10.50.1.10 [2] (recursive is directly connected, HUB1-VPN1), 00:02:40, [1/0]
                     [200/0] via 10.50.2.10 [2] (recursive is directly connected, HUB1-VPN2), 00:02:40, [1/0]
B       10.10.100.0/24 [200/0] via 10.50.1.1 (recursive via HUB1-VPN1 tunnel 100.1.100.1), 00:03:06, [1/0]
                       [200/0] via 10.50.2.1 (recursive via HUB1-VPN2 tunnel 200.2.100.1), 00:03:06, [1/0]

BR2 # get router info bgp summary

VRF 0 BGP router identifier 10.100.100.2, local AS number 65001
BGP table version is 2
1 BGP AS-PATH entries
2 BGP community entries

Neighbor  V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.50.1.1 4      65001      58      37        1    0    0 00:03:47        3
10.50.2.1 4      65001      53      37        1    0    0 00:03:48        3

Total number of neighbors 2
```

```
BR1 # get router info bgp network
VRF 0 BGP table version is 2, local router ID is 10.100.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 10.10.1.0/24     0.0.0.0                           100  32768        0 i <-/1>
*>i10.10.2.0/24     10.50.2.11       0                100      0        0 i <2/4>
*>i                 10.50.2.11       0                100      0        0 i <2/2>
*>i                 10.50.1.11       0                100      0        0 i <1/3>
*>i                 10.50.1.11       0                100      0        0 i <1/1>
*>i10.10.100.0/24   10.50.1.1        0                100      0        0 i <1/1>
*>i                 10.50.2.1        0                100      0        0 i <1/2>

Total number of prefixes 3
```

```
BR2 # get router info bgp network
VRF 0 BGP table version is 2, local router ID is 10.100.100.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.10.1.0/24     10.50.2.10       0                100      0        0 i <2/2>
*>i                 10.50.1.10       0                100      0        0 i <1/1>
*>i                 10.50.2.10       0                100      0        0 i <2/4>
*>i                 10.50.1.10       0                100      0        0 i <1/3>
*> 10.10.2.0/24     0.0.0.0                           100  32768        0 i <-/1>
*>i10.10.100.0/24   10.50.1.1        0                100      0        0 i <1/1>
*>i                 10.50.2.1        0                100      0        0 i <1/2>

Total number of prefixes 3
```

Routing Review at the HUB

Redundant routes are received from each branch with the correct community per tunnel.

```
HUB1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 10.50.1.10 (recursive is directly connected, VPN1), 00:09:17, [1/0]
                     [200/0] via 10.50.2.10 (recursive is directly connected, VPN2), 00:09:17, [1/0]
B       10.10.2.0/24 [200/0] via 10.50.1.11 (recursive is directly connected, VPN1), 00:09:17, [1/0]
                     [200/0] via 10.50.2.11 (recursive is directly connected, VPN2), 00:09:17, [1/0]

HUB1 # get router info bgp network
VRF 0 BGP table version is 15, local router ID is 10.100.100.100
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.10.1.0/24     10.50.1.10      0            100      0       1 i <-/1>
*>i                 10.50.2.10      0            100      0       2 i <-/2>
*>i10.10.2.0/24     10.50.1.11      0            100      0       1 i <-/1>
*>i                 10.50.2.11      0            100      0       2 i <-/2>
*> 10.10.100.0/24   0.0.0.0                      100  32768       0 i <-/1>

Total number of prefixes 3
```

```
HUB1 # get router info bgp summary

VRF 0 BGP router identifier 10.100.100.100, local AS number 65001
BGP table version is 15
1 BGP AS-PATH entries
2 BGP community entries
Next peer check timer due in 4 seconds

Neighbor    V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.50.1.10 4      65001      89      117       15    0    0 00:37:16        1
10.50.1.11 4      65001      85      114       15    0    0 00:37:16        1
10.50.2.10 4      65001      91      122       15    0    0 00:37:16        1
10.50.2.11 4      65001      87      111       15    0    0 00:37:17        1

Total number of neighbors 4
```

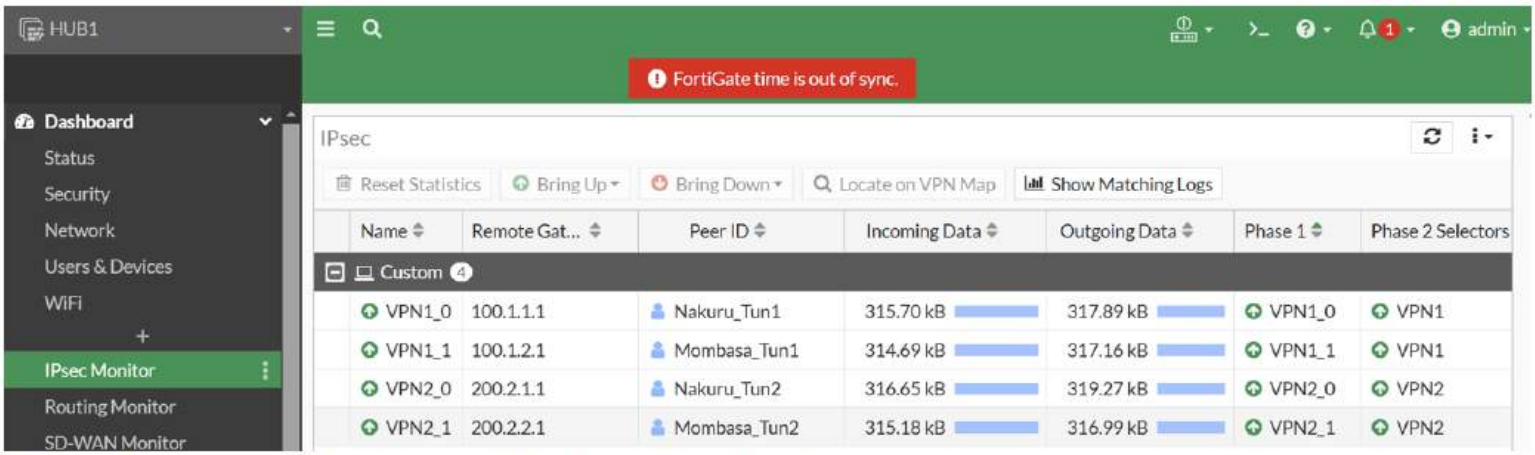
```
HUB1 # get router info bgp network 10.10.1.0/24
VRF 0 BGP routing table entry for 10.10.1.0/24
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to peer-groups:
  VPN1 VPN2
  Original VRF 0
  Local, (Received from a RR-client)
    10.50.1.10 from 10.50.1.10 (10.100.100.1)
      Origin IGP metric 0, route tag 1, localpref 100, valid, internal, best
      Community: 65001:1
      Advertised Path ID: 1
       Last update: Thu Oct 31 10:35:16 2024

  Original VRF 0
  Local, (Received from a RR-client)
    10.50.2.10 from 10.50.2.10 (10.100.100.1)
      Origin IGP metric 0, route tag 2, localpref 100, valid, internal, best
      Community: 65001:2
      Advertised Path ID: 2
       Last update: Thu Oct 31 10:35:16 2024
```

```
HUB1 # get router info bgp network 10.10.2.0/24
VRF 0 BGP routing table entry for 10.10.2.0/24
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to peer-groups:
  VPN1 VPN2
  Original VRF 0
  Local, (Received from a RR-client)
    10.50.1.11 from 10.50.1.11 (10.100.100.2)
      Origin IGP metric 0, route tag 1, localpref 100, valid, internal, best
      Community: 65001:1
      Advertised Path ID: 1
       Last update: Thu Oct 31 10:35:16 2024

  Original VRF 0
  Local, (Received from a RR-client)
    10.50.2.11 from 10.50.2.11 (10.100.100.2)
      Origin IGP metric 0, route tag 2, localpref 100, valid, internal, best
      Community: 65001:2
      Advertised Path ID: 2
       Last update: Thu Oct 31 10:35:16 2024
```

Tunnel IP address assignment

The HUB acts as the DHCP server on each tunnel. Giving more descriptive local IDs help in the quick identification of the peer IDs. To check actual ip address assignment, verify using the command:

```
diagnose vpn ike gateway list
```

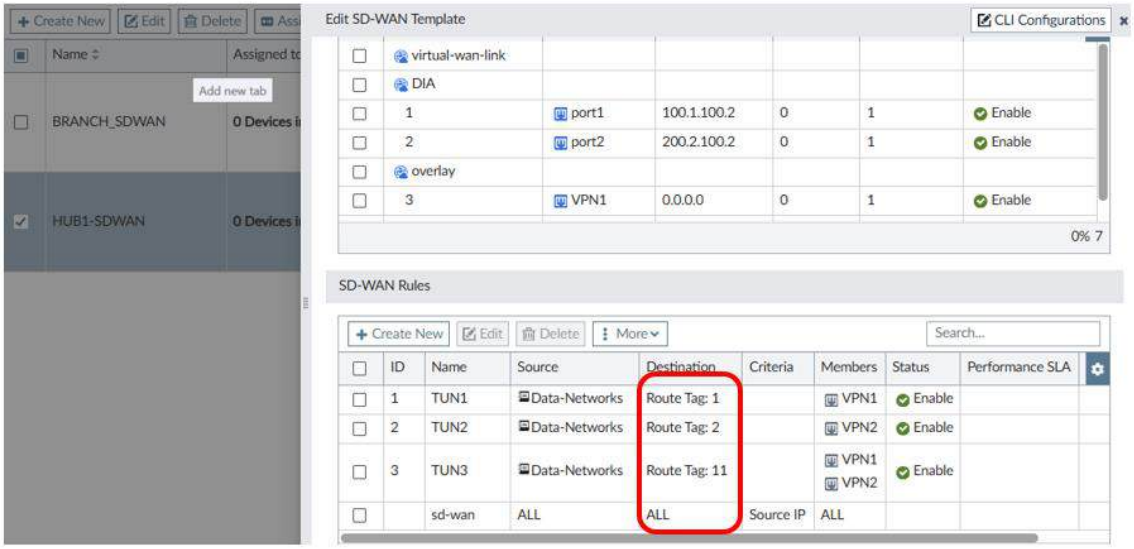



```
HUB1 # diagnose vpn ike gateway list | grep assigned
assigned IPv4 address: 10.50.1.10/255.255.255.0
assigned IPv4 address: 10.50.2.10/255.255.255.0
assigned IPv4 address: 10.50.2.11/255.255.255.0
assigned IPv4 address: 10.50.1.11/255.255.255.0
```

HUB SDWAN Rules

At the hub, SDWAN rules are defined using the route tags obtained from the information shared by the branches. This avoids duplicate healthcheck configuration to probe the same links from the HUB towards Branches.

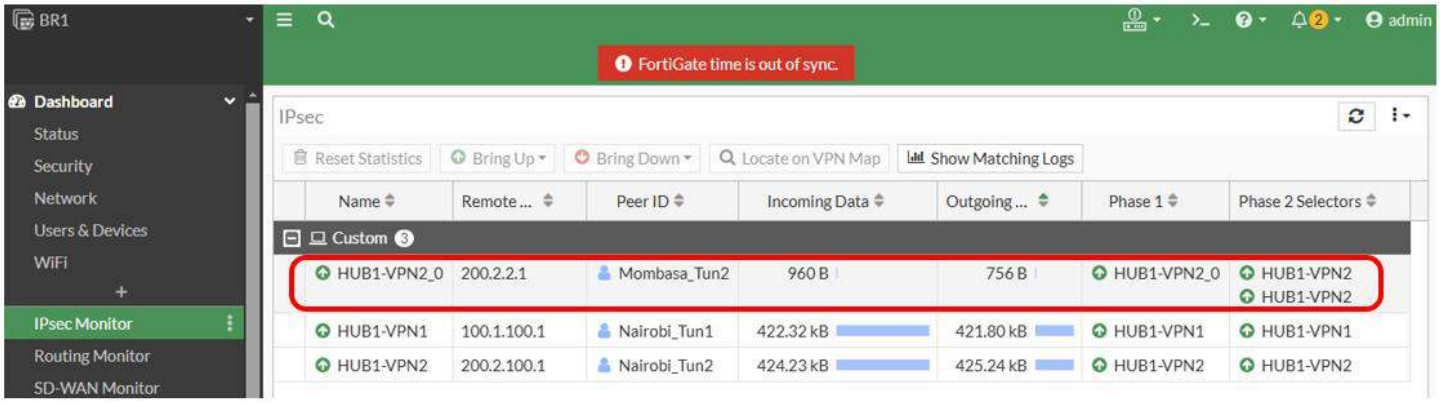
It should be noted that all routes within and out of SLA will still be available in the routing table. The Tags simply define link selection but are not route filters. SDWAN rules are involved in the routing process, firewall policies are still required to allow the traffic flow.

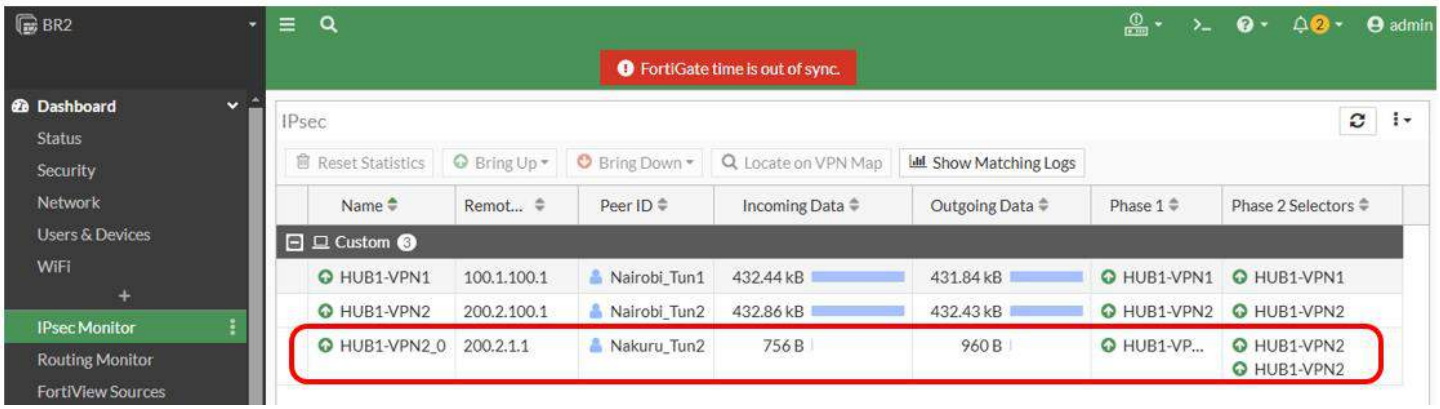



```
config system sdwan
    config service
        edit 1
            set name "TUN1"
            set route-tag 1
            set src "Data-Networks"
            set priority-members 3
        next
        edit 2
            set name "TUN2"
            set route-tag 2
            set src "Data-Networks"
            set priority-members 4
        next
        edit 3
            set name "TUN3"
            set route-tag 11
            set src "Data-Networks"
            set priority-members 3 4
        next
    end
```

ADVPN

The IPSec and BGP templates used earlier were ADVPN ready. Since the underlay routing is using default routing, ADVPN tunnels will be established when spoke-to-spoke traffic is initiated, in this case traffic initiated from Branch1 to Branch2 or vice versa.





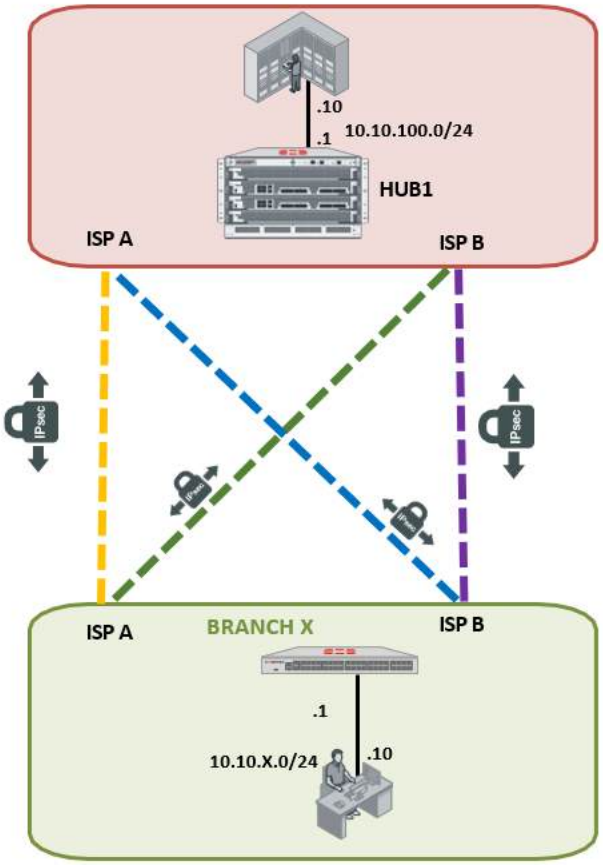

```
BR1 # diagnose sys sdwan health-check
Health Check(HUB_HC):
Seq(2 HUB1-VPN2): state(alive), packet-loss(0.000%) latency(4.618), jitter(2.992), mos(4.399), bandwidth-up(999999), ba
ndwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(2 HUB1-VPN2_0): state(dead), packet-loss(100.000%) sla_map=0x0
Seq(1 HUB1-VPN1): state(alive), packet-loss(0.000%) latency(4.324), jitter(3.892), mos(4.398), bandwidth-up(999999), ba
ndwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
```

```
BR2 # diagnose sys sdwan health-check
Health Check(HUB_HC):
Seq(2 HUB1-VPN2): state(alive), packet-loss(0.000%) latency(3.169), jitter(1.693), mos(4.401), bandwidth-up(999999), ba
ndwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(2 HUB1-VPN2_0): state(dead), packet-loss(100.000%) sla_map=0x0
Seq(1 HUB1-VPN1): state(alive), packet-loss(0.000%) latency(3.255), jitter(2.431), mos(4.400), bandwidth-up(999999), ba
ndwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
```

Link Redundancy Single HUB dual ISP

To enhance redundancy, configure cross-connect links to ensure branches can communicate with the HUB regardless of the ISP link available i.e. any single link at either side can provide access.

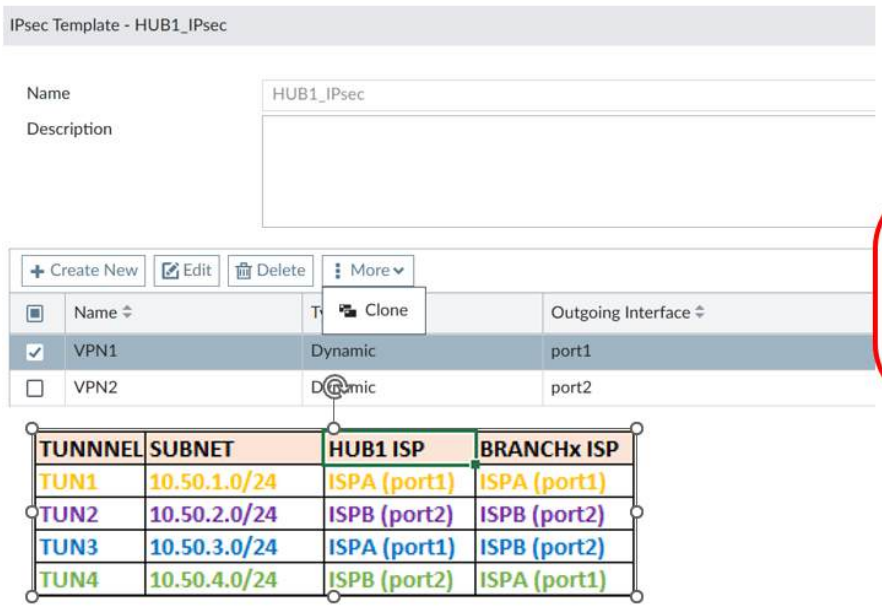
Great caution should be observed when configuring the tunnels when selecting local exit interfaces, destination overlays and Tunnel static IP addresses e.g. Tunnel 3 has an exit of ISPB at the HUB but ISPA from the branches. When provisioning additional tunnels, follow the same sequence used earlier in provisioning to avoid provisioning errors.



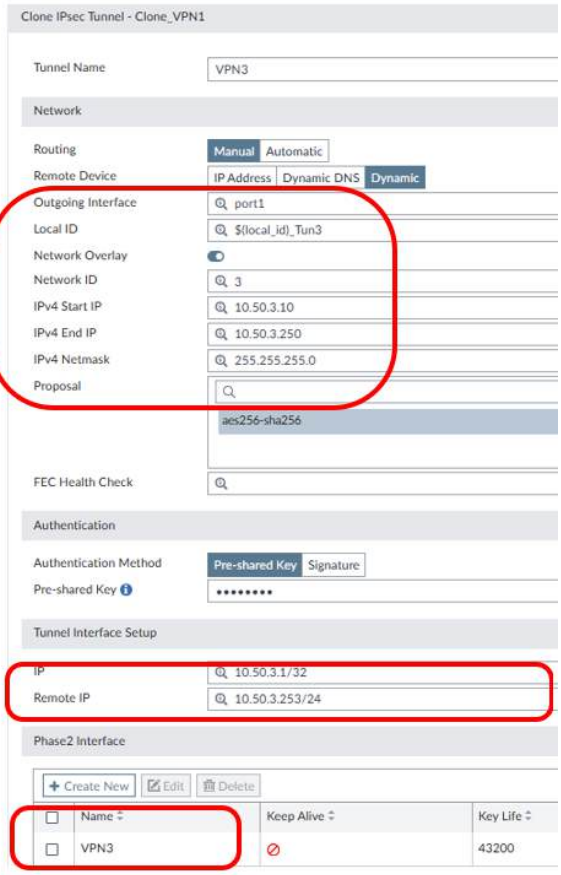
TUNNNEL	SUBNET	HUB1 ISP	BRANCHx ISP
TUN1	10.50.1.0/24	ISPA (port1)	ISPA (port1)
TUN2	10.50.2.0/24	ISPB (port2)	ISPB (port2)
TUN3	10.50.3.0/24	ISPA (port1)	ISPB (port2)
TUN4	10.50.4.0/24	ISPB (port2)	ISPA (port1)

HUB1 IPSEC Provisioning for VPN3

From HUB1 perspective, Tun3 connects to ISPA (port1). Clone the existing VPN tunnels under HUB1_IPsec Template.

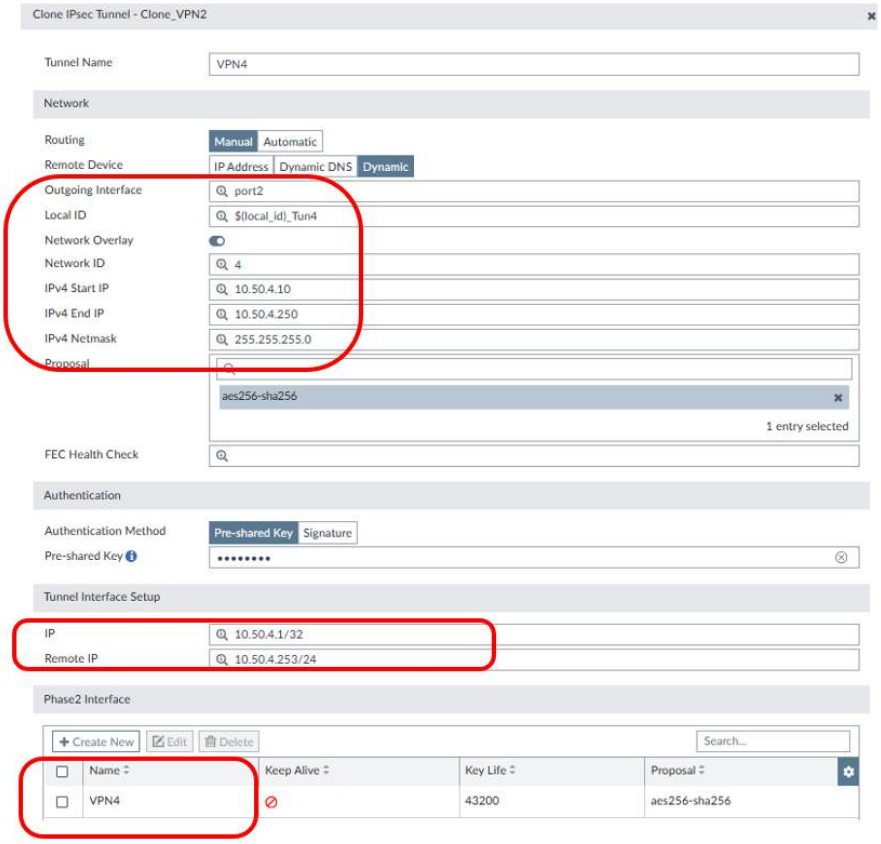


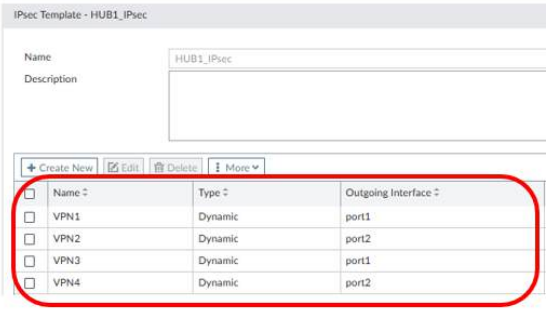
TUNNNEL	SUBNET	HUB1 ISP	BRANCHx ISP
TUN1	10.50.1.0/24	ISPA (port1)	ISPA (port1)
TUN2	10.50.2.0/24	ISPB (port2)	ISPB (port2)
TUN3	10.50.3.0/24	ISPA (port1)	ISPB (port2)
TUN4	10.50.4.0/24	ISPB (port2)	ISPA (port1)



HUB1 IPSEC Provisioning for VPN4

From HUB1 perspective, Tun4 connects to ISPA (port2). Clone the existing VPN tunnels under HUB2_IPsec Template.





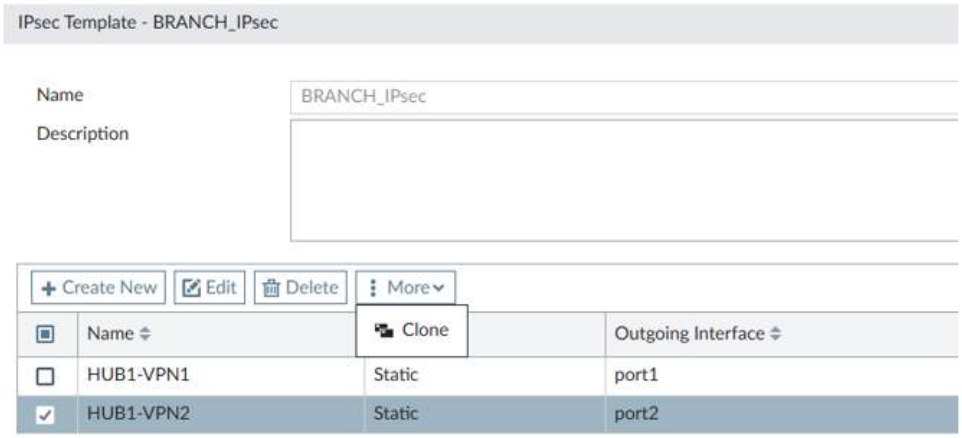
TUNNNEL	SUBNET	HUB1 ISP	BRANCHx ISP
TUN1	10.50.1.0/24	ISPA (port1)	ISPA (port1)
TUN2	10.50.2.0/24	ISPB (port2)	ISPB (port2)
TUN3	10.50.3.0/24	ISPA (port1)	ISPB (port2)
TUN4	10.50.4.0/24	ISPB (port2)	ISPA (port1)

HUB1 IPSEC Provisioning Preview for VPN3 & VPN4

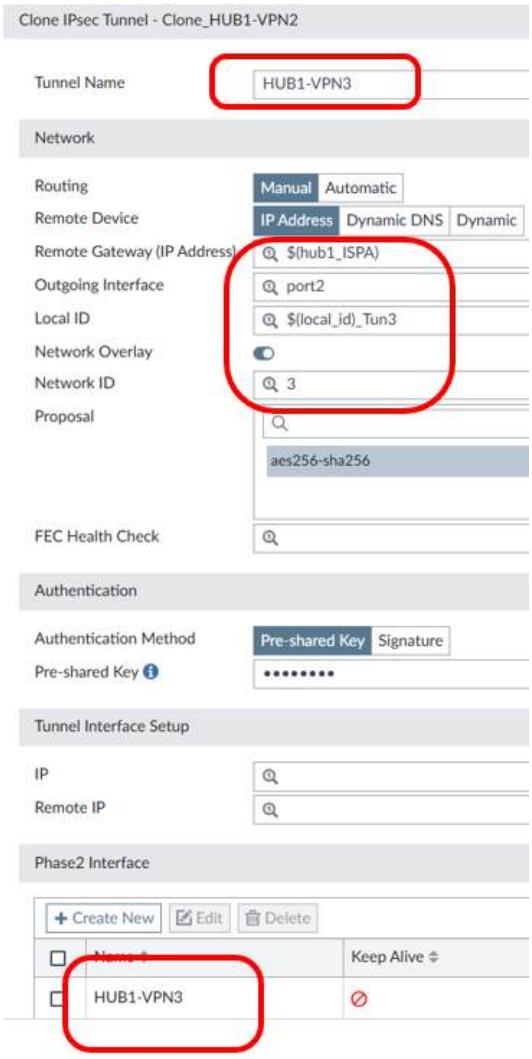
```
config vpn ipsec phase1-interface
    edit "VPN3"
        set type dynamic
        set interface "port1"
        set ike-version 2
        set dpd on-idle
        set comments "VPN: VPN3 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set localid "Nairobi_Tun3"
        set dpd-retryinterval 60
        set net-device disable
        set add-route disable
        set auto-discovery-sender enable
        set ipv4-start-ip 10.50.3.10
        set ipv4-end-ip 10.50.3.250
        set ipv4-netmask 255.255.255.0
        set psksecret ENC Z8Zpc/bwU2j1HxCFsp0zLVsmpXEWQQvc6JVoYq8gt:
        set network-overlay enable
        set network-id 3
    next
end
config system interface
    edit "VPN3"
        set vdom "root"
        set ip 10.50.3.1 255.255.255.255
        set type tunnel
        set remote-ip 10.50.3.253 255.255.255.0
        set snmp-index 119
        set interface "port1"
    next
end
config vpn ipsec phase1-interface
    edit "VPN4"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set dpd on-idle
        set comments "VPN: VPN4 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set localid "Nairobi_Tun4"
        set dpd-retryinterval 60
        set net-device disable
        set add-route disable
        set auto-discovery-sender enable
        set ipv4-start-ip 10.50.4.10
        set ipv4-end-ip 10.50.4.250
        set ipv4-netmask 255.255.255.0
        set psksecret ENC Z8Zpc/bwU2j1HxCFsp0zLVsmpXEWQQvc6JVoYq8gt3RKcH0GzuF
        set network-overlay enable
        set network-id 4
    next
end
config system interface
    edit "VPN4"
        set vdom "root"
        set ip 10.50.4.1 255.255.255.255
        set type tunnel
        set remote-ip 10.50.4.253 255.255.255.0
        set snmp-index 120
        set interface "port2"
    next
end
config vpn ipsec phase2-interface
    edit "VPN3"
        set phase1name "VPN3"
        set proposal aes256-sha256
        set comments "VPN: VPN3 [Created by IPSEC Template]"
    next
    edit "VPN4"
        set phase1name "VPN4"
        set proposal aes256-sha256
        set comments "VPN: VPN4 [Created by IPSEC Template]"
    next
end
```

Branches IPSEC Provisioning for HUB1-VPN3

From the branch perspective, Tun3 connects to ISPB (port2). The remote gateway changes to ISPA underlay at HUB1. Clone the existing HUB1-VPNX tunnels under BRANCH_IPsec Template and modify settings.

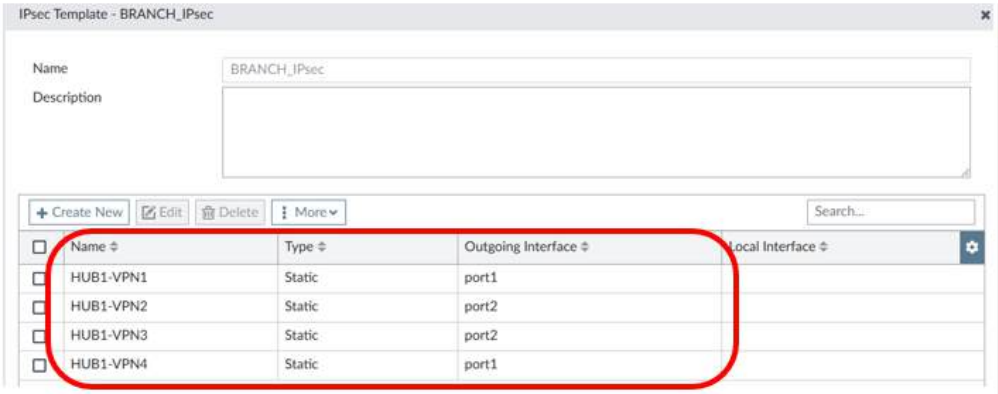


TUNNNEL	SUBNET	HUB1 ISP	BRANCHx ISP
TUN1	10.50.1.0/24	ISPA (port1)	ISPA (port1)
TUN2	10.50.2.0/24	ISPB (port2)	ISPB (port2)
TUN3	10.50.3.0/24	ISPA (port1)	ISPB (port2)
TUN4	10.50.4.0/24	ISPB (port2)	ISPA (port1)

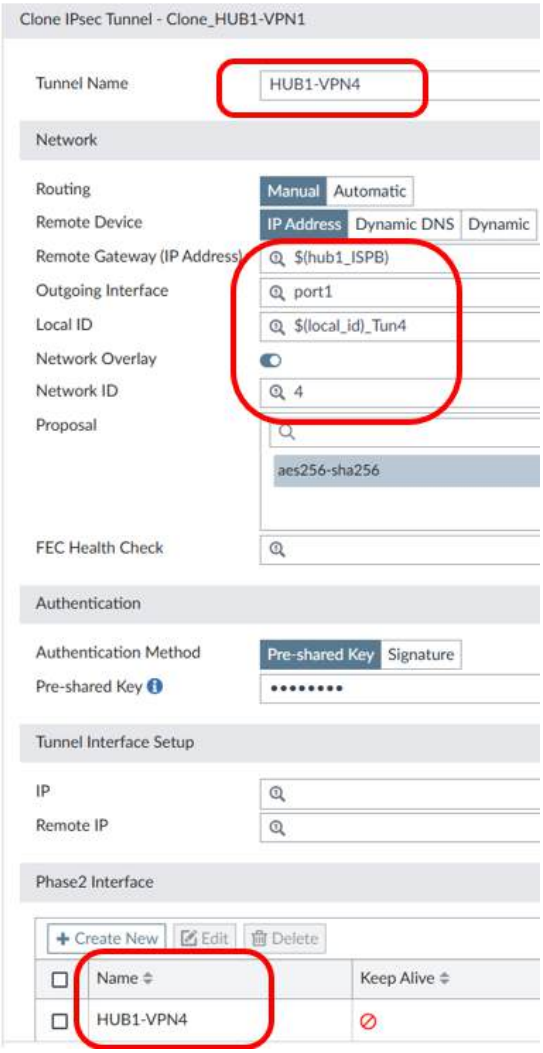


Branches IPSEC Provisioning for HUB1-VPN4

From the branch perspective, Tun4 connects to ISPA (port1). The remote gateway changes to ISPB underlay at HUB1. Clone the existing HUB1-VPNX tunnels under BRANCH_IPsec Template and modify settings.



TUNNNEL	SUBNET	HUB1 ISP	BRANCHx ISP
TUN1	10.50.1.0/24	ISPA (port1)	ISPA (port1)
TUN2	10.50.2.0/24	ISPB (port2)	ISPB (port2)
TUN3	10.50.3.0/24	ISPA (port1)	ISPB (port2)
TUN4	10.50.4.0/24	ISPB (port2)	ISPA (port1)



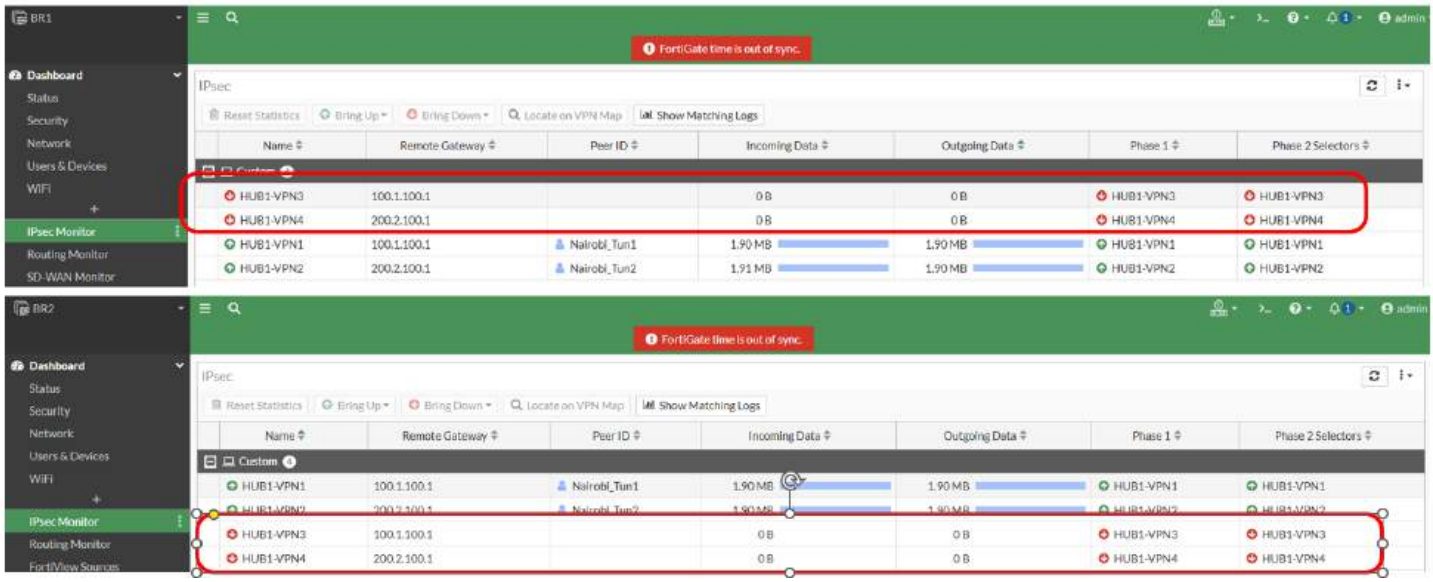
BRANCH IPSEC Provisioning Preview for HUB1-VPN3 & HUB1-VPN4

The two tunnels are created on branch1 and branch2. When we provision additional sites, they will create all 4 tunnels at once.

```
config vpn ipsec phase1-interface
    edit "HUB1-VPN3"
        set interface "port2"
        set ike-version 2
        set comments "VPN: HUB1-VPN3 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set localid "Nakuru_Tun3"
        set remote-gw 100.1.100.1
        set idle-timeout enable
        set net-device enable
        set add-route disable
        set auto-discovery-receiver enable
        set psksecret ENC Z8Zpc/bwU2j1HxCFsp0zLVsmpXEWQQvc6JVoYq8gt3R
        set network-overlay enable
        set network-id 3
        set auto-discovery-shortcuts dependent
    next
end
config system interface
    edit "HUB1-VPN3"
        set vdom "root"
        set type tunnel
        set snmp-index 119
        set interface "port2"
    next
end
config vpn ipsec phase1-interface
    edit "HUB1-VPN4"
        set interface "port1"
        set ike-version 2
        set comments "VPN: HUB1-VPN4 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set localid "Nakuru_Tun4"
        set remote-gw 200.2.100.1
        set idle-timeout enable
        set net-device enable
        set add-route disable
        set auto-discovery-receiver enable
        set psksecret ENC Z8Zpc/bwU2j1HxCFsp0zLVsmpXEWQQvc6JVoYq8g
        set network-overlay enable
        set network-id 4
        set auto-discovery-shortcuts dependent
    next
end
config system interface
    edit "HUB1-VPN4"
        set vdom "root"
        set type tunnel
        set snmp-index 120
        set interface "port1"
    next
end
config vpn ipsec phase2-interface
    edit "HUB1-VPN3"
        set phase1name "HUB1-VPN3"
        set proposal aes256-sha256
        set auto-negotiate enable
        set comments "VPN: HUB1-VPN3 [Created by IPSEC Template]"
    next
    edit "HUB1-VPN4"
        set phase1name "HUB1-VPN4"
        set proposal aes256-sha256
        set auto-negotiate enable
        set comments "VPN: HUB1-VPN4 [Created by IPSEC Template]"
    next
end
```

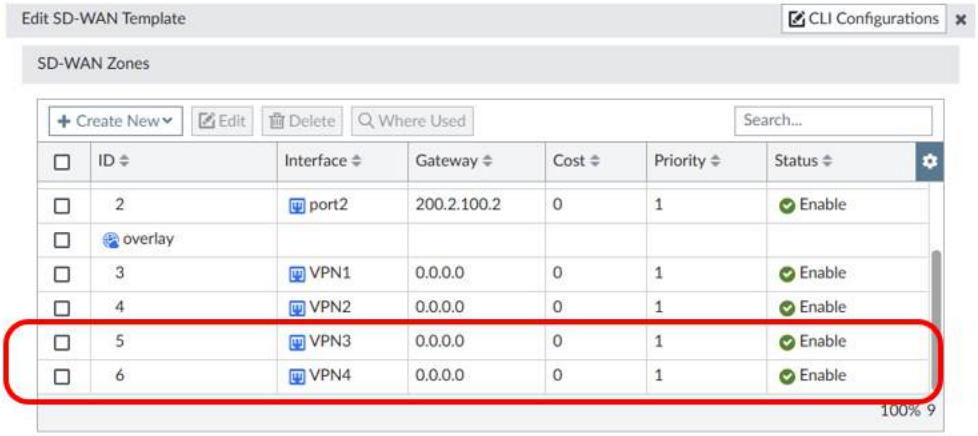
BRANCH IPSEC Verification for HUB1-VPN3 & HUB1-VPN4

The two tunnels will still be down at the branches since firewall policies outbound at the branches and inbound at the HUB for the new tunnels nor grouping them in SDWAN zones is not done. For this lab, the interfaces are grouped in sdwan zones to achieve the same.



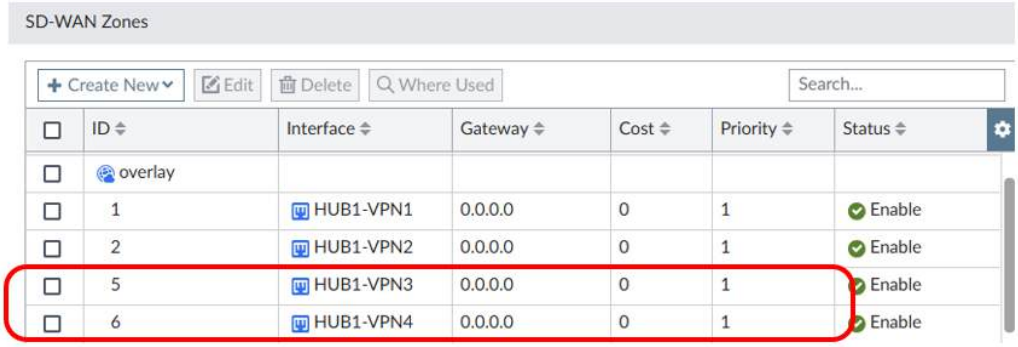
HUB1 VPN1 & VPN2 SDWAN overlay Zone Grouping

The two tunnels are grouped in the SDWAN overlay zone to have a total of 4 VPNx tunnels at HUB1



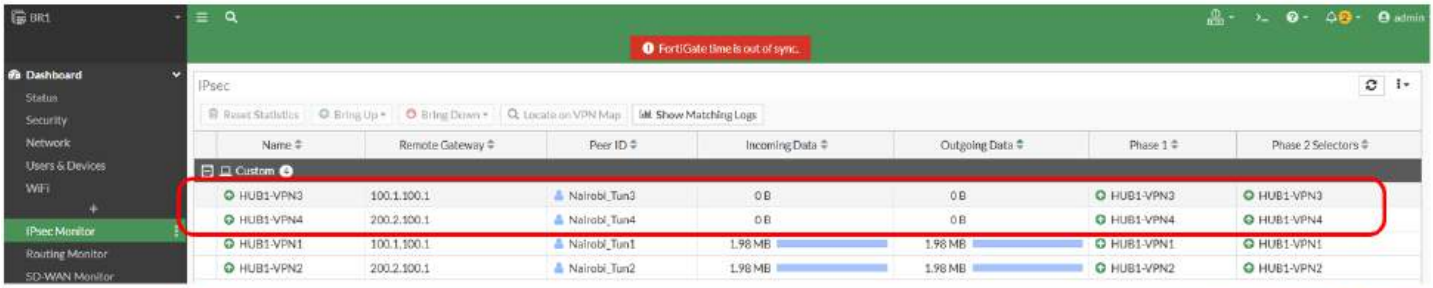

```
config system sdwan
    config members
        edit 5
            set interface "VPN3"
            set zone "overlay"
        next
        edit 6
            set interface "VPN4"
            set zone "overlay"
        next
    end
end
```

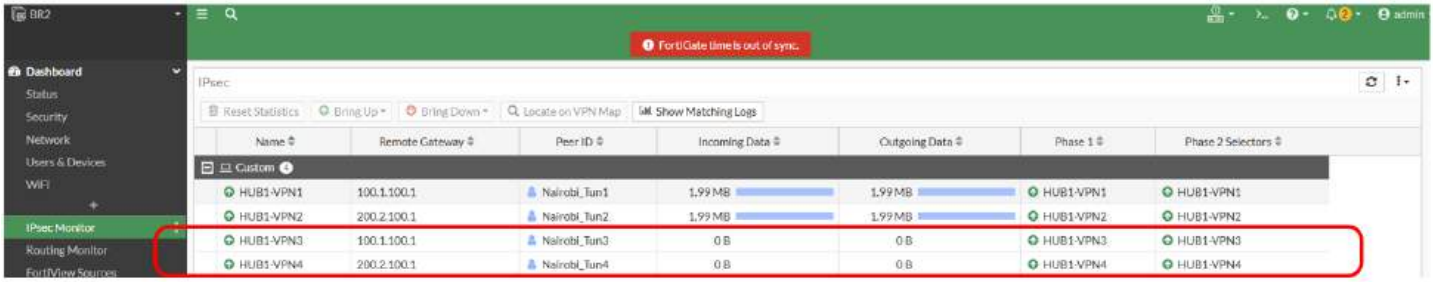
BRANCH HUB1-VPN3 & HUB1-VPN4 SDWAN overlay Zone Grouping




```
config system sdwan
    config members
        edit 5
            set interface "HUB1-VPN3"
            set zone "overlay"
        next
        edit 6
            set interface "HUB1-VPN4"
            set zone "overlay"
        next
    end
end
```

After grouping the new interfaces in sdwan zones at the branches and HUB1, the new tunnels will come up.





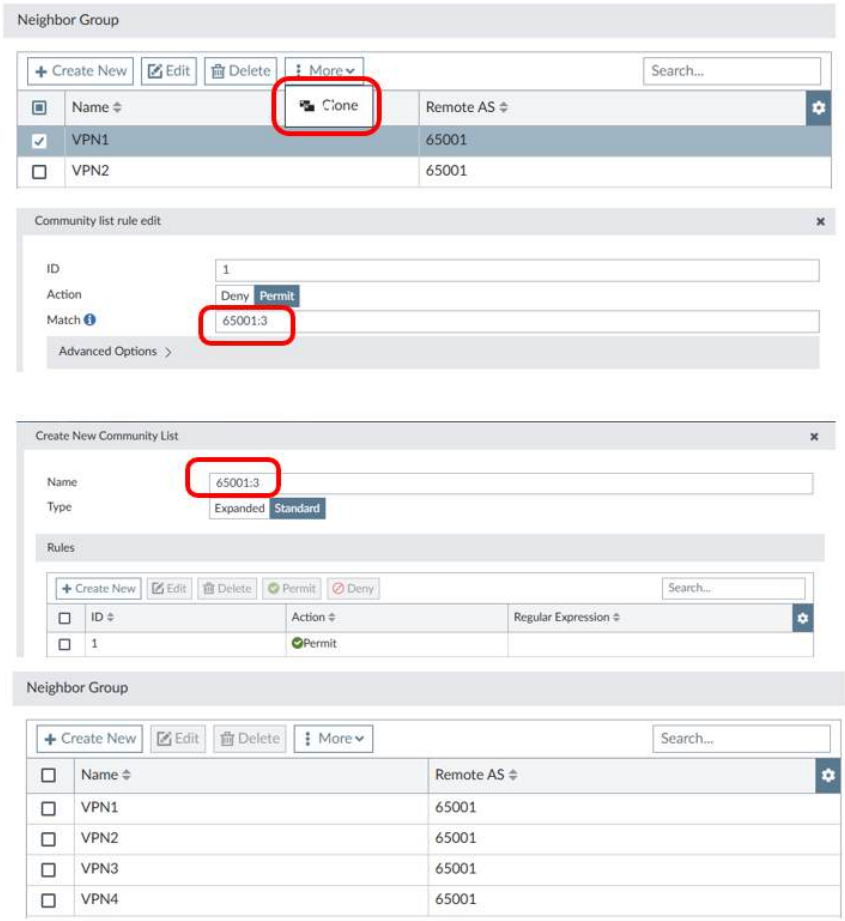
IPSEC HUB1 Status With 4 Tunnels at The Branches

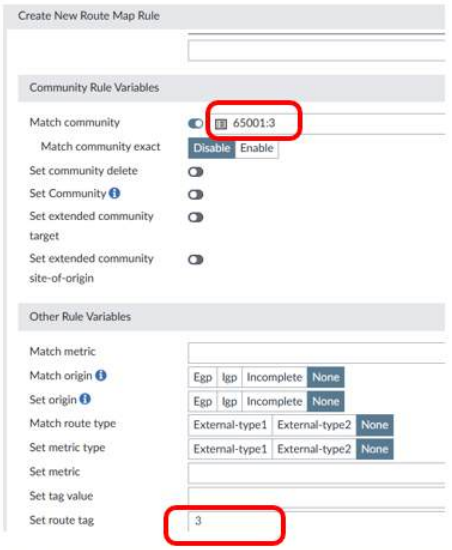
Four tunnels are formed per branch for a total of 8 tunnels. Any additional branch with a similar setting will add 4 tunnels.

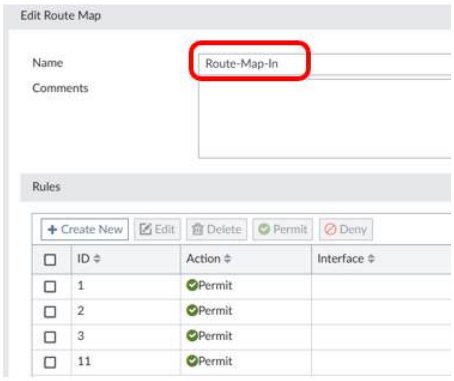

```
HUB1 # diagnose vpn ike gateway list | grep assigned
assigned IPv4 address: 10.50.1.10/255.255.255.0
assigned IPv4 address: 10.50.2.10/255.255.255.0
assigned IPv4 address: 10.50.1.11/255.255.255.0
assigned IPv4 address: 10.50.2.11/255.255.255.0
assigned IPv4 address: 10.50.3.10/255.255.255.0
assigned IPv4 address: 10.50.4.10/255.255.255.0
assigned IPv4 address: 10.50.3.11/255.255.255.0
assigned IPv4 address: 10.50.4.11/255.255.255.0
```

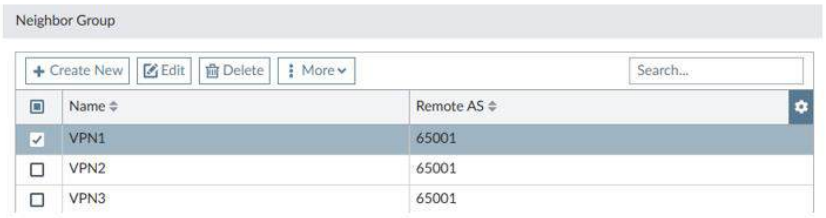
HUB1 VPN3 & VPN4 Neighbor Group BGP Configuration

For BGP, create new neighbor groups for the new tunnel by cloning the existing ones, create new community lists and route maps to match the new tunnels, we start with VPN3 then VPN4

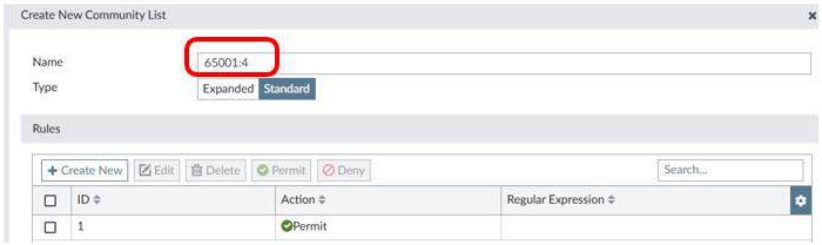


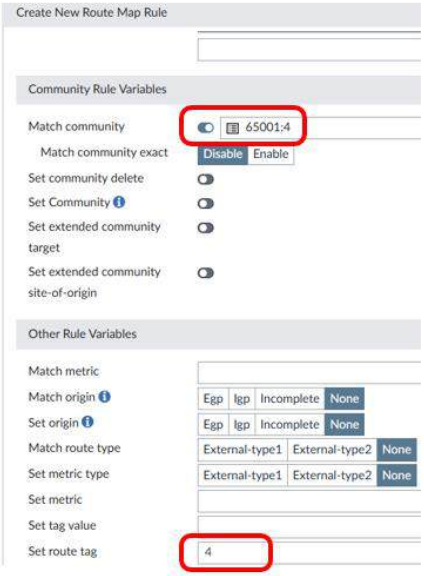


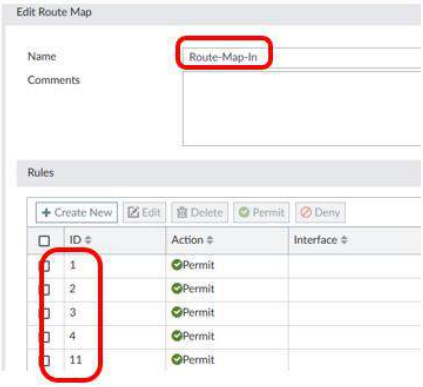



Neighbor Group
Create New
Edit
Delete
More
Search...
Name
Remote AS
VPN1
65001
VPN2
65001
VPN3
65001


Community list rule edit
ID
1
Action
Deny
Permit
Match
65001:4
Advanced Options

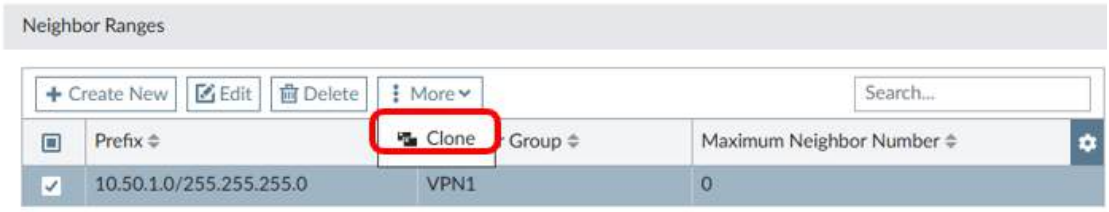

Create New Community List
Name
65001:4
Type
Expanded
Standard
Rules
Create New
Edit
Delete
Permit
Deny
Search...
ID
Action
Regular Expression
1
Permit

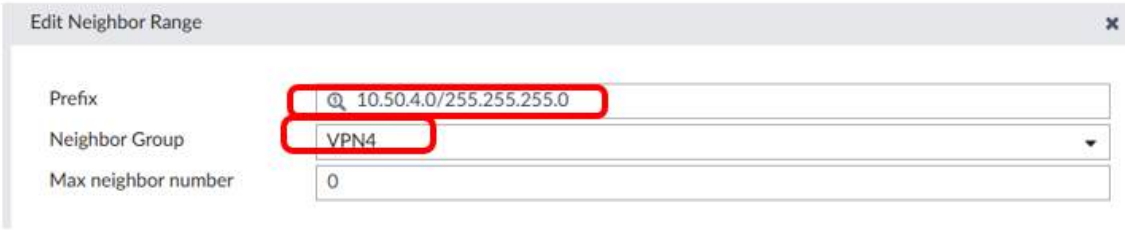

Create New Route Map Rule
Community Rule Variables
Match community
65001:4
Match community exact
Disable
Enable
Set community delete
Set Community
Set extended community target
Set extended community site-of-origin
Other Rule Variables
Match metric
Match origin
Egp
Igp
Incomplete
None
Set origin
Match route type
External-type1
External-type2
Set metric type
Set metric
Set tag value
Set route tag
4

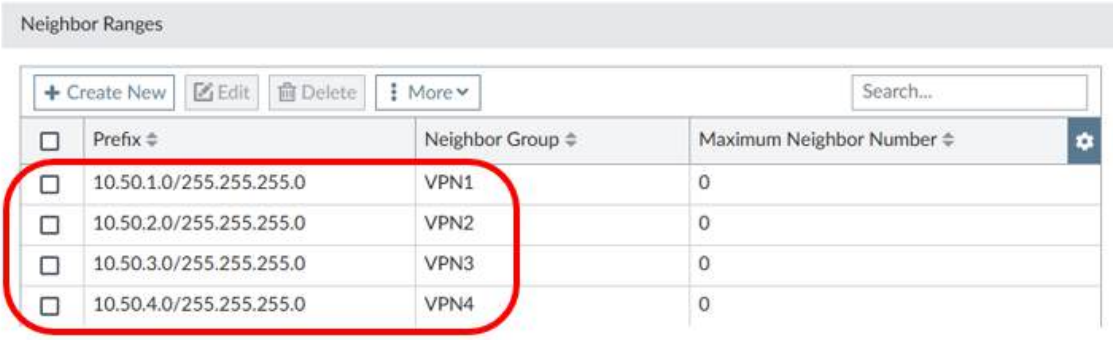

Edit Route Map
Name
Route-Map-In
Comments
Rules
Create New
Edit
Delete
Permit
Deny
ID
Action
Interface
1
Permit
2
Permit
3
Permit
4
Permit
11
Permit

HUB1 VPN3 & VPN4 Neighbor Ranges BGP Configuration

Additional neighbor ranges are created by cloning existing ones, customizing the prefixes and associating with the neighbor groups configured earlier.





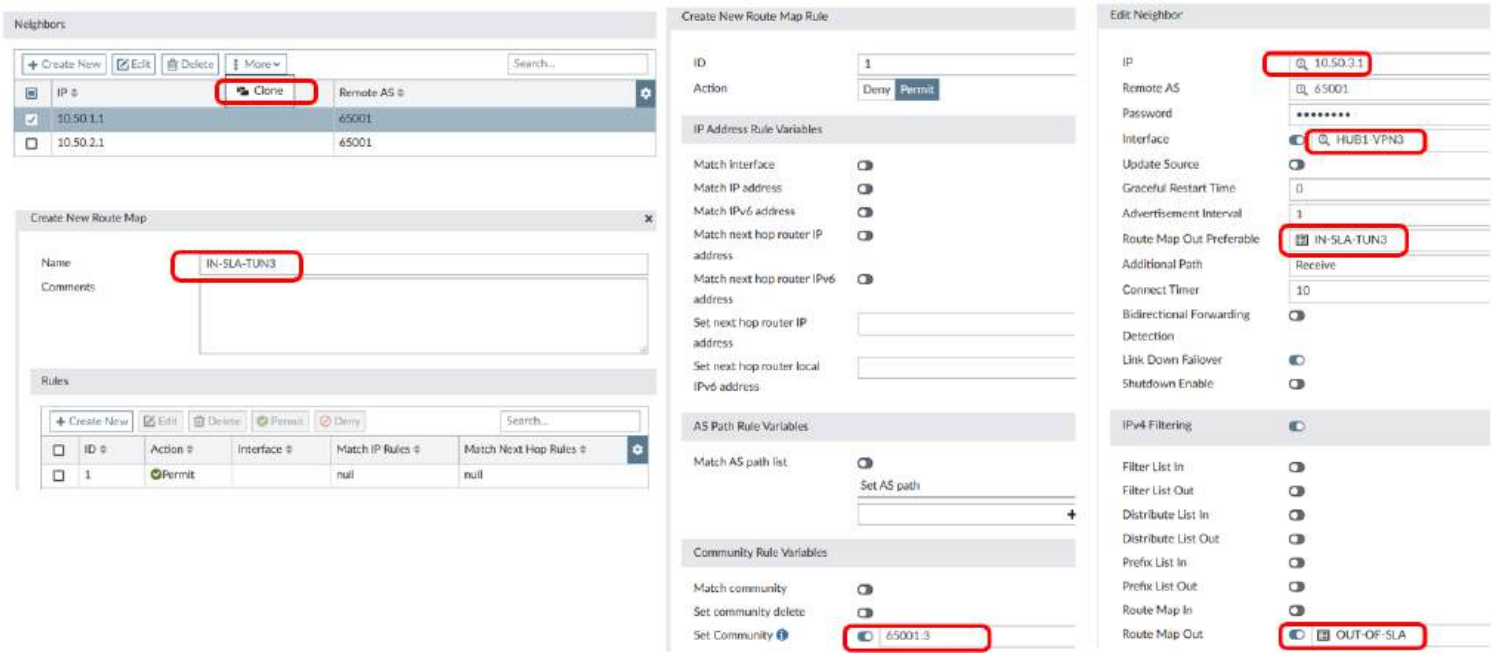


HUB1 BGP Provisioning Preview for VPN3 & VPN4

```
config router community-list
    edit "65001:3"
        config rule
            edit 1
                set action permit
                set match "65001:3"
            next
        end
    next
    edit "65001:4"
        config rule
            edit 1
                set action permit
                set match "65001:4"
            next
        end
    next
end
config router route-map
    edit "Route-Map-In"
        config rule
            edit 3
                set match-community "65001:3"
                set set-route-tag "3"
            next
            edit 4
                set match-community "65001:4"
                set set-route-tag "4"
            next
        end
    next
end
config router bgp
    config neighbor-group
        edit "VPN3"
            set capability-graceful-restart enable
            set link-down-failover enable
            set next-hop-self enable
            set soft-reconfiguration enable
            set remote-as 65001
            set route-map-in "Route-Map-In"
            set additional-path send
            set route-reflector-client enable
        next
        edit "VPN4"
            set capability-graceful-restart enable
            set link-down-failover enable
            set next-hop-self enable
            set soft-reconfiguration enable
            set remote-as 65001
            set route-map-in "Route-Map-In"
            set additional-path send
            set route-reflector-client enable
        next
    end
    config neighbor-range
        edit 3
            set prefix 10.50.3.0 255.255.255.0
            set neighbor-group "VPN3"
        next
        edit 4
            set prefix 10.50.4.0 255.255.255.0
            set neighbor-group "VPN4"
        next
    end
end
```

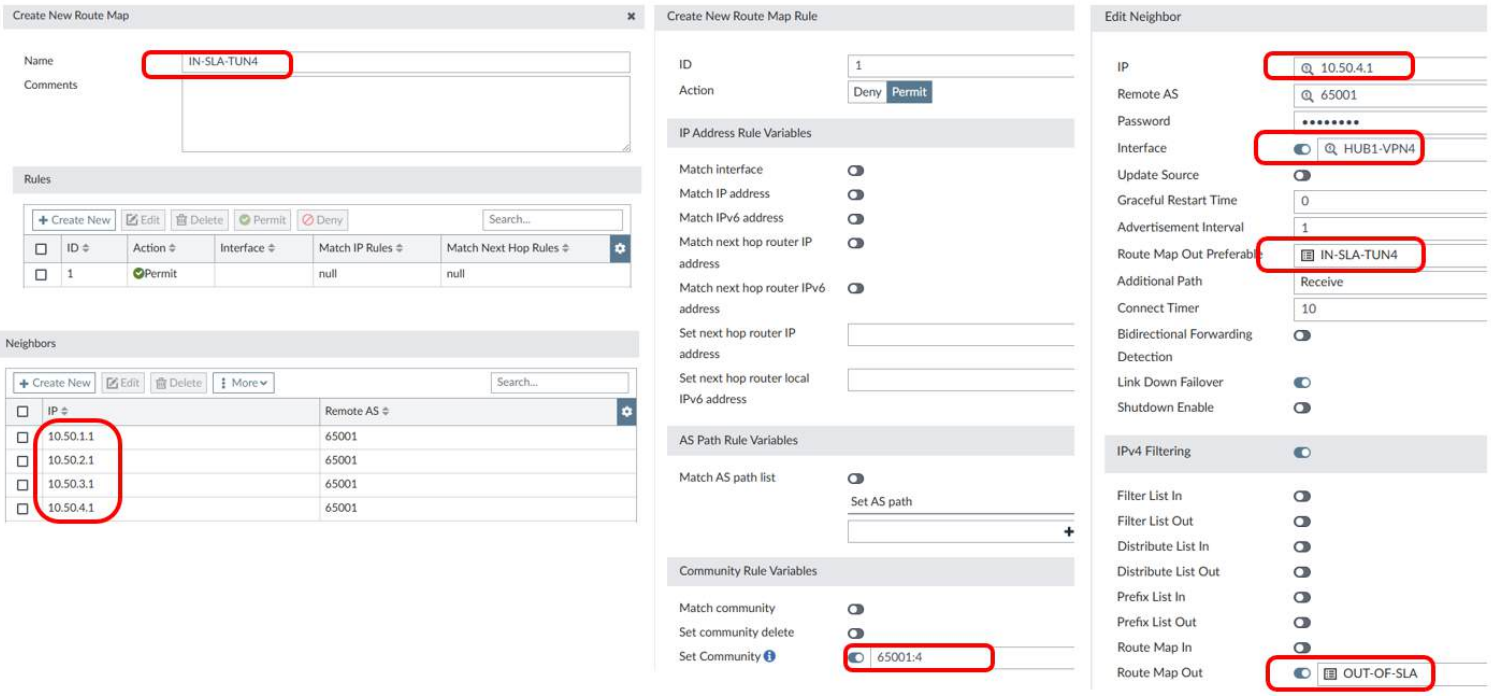
BRANCH Neighbor 3 BGP Configuration

The 3rd neighbor is created by cloning the existing neighbors, changing the ip addresses, interfaces and creating a new route map to set the required community i.e. 65001:3. A new route map out preferable is created for IN-SLA conformity but still use the same route map out for OUT-OF-SLA for all neighbors.



BRANCH Neighbor 4 BGP Configuration

The 4th neighbor is created by cloning the existing neighbors, changing the ip addresses, interfaces and creating a new route map to set the required community i.e. 65001:4. A new route map out preferable is created for IN-SLA conformity but still use the same route map out for OUT-OF-SLA for all neighbors.



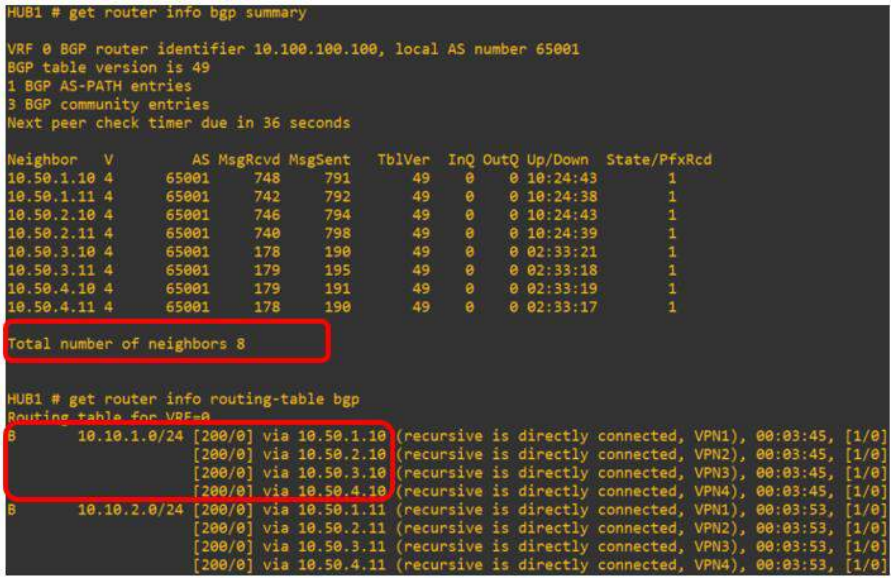
BRANCH PROVISIONING Preview FOR NEIGHBORS 3 & 4

```
config router route-map
    edit "IN-SLA-TUN3"
        config rule
            edit 1
                set set-community "65001:3"
            next
        end
    next
    edit "IN-SLA-TUN4"
        config rule
            edit 1
                set set-community "65001:4"
            next
        end
    next
end
config router bgp
    config neighbor
        edit "10.50.3.1"
            set advertisement-interval 1
            set capability-graceful-restart enable
            set link-down-failover enable
            set soft-reconfiguration enable
            set description "HUB1-VPN1"
            set interface "HUB1-VPN3"
            set remote-as 65001
            set route-map-out "OUT-OF-SLA"
            set connect-timer 10
            set additional-path receive
            set route-map-out-preferable "IN-SLA-TUN3"
        next
        edit "10.50.4.1"
            set advertisement-interval 1
            set capability-graceful-restart enable
            set link-down-failover enable
            set soft-reconfiguration enable
            set description "HUB1-VPN1"
            set interface "HUB1-VPN4"
            set remote-as 65001
            set route-map-out "OUT-OF-SLA"
            set connect-timer 10
            set additional-path receive
            set route-map-out-preferable "IN-SLA-TUN4"
        next
    end
end
```

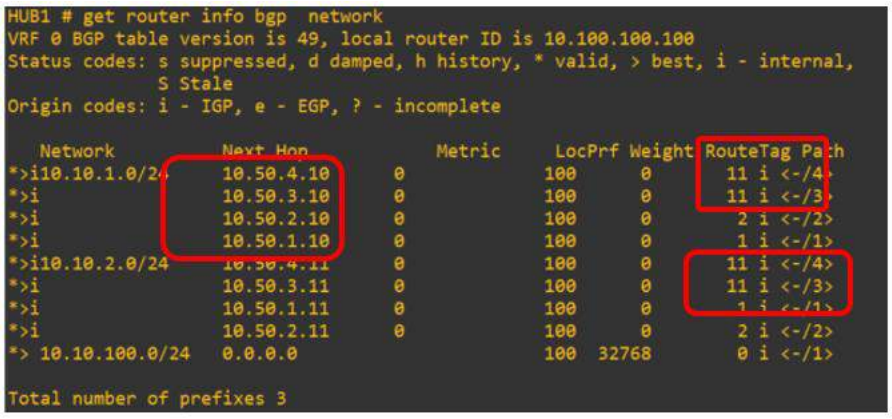
HUB1 BGP Routing verification after 4 Tunnels are Established

Four neighborships are established between HUB1 and Branch1 and Branch 2 over the four tunnels for a total of 8 neighbors for HUB1

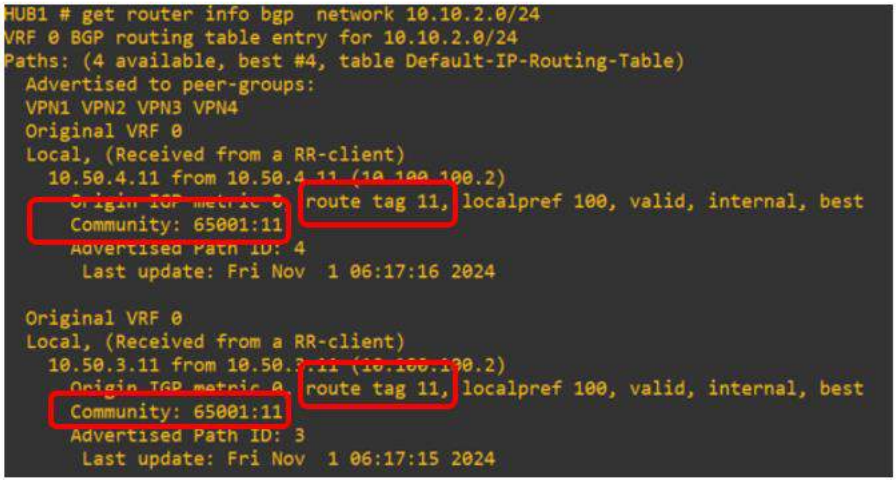
Route Tags are mapped based on the Tunnel/neighbor advertising the network from the branch. For the 3rd and 4th tunnels, the routes are received and placed in the routing table but have a route tag of 11 (OUT-OF-SLA) because the health check and HUB policy for loopback access have not been configured yet.



```
HUB1 # get router info bgp summary

VRF 0 BGP router identifier 10.100.100.100, local AS number 65001
BGP table version is 49
1 BGP AS-PATH entries
3 BGP community entries
Next peer check timer due in 36 seconds

Neighbor   V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.50.1.10 4      65001     748     791       49    0    0 10:24:43        1
10.50.1.11 4      65001     742     792       49    0    0 10:24:38        1
10.50.2.10 4      65001     746     794       49    0    0 10:24:43        1
10.50.2.11 4      65001     740     798       49    0    0 10:24:39        1
10.50.3.10 4      65001     178     190       49    0    0 02:33:21        1
10.50.3.11 4      65001     179     195       49    0    0 02:33:18        1
10.50.4.10 4      65001     179     191       49    0    0 02:33:19        1
10.50.4.11 4      65001     178     190       49    0    0 02:33:17        1

Total number of neighbors 8


HUB1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 10.50.1.10 (recursive is directly connected, VPN1), 00:03:45, [1/0]
                     [200/0] via 10.50.2.10 (recursive is directly connected, VPN2), 00:03:45, [1/0]
                     [200/0] via 10.50.3.10 (recursive is directly connected, VPN3), 00:03:45, [1/0]
                     [200/0] via 10.50.4.10 (recursive is directly connected, VPN4), 00:03:45, [1/0]
B       10.10.2.0/24 [200/0] via 10.50.1.11 (recursive is directly connected, VPN1), 00:03:53, [1/0]
                     [200/0] via 10.50.2.11 (recursive is directly connected, VPN2), 00:03:53, [1/0]
                     [200/0] via 10.50.3.11 (recursive is directly connected, VPN3), 00:03:53, [1/0]
                     [200/0] via 10.50.4.11 (recursive is directly connected, VPN4), 00:03:53, [1/0]
```



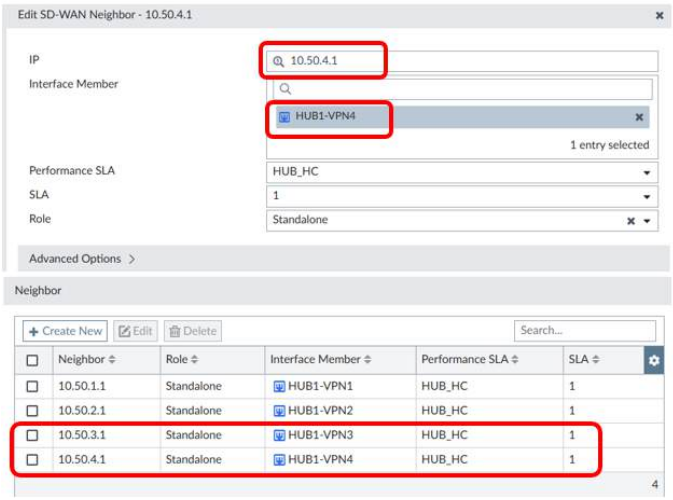
```
HUB1 # get router info bgp  network
VRF 0 BGP table version is 49, local router ID is 10.100.100.100
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.10.1.0/24     10.50.4.10               0          100      0      11 i <-/4>
*>i                 10.50.3.10               0          100      0      11 i <-/3>
*>i                 10.50.2.10               0          100      0       2 i <-/2>
*>i                 10.50.1.10               0          100      0       1 i <-/1>
*>i10.10.2.0/24     10.50.4.11               0          100      0      11 i <-/4>
*>i                 10.50.3.11               0          100      0      11 i <-/3>
*>i                 10.50.1.11               0          100      0       1 i <-/1>
*>i                 10.50.2.11               0          100      0       2 i <-/2>
*> 10.10.100.0/24   0.0.0.0                  0          100  32768       0 i <-/1>

Total number of prefixes 3
```



```
HUB1 # get router info bgp  network 10.10.2.0/24
VRF 0 BGP routing table entry for 10.10.2.0/24
Paths: (4 available, best #4, table Default-IP-Routing-Table)
  Advertised to peer-groups:
  VPN1 VPN2 VPN3 VPN4
  Original VRF 0
  Local, (Received from a RR-client)
    10.50.4.11 from 10.50.4.11 (10.100.100.2)
      Origin IGP metric 0, route tag 11, localpref 100, valid, internal, best
      Community: 65001:11
      Advertised Path ID: 4
       Last update: Fri Nov  1 06:17:16 2024

  Original VRF 0
  Local, (Received from a RR-client)
    10.50.3.11 from 10.50.3.11 (10.100.100.2)
      Origin IGP metric 0, route tag 11, localpref 100, valid, internal, best
      Community: 65001:11
      Advertised Path ID: 3
       Last update: Fri Nov  1 06:17:15 2024
```

Healthcheck Configuration at Branch1 for Neighbor 3&4

These are configurations done within the SDWAN template to associate the BGP neighbors with SDWAN to track link performance metrics for the additional tunnels. With this configured, the new tunnels will have the correct tags i.e. tags 3 & 4.




```
config system sdwan
    config neighbor
        edit "10.50.3.1"
            set health-check "HUB_HC"
            set sla-id 1
            set member 5
        next
        edit "10.50.4.1"
            set health-check "HUB_HC"
            set sla-id 1
            set member 6
        next
    end
end
```

```
HUB1 # get router info bgp  network
VRF 0 BGP table version is 53, local router ID is 10.100.100.100
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.10.1.0/24     10.50.4.10       0             100      0        4 i <-/4>
*>i                 10.50.3.10       0             100      0        3 i <-/3>
*>i                 10.50.2.10       0             100      0        2 i <-/2>
*>i                 10.50.1.10       0             100      0        1 i <-/1>
*>i10.10.2.0/24     10.50.4.11       0             100      0        4 i <-/4>
*>i                 10.50.3.11       0             100      0        3 i <-/3>
*>i                 10.50.1.11       0             100      0        1 i <-/1>
*>i                 10.50.2.11       0             100      0        2 i <-/2>
*> 10.10.100.0/24   0.0.0.0                        100  32768       0 i <-/1>

Total number of prefixes 3
```

Branch 1 & 2 BGP Routing Verification after 4 Tunnels are Established

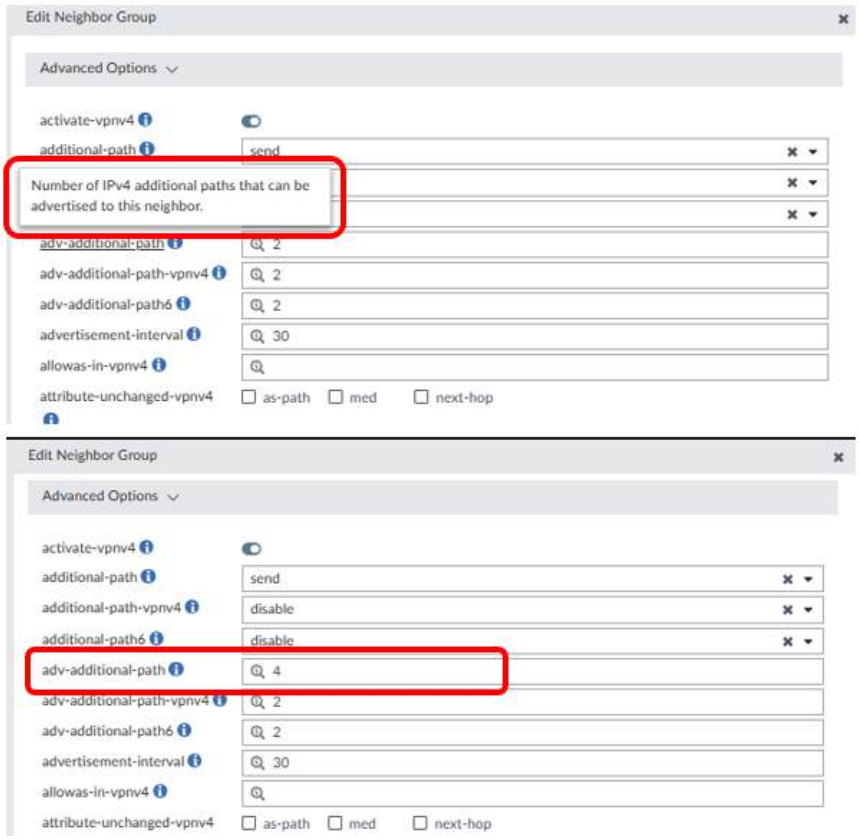
Four neighborships are established between HUB1 and Branch1& Branch2 over the four tunnels with the upstream tunnel static IP addresses as the neighbor. Four redundant routes are established for each network learnt from HUB1.

The setup only has 2 networks learnt from remote spokes; this is because of the default setting on the BGP template that limits the HUBs to advertise two best routes per network for learnt routes.

```
BR1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.2.0/24 [200/0] via 10.50.1.11 [4] (recursive is directly connected, HUB1-VPN1), 00:02:50, [1/0]
                     [200/0] via 10.50.2.11 [4] (recursive is directly connected, HUB1-VPN2), 00:02:50, [1/0]
B       10.10.100.0/24 [200/0] via 10.50.1.1 (recursive via HUB1-VPN1 tunnel 100.1.100.1), 02:57:37, [1/0]
                       [200/0] via 10.50.2.1 (recursive via HUB1-VPN2 tunnel 200.2.100.1), 02:57:37, [1/0]
                       [200/0] via 10.50.3.1 (recursive via HUB1-VPN3 tunnel 10.0.0.1), 02:57:37, [1/0]
                       [200/0] via 10.50.4.1 (recursive via HUB1-VPN4 tunnel 10.0.0.2), 02:57:37, [1/0]


BR1 # get router info bgp summary

VRF 0 BGP router identifier 10.100.100.1, local AS number 65001
BGP table version is 25
1 BGP AS-PATH entries
4 BGP community entries

Neighbor  V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.50.1.1 4      65001     827     778       24    0    0 10:49:40        3
10.50.2.1 4      65001     828     777       24    0    0 10:49:40        3
10.50.3.1 4      65001     227     208       24    0    0 02:58:18        3
10.50.4.1 4      65001     225     208       24    0    0 02:58:16        3

Total number of neighbors 4
```

```
BR2 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 10.50.1.10 [4] (recursive is directly connected, HUB1-VPN1), 00:15:39, [1/0]
                     [200/0] via 10.50.2.10 [4] (recursive is directly connected, HUB1-VPN2), 00:15:39, [1/0]
B       10.10.100.0/24 [200/0] via 10.50.1.1 (recursive via HUB1-VPN1 tunnel 100.1.100.1), 02:58:38, [1/0]
                       [200/0] via 10.50.2.1 (recursive via HUB1-VPN2 tunnel 200.2.100.1), 02:58:38, [1/0]
                       [200/0] via 10.50.3.1 (recursive via HUB1-VPN3 tunnel 10.0.0.1), 02:58:38, [1/0]
                       [200/0] via 10.50.4.1 (recursive via HUB1-VPN4 tunnel 10.0.0.2), 02:58:38, [1/0]


BR2 # get router info bgp summary

VRF 0 BGP router identifier 10.100.100.2, local AS number 65001
BGP table version is 22
1 BGP AS-PATH entries
4 BGP community entries

Neighbor  V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.50.1.1 4      65001     828     773       22    0    0 10:50:33        3
10.50.2.1 4      65001     832     771       22    0    0 10:50:34        3
10.50.3.1 4      65001     231     208       22    0    0 02:59:13        3
10.50.4.1 4      65001     227     208       22    0    0 02:59:12        3

Total number of neighbors 4
```

To resolve this and receive all available paths, each neighbor group at the HUB is configured to advertise 4 paths downstream to each spoke.




```
config router bgp
    config neighbor-group
        edit "VPN1"
            set adv-additional-path 4
        next
        edit "VPN2"
            set adv-additional-path 4
        next
        edit "VPN3"
            set adv-additional-path 4
        next
        edit "VPN4"
            set adv-additional-path 4
        next
    end
end
```

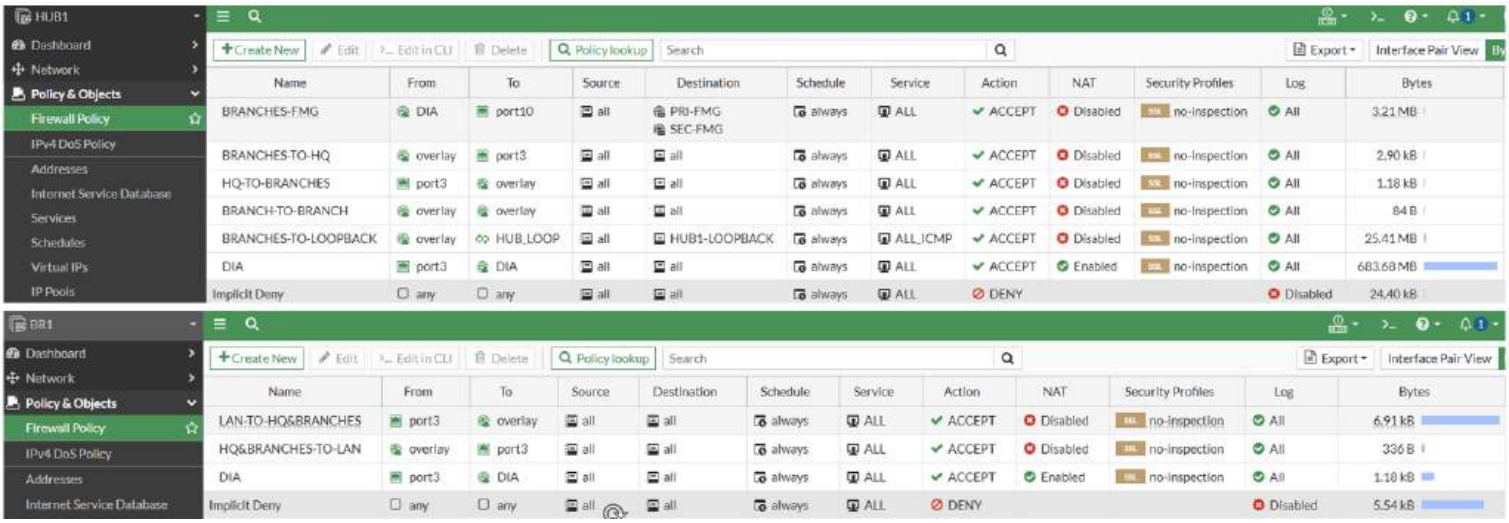
With the additional paths raised to 4, the spokes/branches now have 4 paths to each other over the available tunnels.

```
BR1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.2.0/24 [200/0] via 10.50.1.10 (recursive is directly connected, HUB1-VPN1), 00:00:02, [1/0]
                     [200/0] via 10.50.2.10 (recursive is directly connected, HUB1-VPN2), 00:00:02, [1/0]
                     [200/0] via 10.50.3.10 (recursive is directly connected, HUB1-VPN3), 00:00:02, [1/0]
                     [200/0] via 10.50.4.10 (recursive is directly connected, HUB1-VPN4), 00:00:02, [1/0]
B       10.10.100.0/24 [200/0] via 10.50.1.1 (recursive via HUB1-VPN1 tunnel 100.1.100.1), 00:00:22, [1/0]
                       [200/0] via 10.50.2.1 (recursive via HUB1-VPN2 tunnel 200.2.100.1), 00:00:22, [1/0]
                       [200/0] via 10.50.3.1 (recursive via HUB1-VPN3 tunnel 10.0.0.1), 00:00:22, [1/0]
                       [200/0] via 10.50.4.1 (recursive via HUB1-VPN4 tunnel 10.0.0.2), 00:00:22, [1/0]
```

```
BR2 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 10.50.1.11 [4] (recursive is directly connected, HUB1-VPN1), 00:00:18, [1/0]
                     [200/0] via 10.50.2.11 [4] (recursive is directly connected, HUB1-VPN2), 00:00:18, [1/0]
                     [200/0] via 10.50.3.11 [4] (recursive is directly connected, HUB1-VPN3), 00:00:18, [1/0]
                     [200/0] via 10.50.4.11 [4] (recursive is directly connected, HUB1-VPN4), 00:00:18, [1/0]
B       10.10.100.0/24 [200/0] via 10.50.1.1 (recursive via HUB1-VPN1 tunnel 100.1.100.1), 00:00:18, [1/0]
                       [200/0] via 10.50.2.1 (recursive via HUB1-VPN2 tunnel 200.2.100.1), 00:00:18, [1/0]
                       [200/0] via 10.50.3.1 (recursive via HUB1-VPN3 tunnel 10.0.0.1), 00:00:18, [1/0]
                       [200/0] via 10.50.4.1 (recursive via HUB1-VPN4 tunnel 10.0.0.2), 00:00:18, [1/0]
```

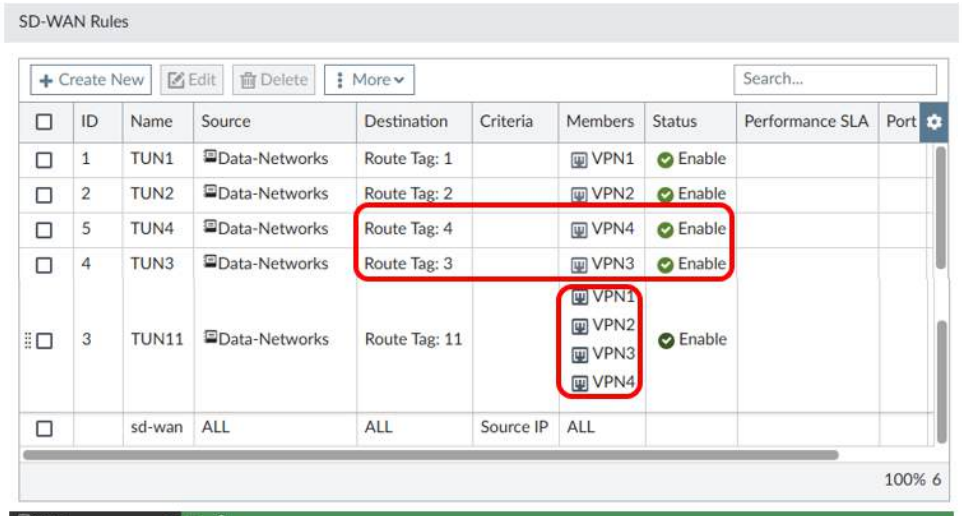
Firewall policy consideration with 4 Tunnels

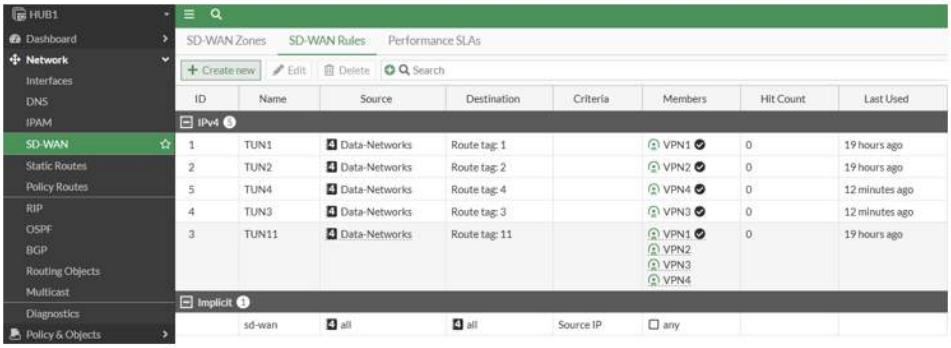
For the firewall policies at HUB1 and Branches, no change is required since the referenced overlay objects will be updated to include the additional tunnels (3&4) when device settings are updated under sdwan.



HUB SDWAN Rules with 4 tunnels.

Additional sdwan rules to reference the additional tags are created and added as member interfaces for the OUT of SLA state with route tag 11.



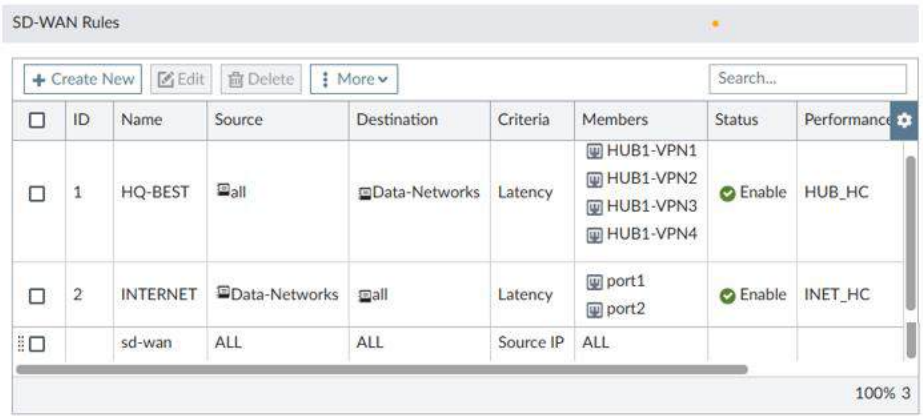



```
config system sdwan
    config service
        edit 5
            set name "TUN4"
            set route-tag 4
            set src "Data-Networks"
            set priority-members 6
        next
        edit 4
            set name "TUN3"
            set route-tag 3
            set src "Data-Networks"
            set priority-members 5
        next
        edit 3
            set priority-members 3 4 5 6
        next
        move 3 after 4
    end
end
```

BRANCH SDWAN Rules with 4 tunnels

Normal sdwan rules are configured at the branches. For internet access using the underlay DIA, we define a new healthcheck to 8.8.8.8 and use the HUB_HC to track links to HQ.

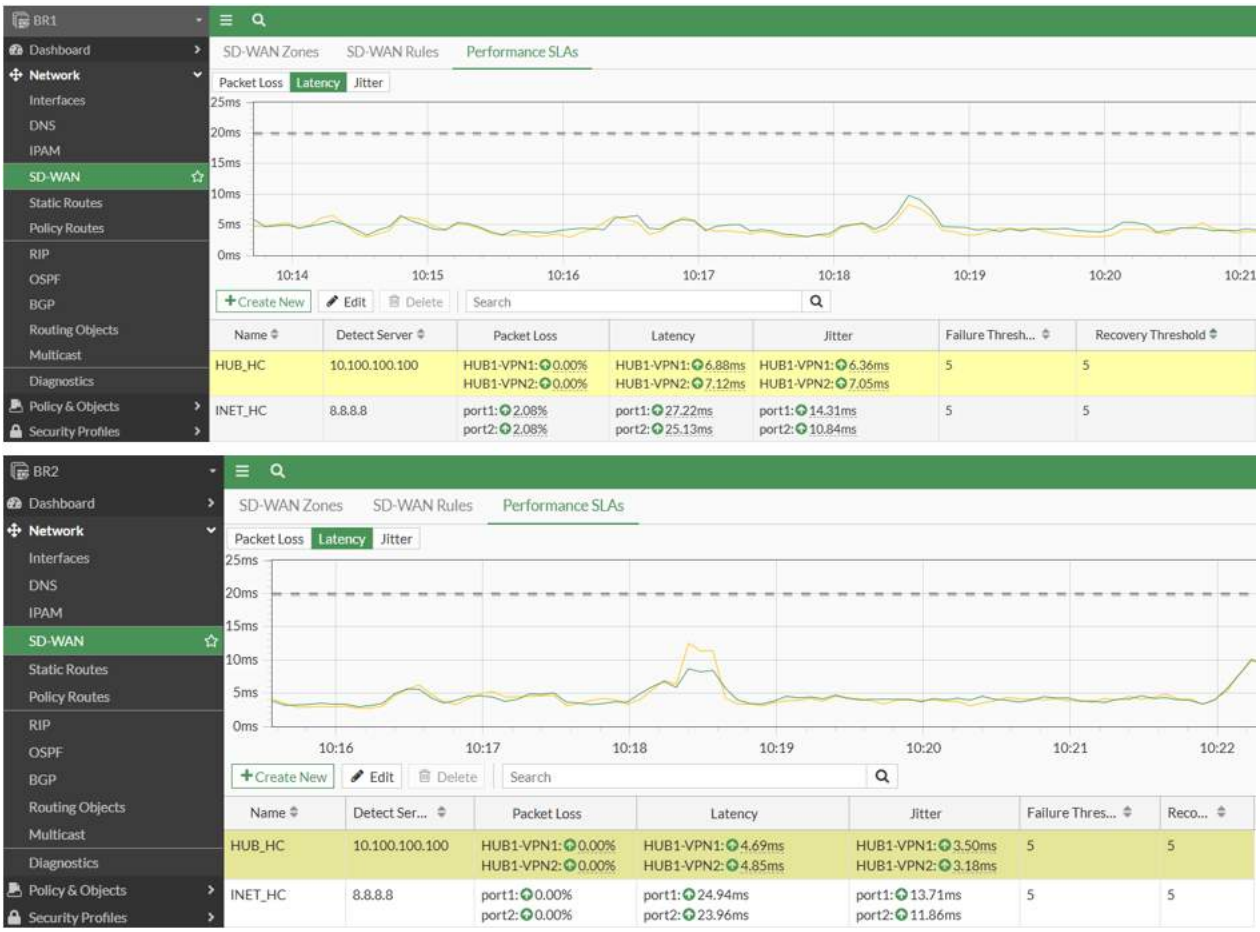
Interface membership is restricted, i.e. Internet access to use DIA (port1 and port2) while the overlay tunnels are used access to the HUB networks.




```
config system sdwan
    config health-check
        edit "INET_HC"
            set server 8.8.8.8
            set members 4 3
            config sla
                edit 1
                    set latency-threshold 50
                    set jitter-threshold 50
                    set packetloss-threshold 5
                next
            end
        next
    end
    config service
        edit 1
            set name "HQ-BEST"
            set mode priority
            set dst "Data-Networks"
            set src "all"
            set health-check "HUB_HC"
            set priority-members 1 2 5 6
        next
        edit 2
            set name "INTERNET"
            set mode priority
            set dst "all"
            set src "Data-Networks"
            set health-check "INET_HC"
            set priority-members 3 4
        next
    end
end
```

BRANCH SDWAN Rules with 4 tunnels Verification

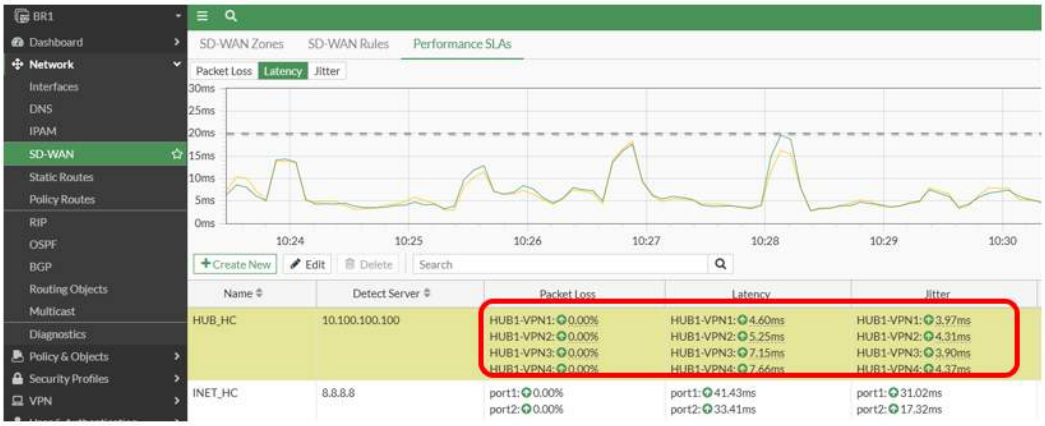
From the tracking, we observe that HUB1-VPN3 and HUB1-VPN4 are not used in tracking the loopback at the HUB, we will add this next.



After associating the healthcheck with all overlay interface, we now track over all available paths.

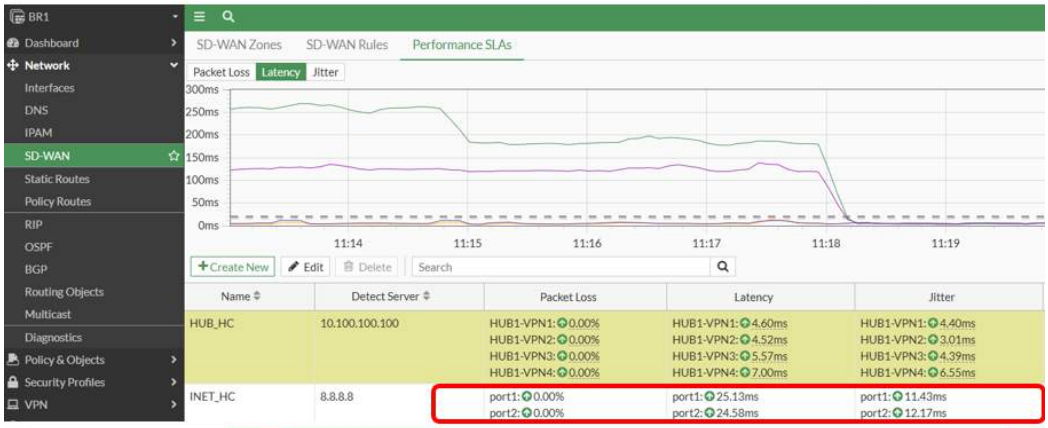

```
config system sdwan
    config health-check
        edit "HUB_HC"
            set members 6 5 1 2
        next
    end
end
```

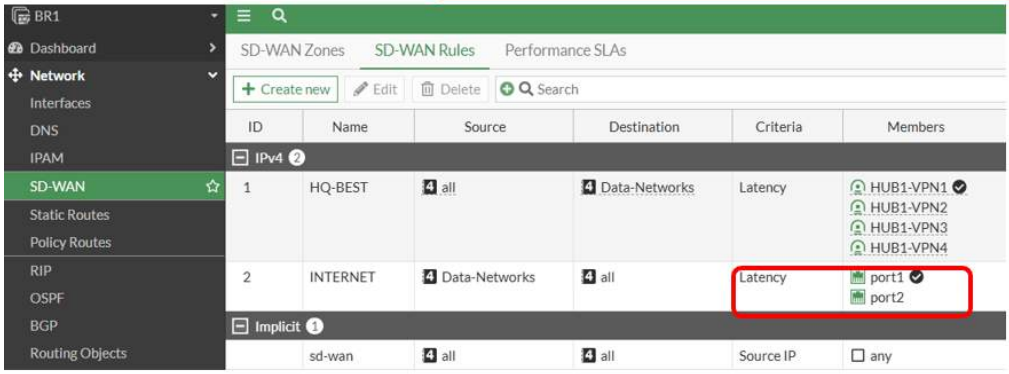
All overlay interfaces are now used with the SLA.



LINK Performance Fluctuation

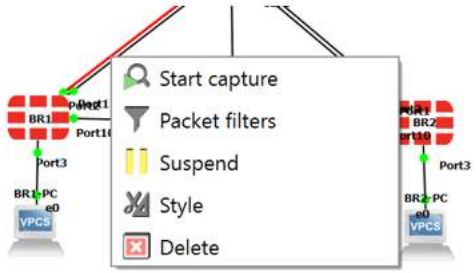
Based on the diagram below, when both links are available, port1 is selected because it has the best quality compared to port2.

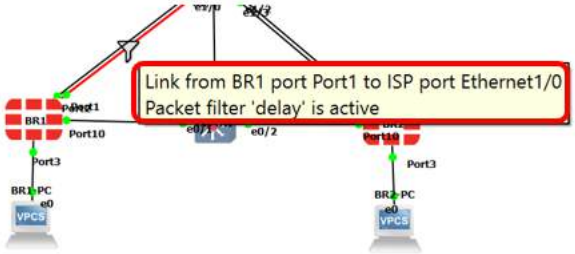


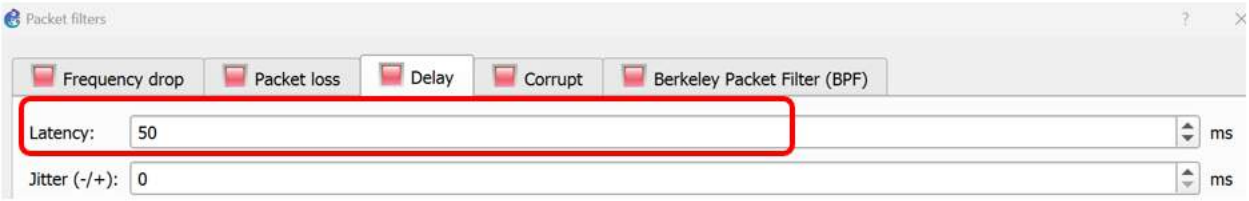


GNS3 packet filters allow manipulation of link quality.

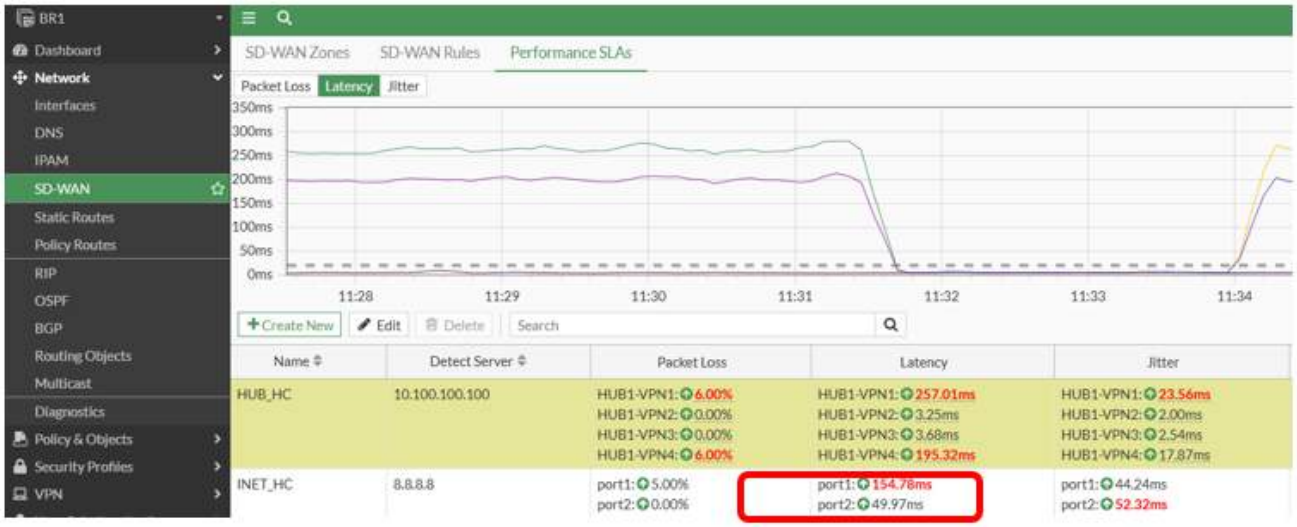
For this demo, the latency of the preferred link is raised to make the secondary link preferred i.e. port2 now has less latency and will be the preferred path.

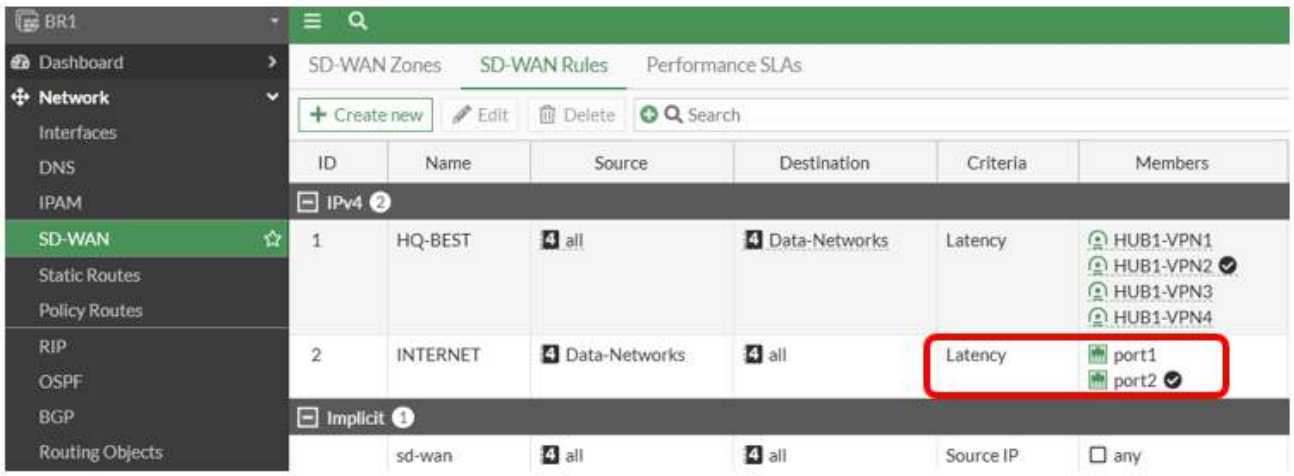






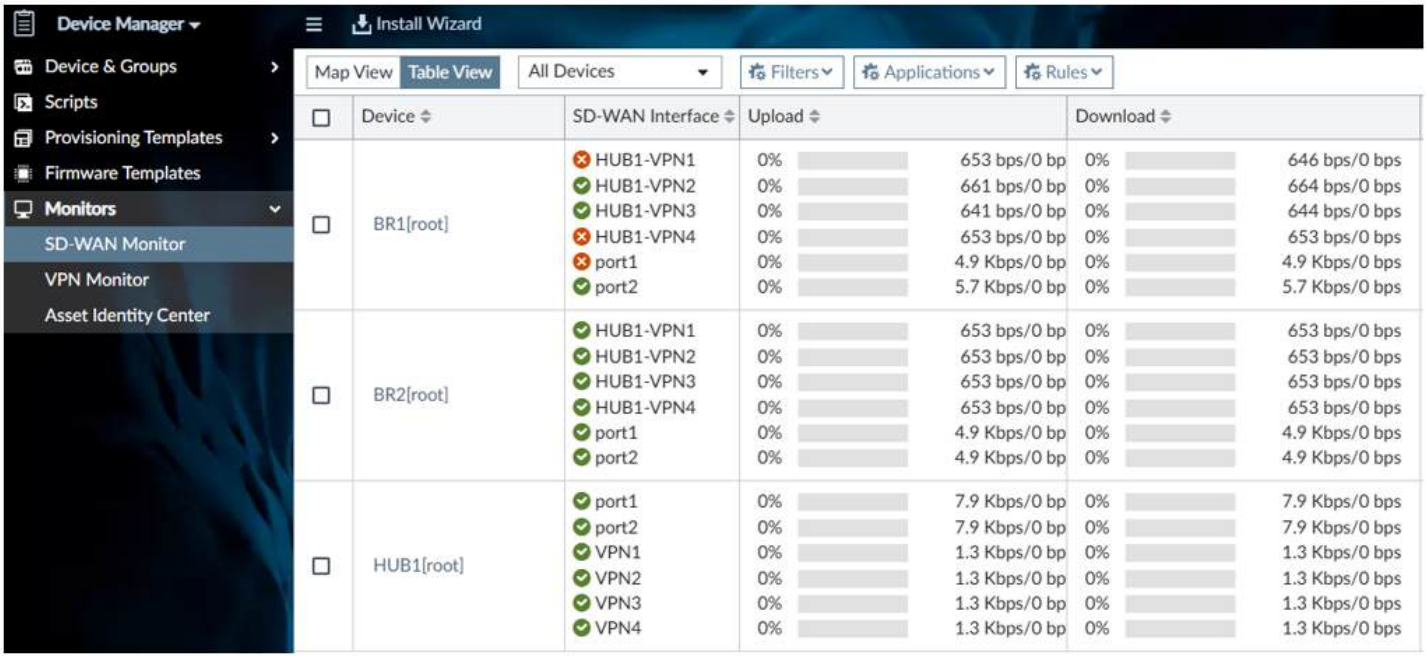
Port2 is selected as the best path since it has less latency compared to port1.

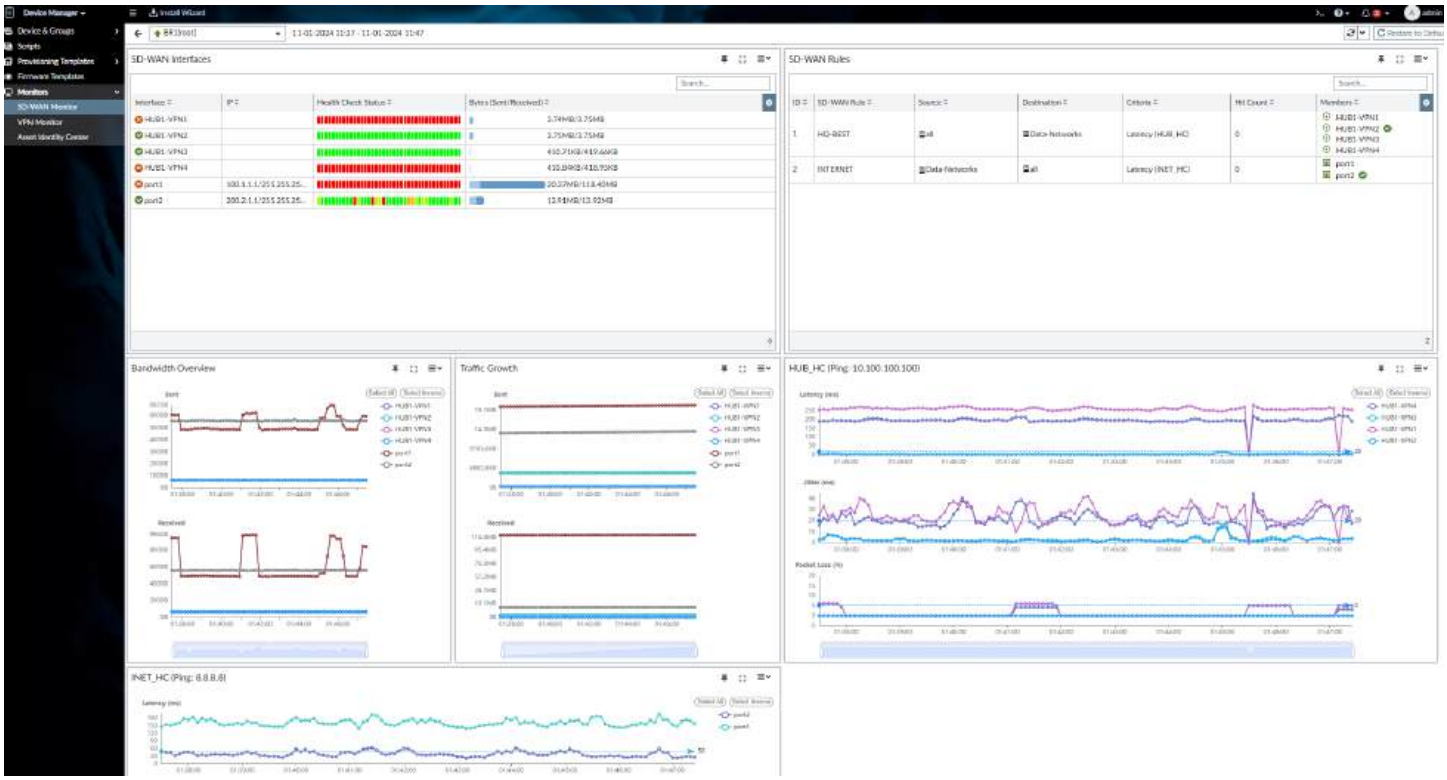




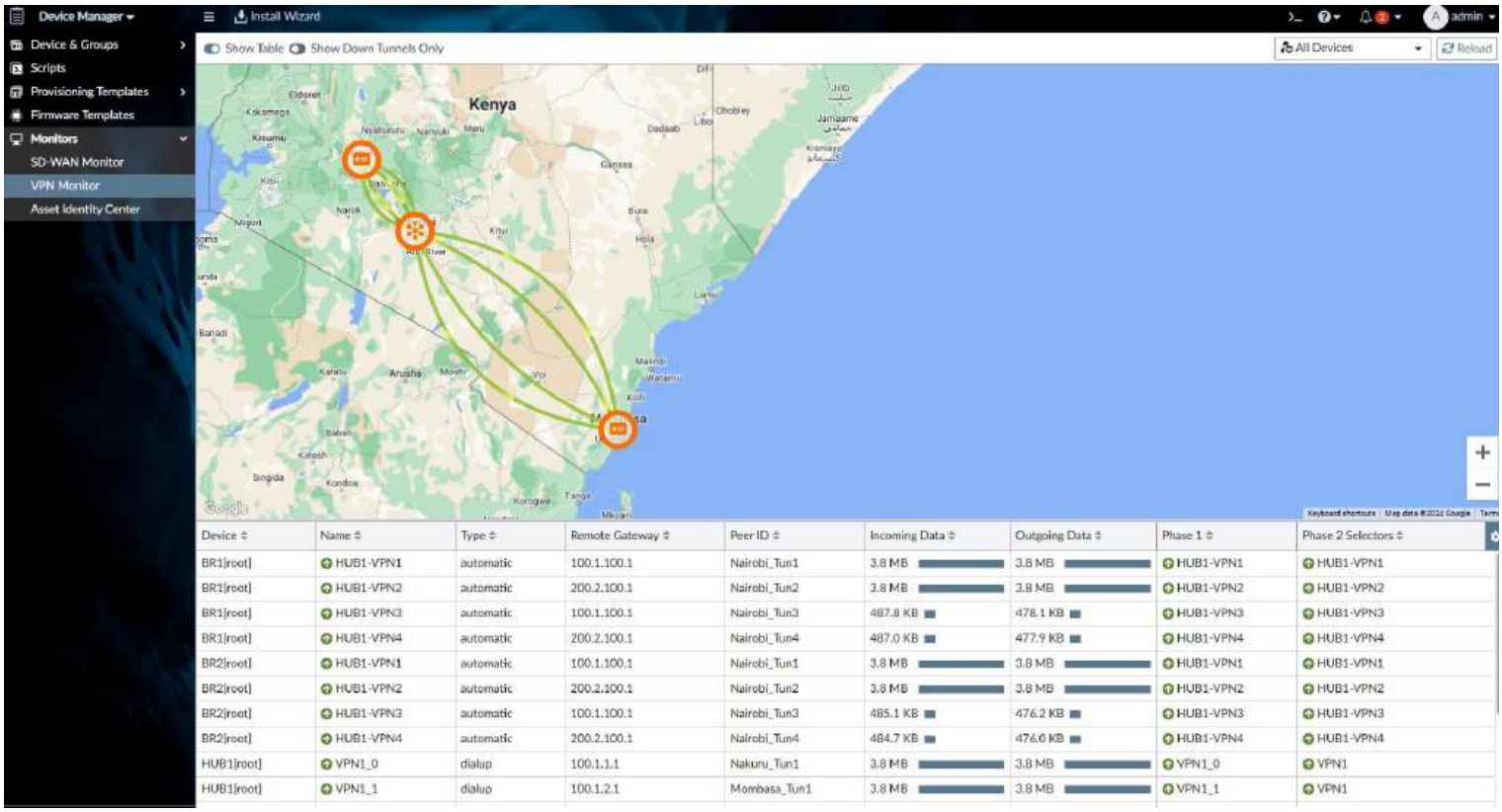
SDWAN MONITORING From FortiManager

FortiManager offers monitoring options to have visibility of all available tunnels





To have visibility to all tunnels and get a map view, edit each device and assign the accurate GPS coordinates.



Device	Name	Type	Remote Gateway	Peer ID	Incoming Data	Outgoing Data	Phase 1	Phase 2 Selectors
BR1[root]	HUB1-VPN1	automatic	100.1.100.1	Nairobi_Tun1	3.8 MB	3.8 MB	HUB1-VPN1	HUB1-VPN1
BR1[root]	HUB1-VPN2	automatic	200.2.100.1	Nairobi_Tun2	3.8 MB	3.8 MB	HUB1-VPN2	HUB1-VPN2
BR1[root]	HUB1-VPN3	automatic	100.1.100.1	Nairobi_Tun3	487.8 KB	478.1 KB	HUB1-VPN3	HUB1-VPN3
BR1[root]	HUB1-VPN4	automatic	200.2.100.1	Nairobi_Tun4	487.0 KB	477.9 KB	HUB1-VPN4	HUB1-VPN4
BR2[root]	HUB1-VPN1	automatic	100.1.100.1	Nairobi_Tun1	3.8 MB	3.8 MB	HUB1-VPN1	HUB1-VPN1
BR2[root]	HUB1-VPN2	automatic	200.2.100.1	Nairobi_Tun2	3.8 MB	3.8 MB	HUB1-VPN2	HUB1-VPN2
BR2[root]	HUB1-VPN3	automatic	100.1.100.1	Nairobi_Tun3	485.1 KB	476.2 KB	HUB1-VPN3	HUB1-VPN3
BR2[root]	HUB1-VPN4	automatic	200.2.100.1	Nairobi_Tun4	484.7 KB	476.0 KB	HUB1-VPN4	HUB1-VPN4
HUB1[root]	VPN1_0	dialup	100.1.1.1	Nakuru_Tun1	3.8 MB	3.8 MB	VPN1_0	VPN1
HUB1[root]	VPN1_1	dialup	100.1.2.1	Mombasa_Tun1	3.8 MB	3.8 MB	VPN1_1	VPN1

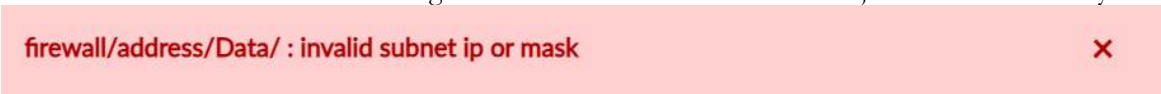
1. How many FortiManager instances can a FortiGate unit reference at the same time
   a) 2
   b) 5
   c) 10

2. Which configuration should be sent to the FortiGate units first during the initial design phase
   a) Device settings
   b) Policy package
   c) Both at the same time

3. After configuring IPsec tunnels at the HUB and branches, it was observed that the tunnels were still down. Authentication and all required configurations were noted to be in order. What is required to bring them up?
   a) Configure interfaces into SDWAN zones at the branch and HUB
   b) Configure outbound overlay policies at the branch and inbound policies at the HUB.
   c) Any of the above

4. When pushing firewall policies to the branch to support sdwan zones for Internet access and Overlay tunnels, network access was lost and FortiManager could no longer access the branch. What is the likely cause for this?
   a) Missing firewall policies at the branch.
   b) Network outage at the ISP level.
   c) Underlay interfaces configured with missing gateways

5. The error below was shown when using a Metadata variable in an address object. What is the likely cause



   a) Duplicate Metadata entries
   b) Invalid IP address.
   c) Missing default value in the Metadata Variable.

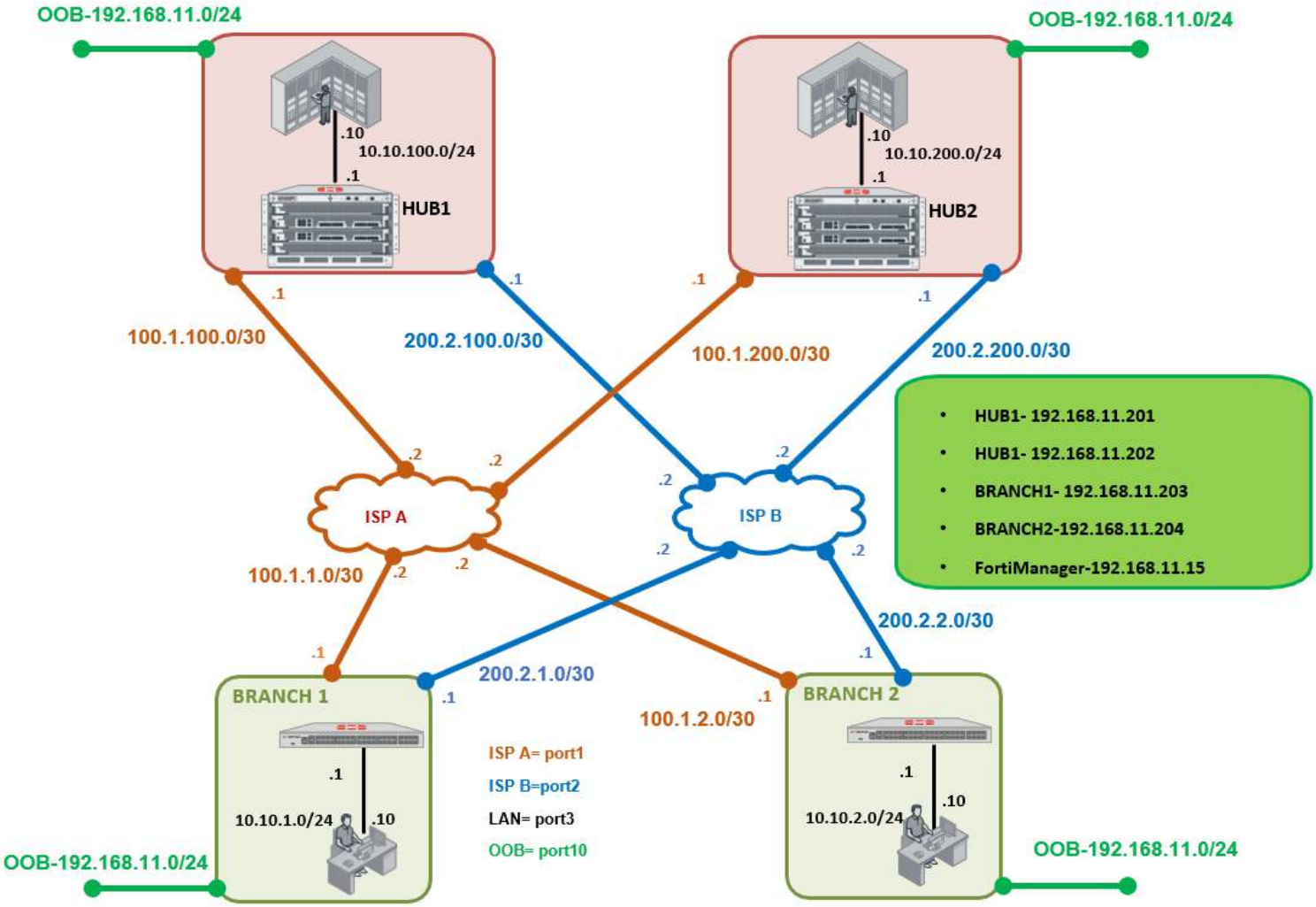
6. A route map is configured to define out of SLA state during high latency in the network. Which section should the route map be applied at the branch:
   a) Route map out
   b) Route map out preferable
   c) Route map in

7. Why is a loopback interface preferred for tracking at the HUB and not a server in the DC.
   a) It is more stable with guaranteed uptime
   b) It is not subject to interference from HQ LAN delays
   c) It is easier to advertise via IBGP
   d) All the above

Chapter 4: FortiManager Dual Hub with Dual ISP Underlay Deployment

This is a common deployment where enterprises have a disaster recovery site on-premise or cloud. The branches connect to both hubs to access hosted services. An added benefit of this setup is the additional paths created between branches, if four tunnels are setup to each HUB a total of 8 paths are formed between branches.

The topology for this design is shown below:

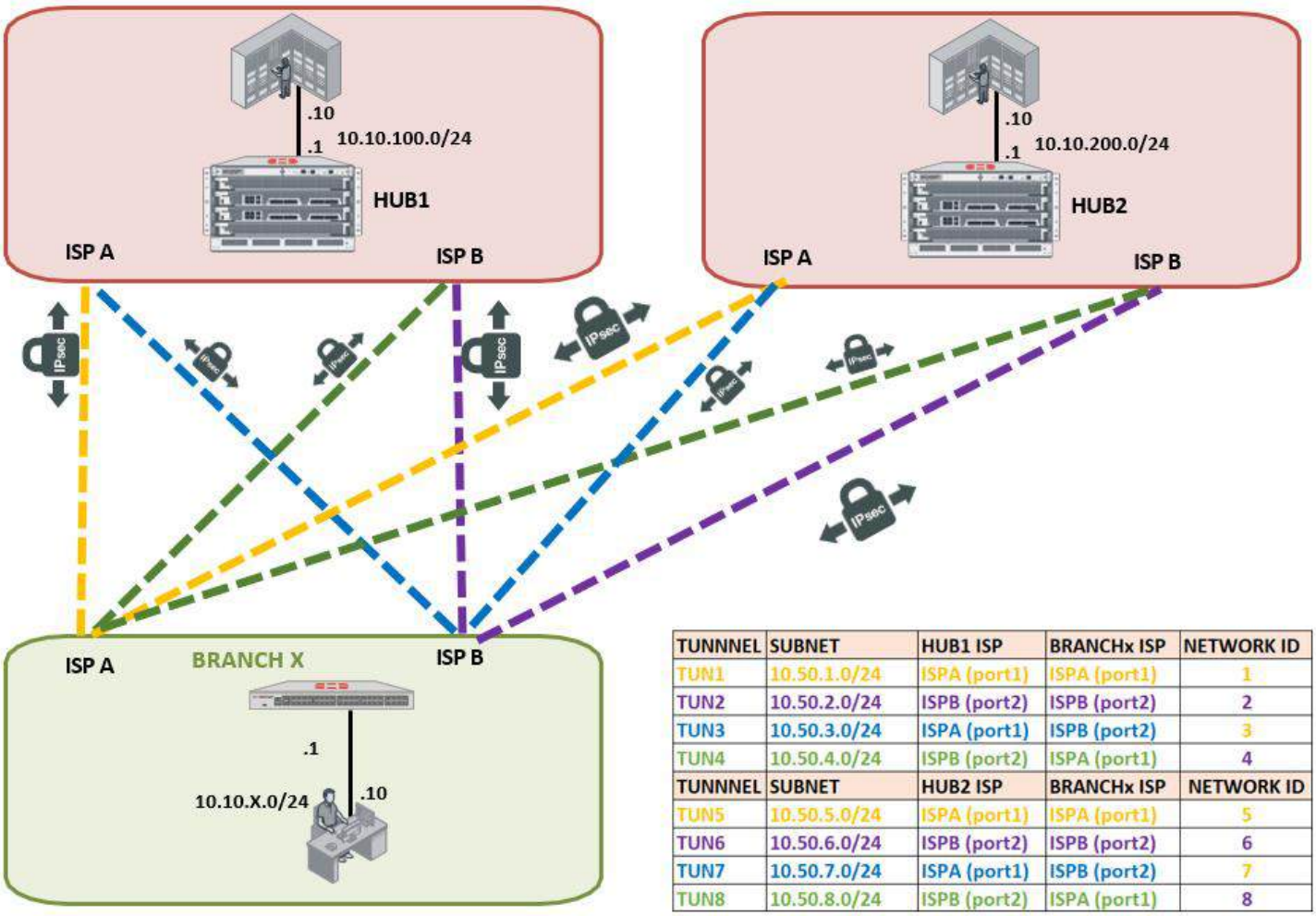


IPSEC Redundancy Between a Branch and 2 HUBS

The setup has additional redundant connections to HUB2 for a total of 8 tunnels per branch. It should be noted that SDWAN is responsible for routing and optimized link selection, whether an application willbe active in HUB1 or HUB2 is determined by other factors e.g. DNS.

Since unique **network IDs** and subnets are required per tunnel, Tunnel assignment is extended from 5-8 as the unique identifiers. It is not mandatory to configure 4 tunnels per HUB, if link resilience is guaranteed, 2 tunnels could be configured per HUB for a total of 4 per branch.

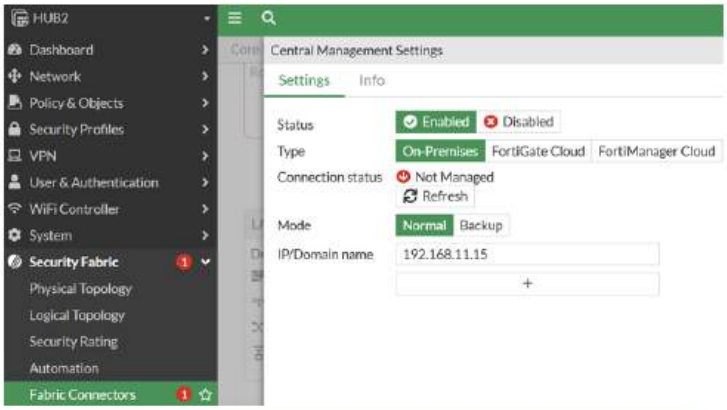
Cross-connect link redundancy is not possible in some scenario e.g. in MPLS networks between different ISPs without route reachability for underlays.

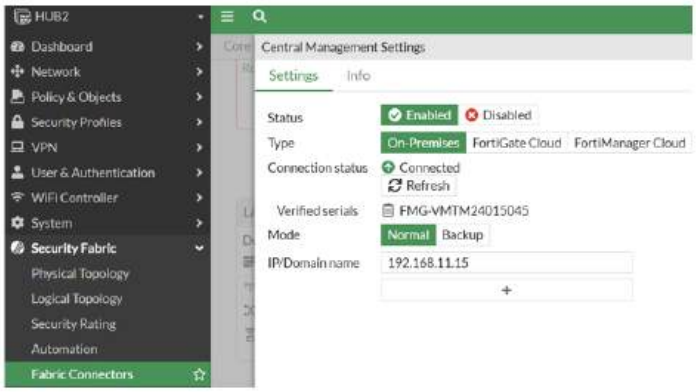


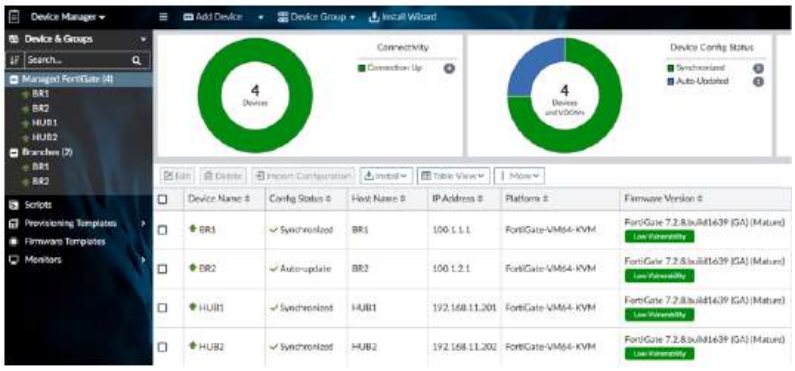
TUNNNEL	SUBNET	HUB1 ISP	BRANCHx ISP	NETWORK ID
TUN1	10.50.1.0/24	ISPA (port1)	ISPA (port1)	1
TUN2	10.50.2.0/24	ISPB (port2)	ISPB (port2)	2
TUN3	10.50.3.0/24	ISPA (port1)	ISPB (port2)	3
TUN4	10.50.4.0/24	ISPB (port2)	ISPA (port1)	4
TUNNNEL	SUBNET	HUB2 ISP	BRANCHx ISP	NETWORK ID
TUN5	10.50.5.0/24	ISPA (port1)	ISPA (port1)	5
TUN6	10.50.6.0/24	ISPB (port2)	ISPB (port2)	6
TUN7	10.50.7.0/24	ISPA (port1)	ISPB (port2)	7
TUN8	10.50.8.0/24	ISPB (port2)	ISPA (port1)	8

Register HUB2 To FortiManager

HUB2 will use OOB network for centralized management . After authorization, setup the unit sequentially starting with IPsec.

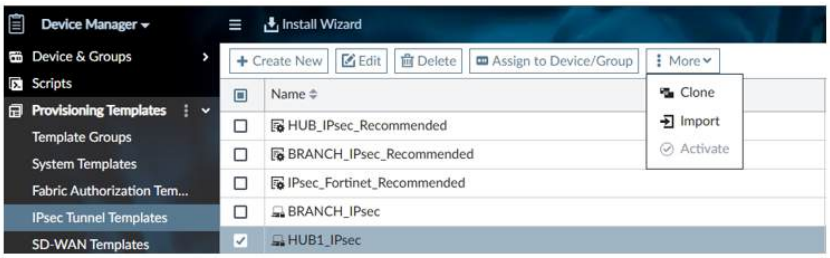




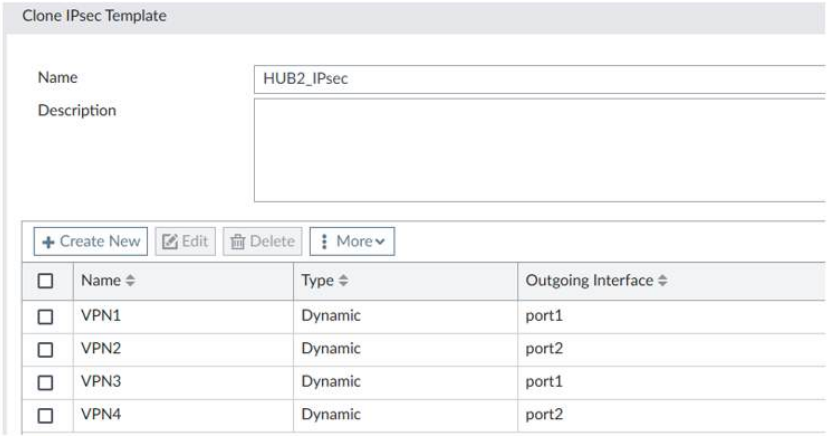


IPSEC HUB2 Configuration

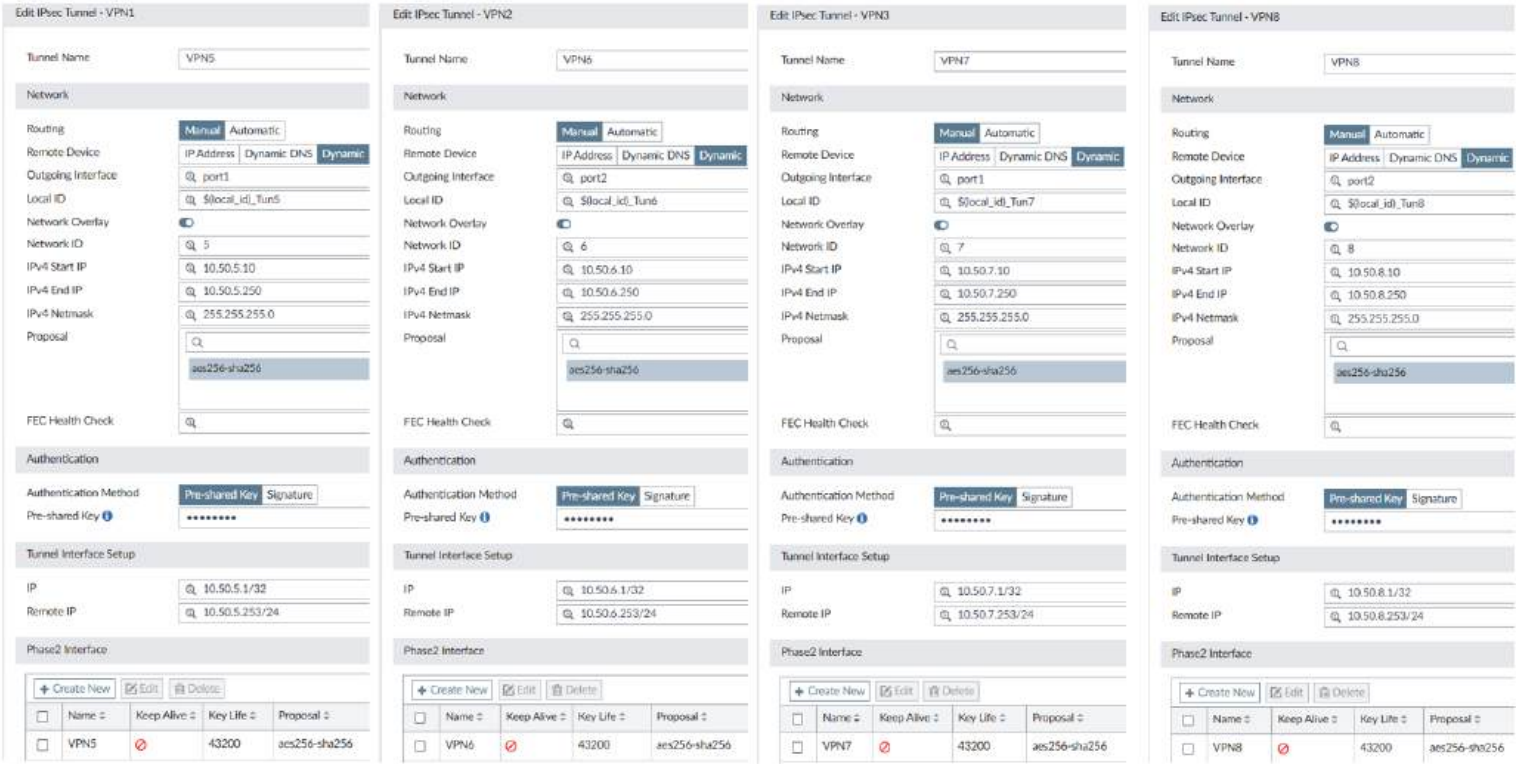
To create VPN5 – VPN8, clone the existing HUB1_IPsec template and then clone the already configured tunnels for HUB1. The Template for HUB2 will have VPNs 5-8 only with configurations changed to match the new tunnels.



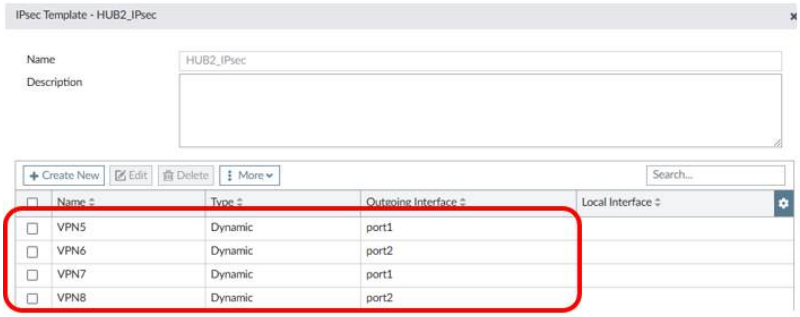
TUNNNEL	SUBNET	HUB1 ISP	BRANCHx ISP	NETWORK ID
TUN1	10.50.1.0/24	ISPA (port1)	ISPA (port1)	1
TUN2	10.50.2.0/24	ISPB (port2)	ISPB (port2)	2
TUN3	10.50.3.0/24	ISPA (port1)	ISPB (port2)	3
TUN4	10.50.4.0/24	ISPB (port2)	ISPA (port1)	4
TUNNNEL	SUBNET	HUB2 ISP	BRANCHx ISP	NETWORK ID
TUN5	10.50.5.0/24	ISPA (port1)	ISPA (port1)	5
TUN6	10.50.6.0/24	ISPB (port2)	ISPB (port2)	6
TUN7	10.50.7.0/24	ISPA (port1)	ISPB (port2)	7
TUN8	10.50.8.0/24	ISPB (port2)	ISPA (port1)	8

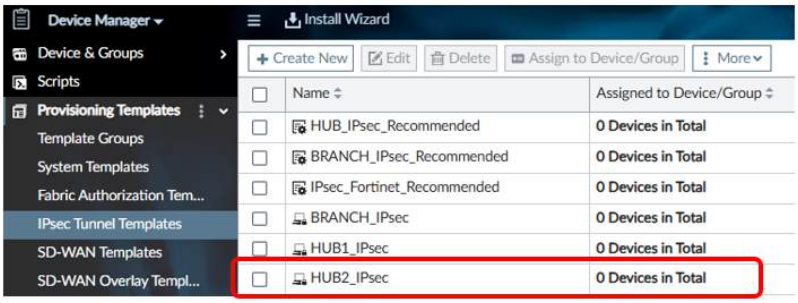


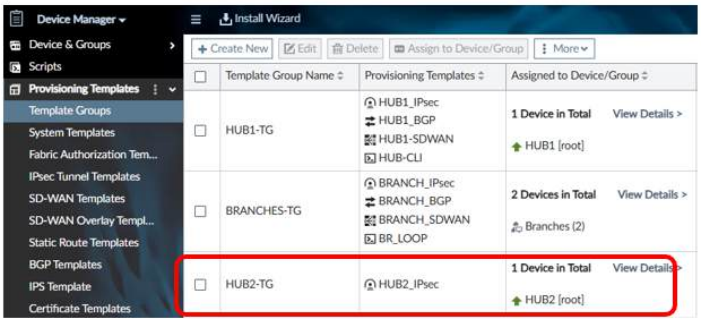
IPSEC HUB2 VPN5-8 CONFIGURATION



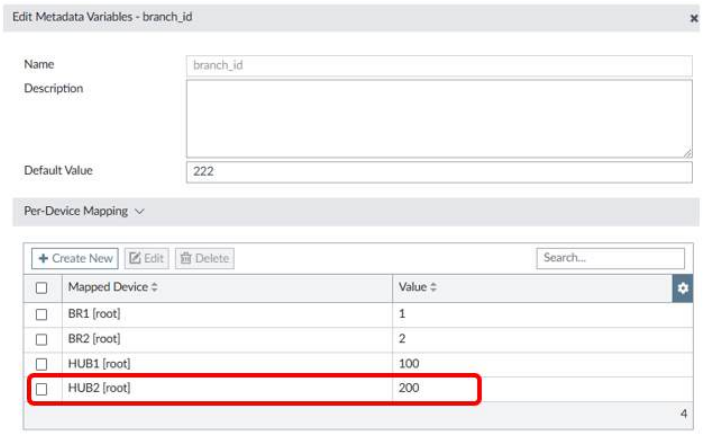
A new IPsec Template (HUB2_IPsec) is created and attached to the new Template Group (HUB2_TG) for HUB2

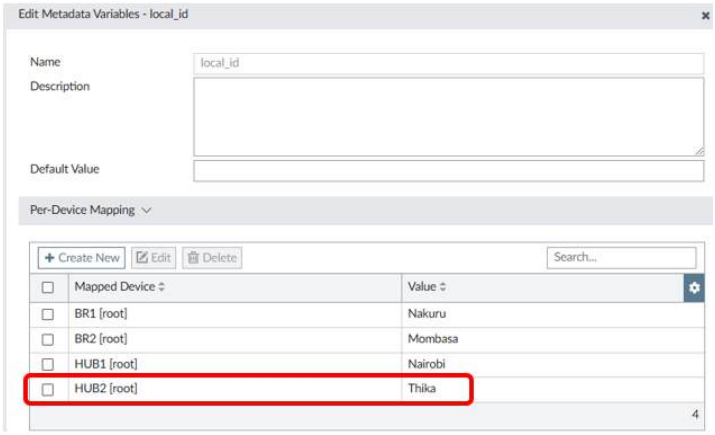






For HUB2, new mappings are created for the required Metadata Variables i.e. local ID **(Thika)** and Branch ID **(200).** This avoids errors when pushing the policies to the devices.





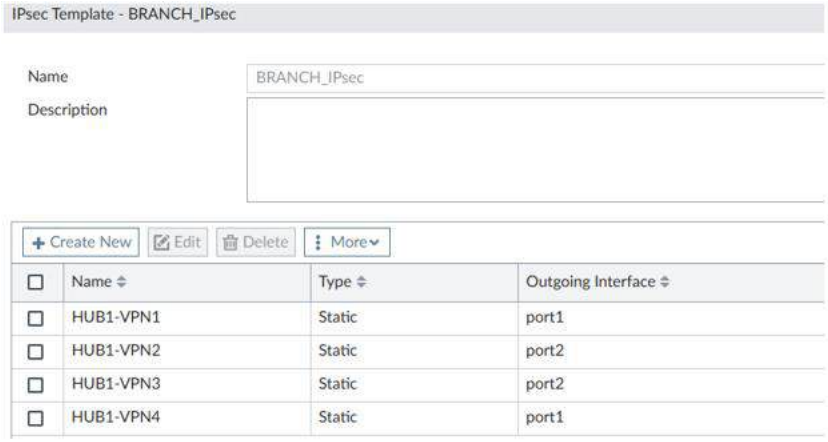
IPSEC HUB2 Configuration Preview

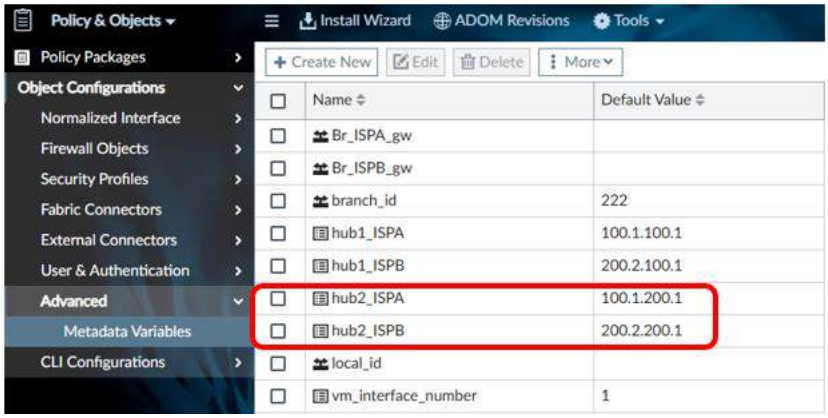
```
config vpn ipsec phase1-interface
    edit "VPN5"
        set type dynamic
        set interface "port1"
        set ike-version 2
        set dpd on-idle
        set comments "VPN: VPN5 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set localid "Thika_Tun5"
        set dpd-retryinterval 60
        set net-device disable
        set add-route disable
        set auto-discovery-sender enable
        set ipv4-start-ip 10.50.5.10
        set ipv4-end-ip 10.50.5.250
        set ipv4-netmask 255.255.255.0
        set psksecret ENC Z8Zpc/bwU2j1HxCFsp0zLVsmpXEWQQvc6JVo
        set network-overlay enable
        set network-id 5
    next
end
config system interface
    edit "VPN5"
        set vdom "root"
        set ip 10.50.5.1 255.255.255.255
        set type tunnel
        set remote-ip 10.50.5.253 255.255.255.0
        set snmp-index 116
        set interface "port1"
    next
end
config vpn ipsec phase1-interface
    edit "VPN6"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set dpd on-idle
        set comments "VPN: VPN6 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set localid "Thika_Tun6"
        set dpd-retryinterval 60
        set net-device disable
        set add-route disable
        set auto-discovery-sender enable
        set ipv4-start-ip 10.50.6.10
        set ipv4-end-ip 10.50.6.250
        set ipv4-netmask 255.255.255.0
        set psksecret ENC Z8Zpc/bwU2j1HxCFsp0zLVsmpXEWQQvc6JVo
        set network-overlay enable
        set network-id 6
    next
end
config system interface
    edit "VPN6"
        set vdom "root"
        set ip 10.50.6.1 255.255.255.255
        set type tunnel
        set remote-ip 10.50.6.253 255.255.255.0
        set snmp-index 117
        set interface "port2"
    next
end
config vpn ipsec phase1-interface
    edit "VPN7"
        set type dynamic
        set interface "port1"
        set ike-version 2
        set dpd on-idle
        set comments "VPN: VPN7 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set localid "Thika_Tun7"
        set dpd-retryinterval 60
        set net-device disable
        set add-route disable
        set auto-discovery-sender enable
        set ipv4-start-ip 10.50.7.10
        set ipv4-end-ip 10.50.7.250
        set ipv4-netmask 255.255.255.0
        set psksecret ENC Z8Zpc/bwU2j1HxCFsp0zLVsmpXEWQQvc6JVo
        set network-overlay enable
        set network-id 7
    next
end
config system interface
    edit "VPN7"
        set vdom "root"
        set ip 10.50.7.1 255.255.255.255
        set type tunnel
        set remote-ip 10.50.7.253 255.255.255.0
        set snmp-index 118
        set interface "port1"
    next
end
config vpn ipsec phase1-interface
    edit "VPN8"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set dpd on-idle
        set comments "VPN: VPN8 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set localid "Thika_Tun8"
        set dpd-retryinterval 60
        set net-device disable
        set add-route disable
        set auto-discovery-sender enable
        set ipv4-start-ip 10.50.8.10
        set ipv4-end-ip 10.50.8.250
        set ipv4-netmask 255.255.255.0
        set psksecret ENC Z8Zpc/bwU2j1HxCFsp0zLVsmpXEWQQvc6JVo
        set network-overlay enable
        set network-id 8
    next
end
config system interface
    edit "VPN8"
        set vdom "root"
```

```
        set ip 10.50.8.1 255.255.255.255
        set type tunnel
        set remote-ip 10.50.8.253 255.255.255.0
        set snmp-index 119
        set interface "port2"
    next
end
config vpn ipsec phase2-interface
    edit "VPN5"
        set phase1name "VPN5"
        set proposal aes256-sha256
        set comments "VPN: VPN5 [Created by IPSEC Template]"
    next
    edit "VPN6"
        set phase1name "VPN6"
        set proposal aes256-sha256
        set comments "VPN: VPN6 [Created by IPSEC Template]"
    next
    edit "VPN7"
        set phase1name "VPN7"
        set proposal aes256-sha256
        set comments "VPN: VPN7 [Created by IPSEC Template]"
    next
    edit "VPN8"
        set phase1name "VPN8"
        set proposal aes256-sha256
        set comments "VPN: VPN8 [Created by IPSEC Template]"
    next
end

```

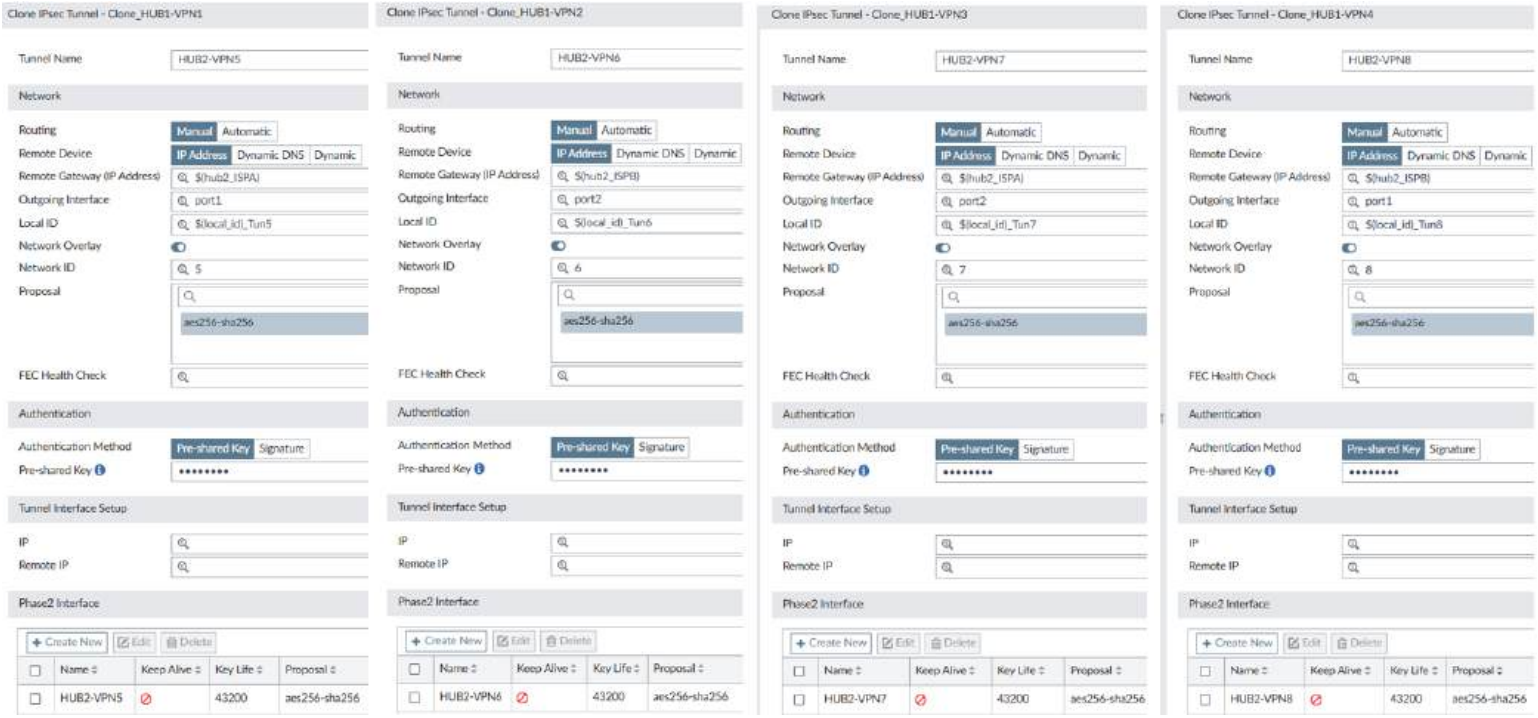
IPSEC BRANCH Configuration for HUB2

For branches, add the new HUB2-VPNx tunnels by cloning HUB1-VPNx tunnels and define HUB2 ISPA & ISPB metadata variables referenced in the IPsec tunnels to be created.

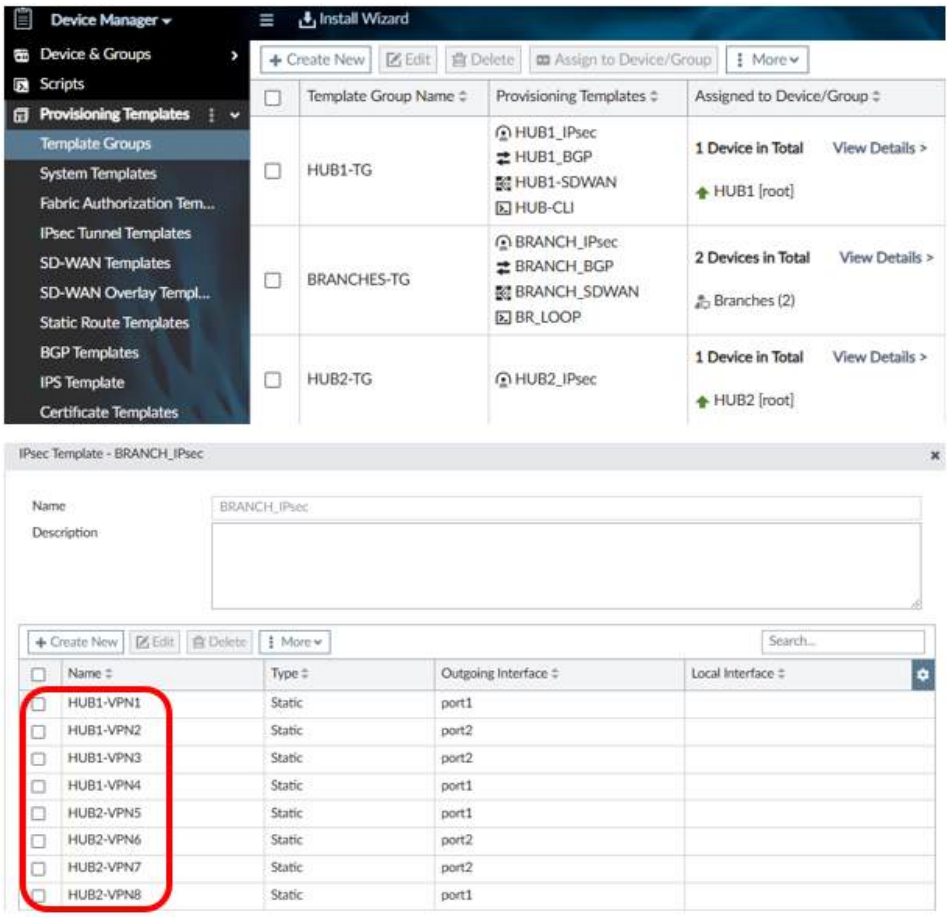




IPSEC BRANCH HUB2-VPN5-8 CONFIGURATION



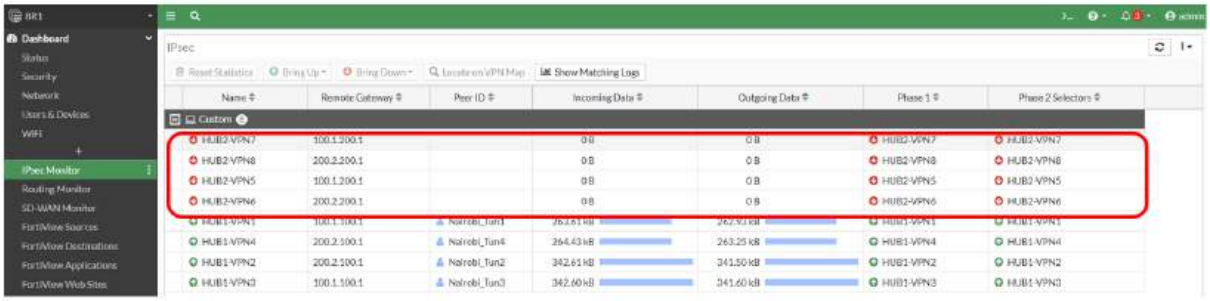
The same IPSec template used for HUB1 is used to add the additional tunnels for branches. A total of 8 IPSec tunnels are configured for each branch 1-4 for HUB1 and 5-8 for HUB2.

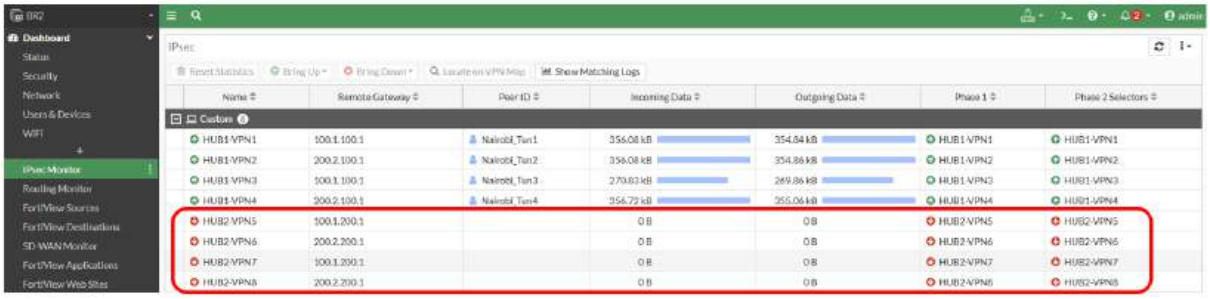


IPSEC BRANCH HUB2-VPN5-8 Configuration Preview

```
config vpn ipsec phase1-interface
    edit "HUB2-VPN5"
        set interface "port1"
        set ike-version 2
        set comments "VPN: HUB2-VPN5 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set localid "Nakuru_Tun5"
        set remote-gw 100.1.200.1
        set idle-timeout enable
        set net-device enable
        set add-route disable
        set auto-discovery-receiver enable
        set psksecret ENC Z8Zpc/bwU2j1HxCFsp0zLVsmpXEWQQvc6JVoYq8gt3
        set network-overlay enable
        set network-id 5
        set auto-discovery-shortcuts dependent
    next
end
config system interface
    edit "HUB2-VPN5"
        set vdom "root"
        set type tunnel
        set snmp-index 121
        set interface "port1"
    next
end
config vpn ipsec phase1-interface
    edit "HUB2-VPN6"
        set interface "port2"
        set ike-version 2
        set comments "VPN: HUB2-VPN6 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set localid "Nakuru_Tun6"
        set remote-gw 200.2.200.1
        set idle-timeout enable
        set net-device enable
        set add-route disable
        set auto-discovery-receiver enable
        set psksecret ENC Z8Zpc/bwU2j1HxCFsp0zLVsmpXEWQQvc6JVoYq8gt3R
        set network-overlay enable
        set network-id 6
        set auto-discovery-shortcuts dependent
    next
end
config system interface
    edit "HUB2-VPN6"
        set vdom "root"
        set type tunnel
        set snmp-index 122
        set interface "port2"
    next
end
config vpn ipsec phase1-interface
    edit "HUB2-VPN7"
        set interface "port2"
        set ike-version 2
        set comments "VPN: HUB2-VPN7 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set localid "Nakuru_Tun7"
        set remote-gw 100.1.200.1
        set idle-timeout enable
        set net-device enable
        set add-route disable
        set auto-discovery-receiver enable
        set psksecret ENC Z8Zpc/bwU2j1HxCFsp0zLVsmpXEWQQvc6JVoYq8gt3R
        set network-overlay enable
        set network-id 7
        set auto-discovery-shortcuts dependent
    next
end
config system interface
    edit "HUB2-VPN7"
        set vdom "root"
        set type tunnel
        set snmp-index 123
        set interface "port2"
    next
end
config vpn ipsec phase1-interface
    edit "HUB2-VPN8"
        set interface "port1"
        set ike-version 2
        set comments "VPN: HUB2-VPN8 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set localid "Nakuru_Tun8"
        set remote-gw 200.2.200.1
        set idle-timeout enable
        set net-device enable
        set add-route disable
        set auto-discovery-receiver enable
        set psksecret ENC Z8Zpc/bwU2j1HxCFsp0zLVsmpXEWQQvc6JVoYq8gt3R
        set network-overlay enable
        set network-id 8
        set auto-discovery-shortcuts dependent
    next
end
config system interface
    edit "HUB2-VPN8"
        set vdom "root"
        set type tunnel
        set snmp-index 124
        set interface "port1"
    next
end
config vpn ipsec phase2-interface
    edit "HUB2-VPN5"
        set phase1name "HUB2-VPN5"
        set proposal aes256-sha256
        set auto-negotiate enable
        set comments "VPN: HUB2-VPN5 [Created by IPSEC Template]"
    next
    edit "HUB2-VPN6"
        set phase1name "HUB2-VPN6"
        set proposal aes256-sha256
        set auto-negotiate enable
        set comments "VPN: HUB2-VPN6 [Created by IPSEC Template]"
    next
    edit "HUB2-VPN7"
        set phase1name "HUB2-VPN7"
        set proposal aes256-sha256
        set auto-negotiate enable
        set comments "VPN: HUB2-VPN7 [Created by IPSEC Template]"
    next
    edit "HUB2-VPN8"
        set phase1name "HUB2-VPN8"
        set proposal aes256-sha256
        set auto-negotiate enable
        set comments "VPN: HUB2-VPN8 [Created by IPSEC Template]"
    next
end
```

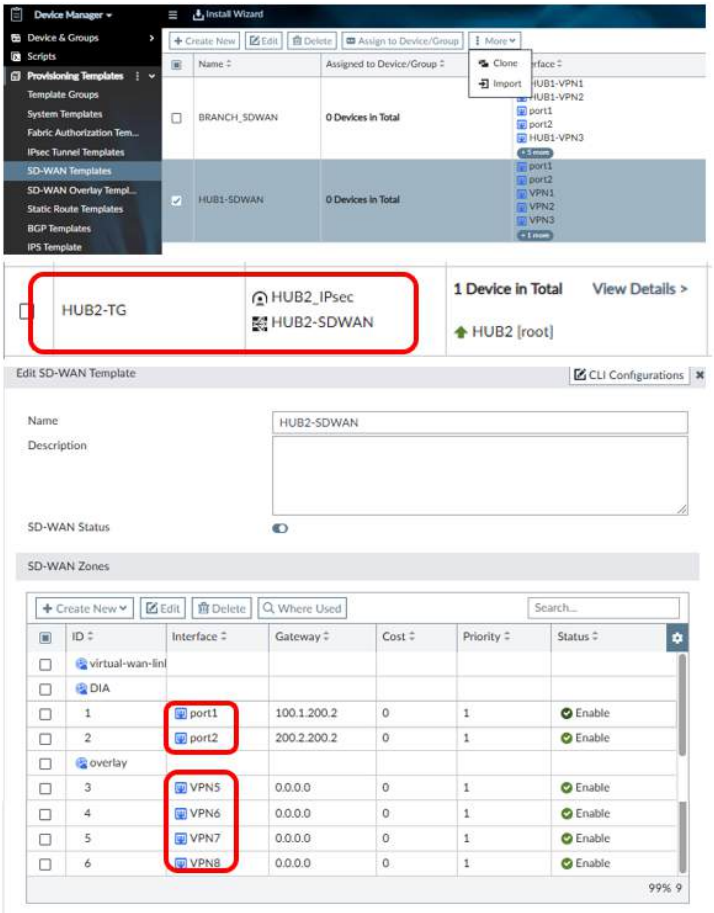
After configuration, the tunnels towards HUB2 will all be down because they are not assigned to SDWAN zones nor configured security policies required. Zones are configured next to resolve this.





SDWAN HUB2 Configuration

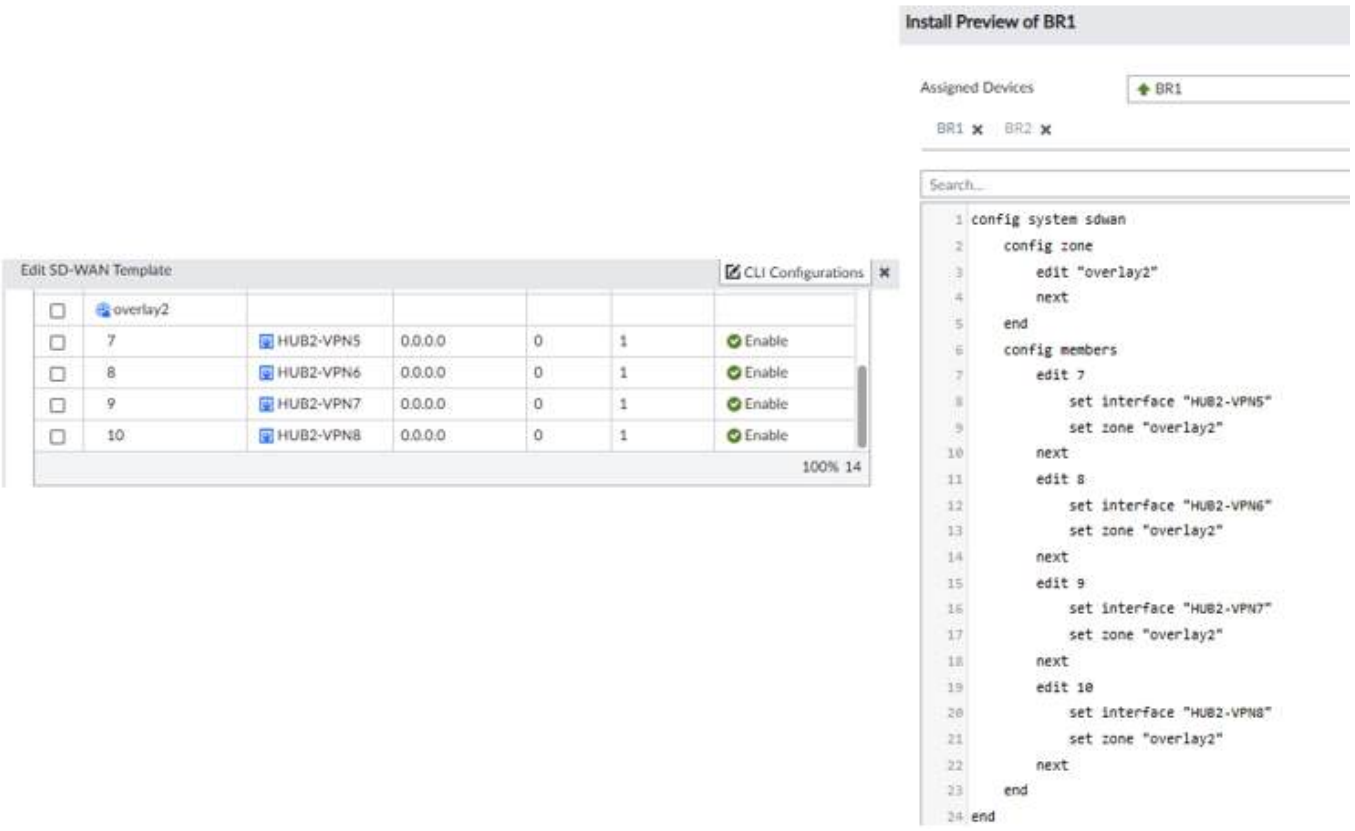
For HUB2, we create the DIA zone and overlay zone. The VPN interfaces configured earlier VPN5-VPN8 are assigned to the overlay zone while Port1 and port2 undelay interfaces are assigned to the DIA zone for internet access.



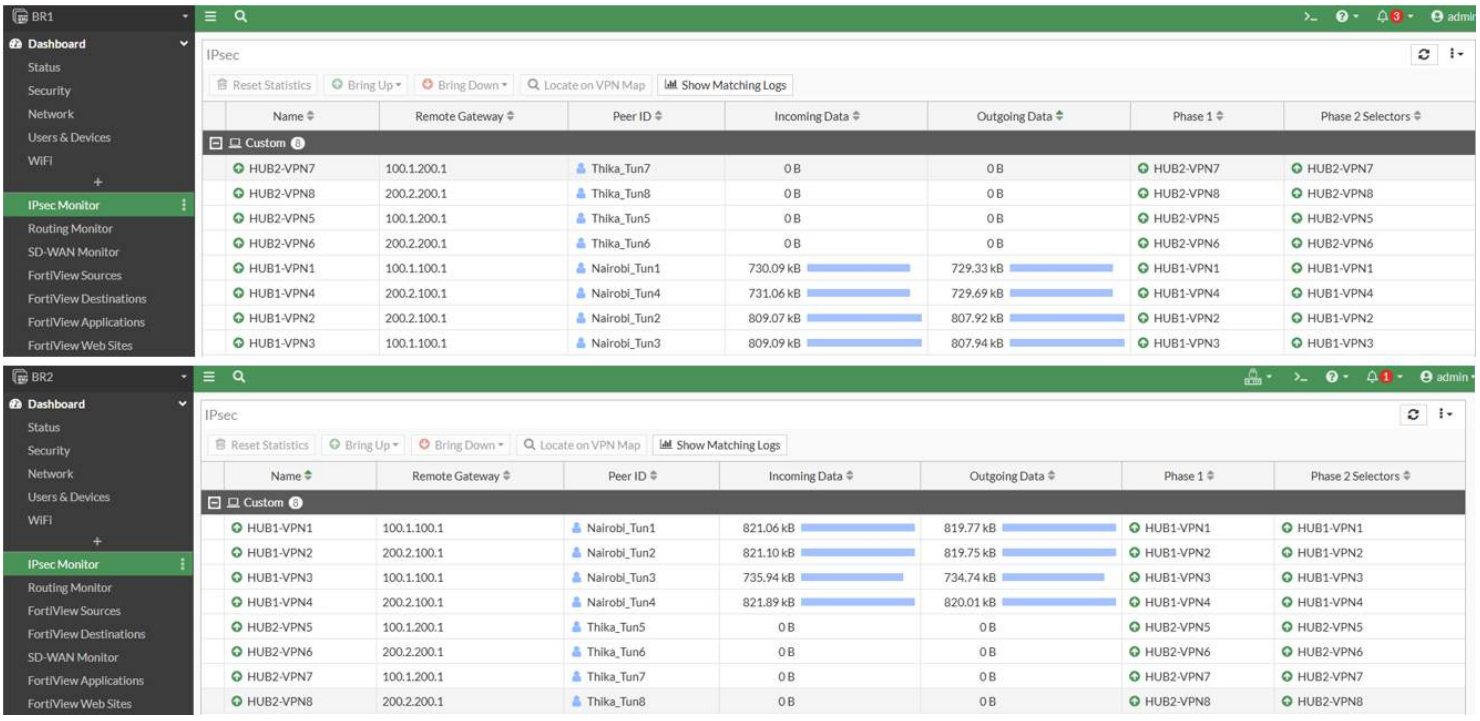
```
config system sdwan
    set status enable
    config zone
        edit "DIA"
        next
        edit "overlay"
        next
    end
    config members
        edit 1
            set interface "port1"
            set zone "DIA"
            set gateway 100.1.200.2
        next
        edit 2
            set interface "port2"
            set zone "DIA"
            set gateway 200.2.200.2
        next
        edit 3
            set interface "VPN5"
            set zone "overlay"
        next
        edit 4
            set interface "VPN6"
            set zone "overlay"
        next
        edit 5
            set interface "VPN7"
            set zone "overlay"
        next
        edit 6
            set interface "VPN8"
            set zone "overlay"
        next
    end
end
```

SDWAN ZONES BRANCH Configuration for HUB2

A new zone is configured for the tunnels towards HUB2 i.e. **overlay2**. The new tunnels are assigned to this zone.



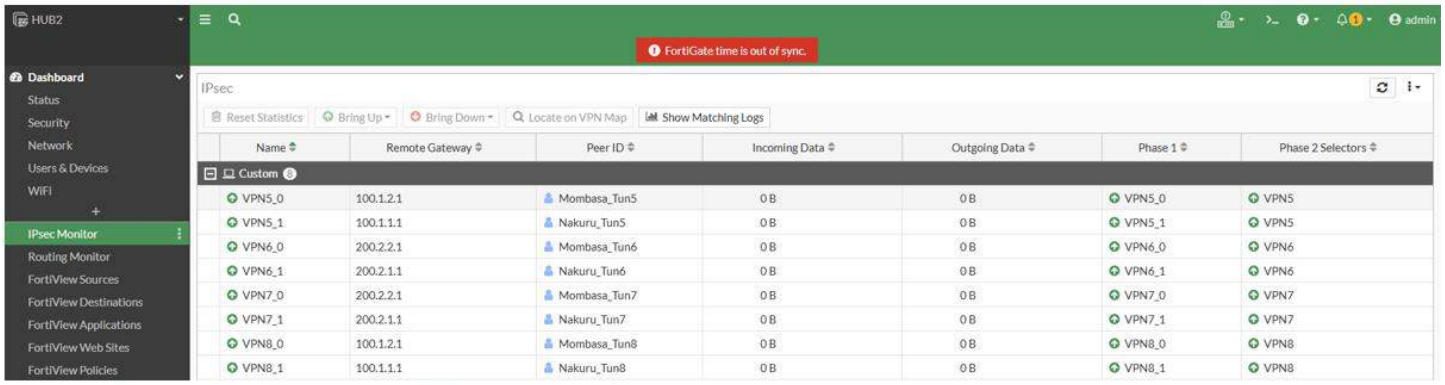
After configuring the zones at HUB2 and branches, all eight tunnels are now up.



SDWAN HUB2 Verification

From HUB2, 8 tunnels are established towards the branches. This is the same number of tunnels at HUB1. Any additional site with 2 ISP links will add four tunnels at each HUB.

With internet links, we can easily change ISP links at the branches with minor configuration changes. The HUBs require static IP addressing but branches can use DHCP assigned WAN ip addressing.

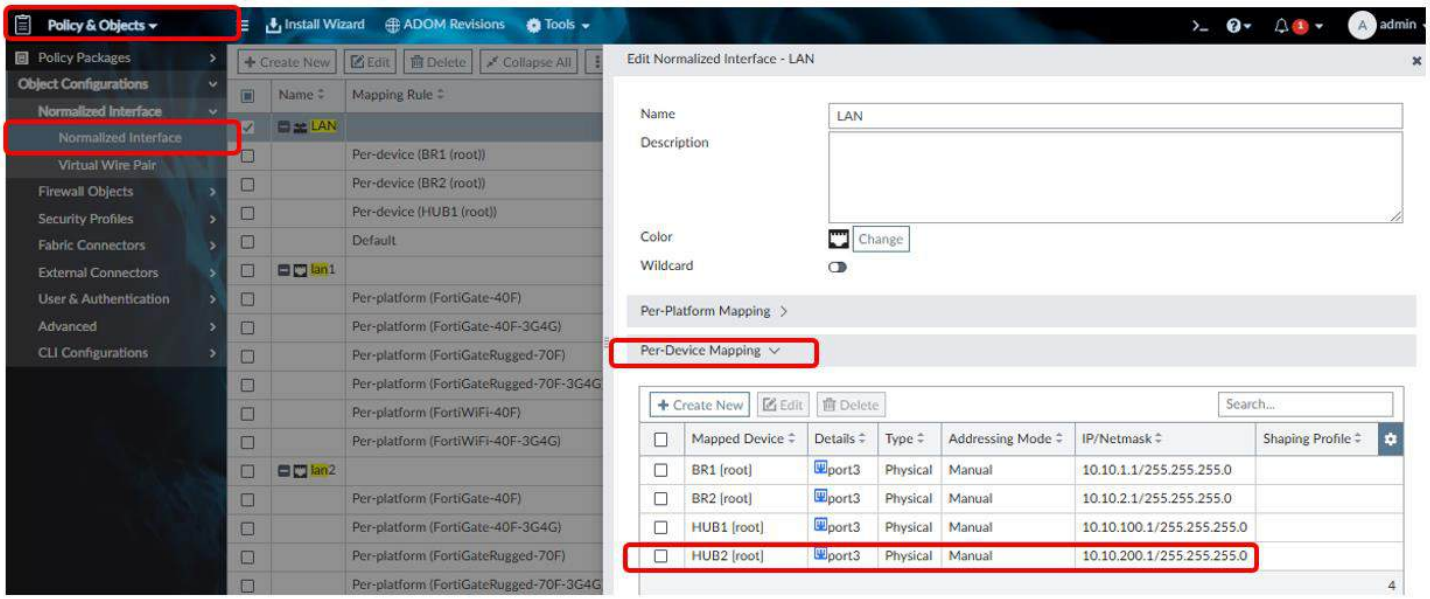



```
HUB2 # diagnose vpn ike gateway list | grep assigned
assigned IPv4 address: 10.50.5.10/255.255.255.0
assigned IPv4 address: 10.50.7.10/255.255.255.0
assigned IPv4 address: 10.50.7.11/255.255.255.0
assigned IPv4 address: 10.50.5.11/255.255.255.0
assigned IPv4 address: 10.50.8.10/255.255.255.0
assigned IPv4 address: 10.50.6.10/255.255.255.0
assigned IPv4 address: 10.50.8.11/255.255.255.0
assigned IPv4 address: 10.50.6.11/255.255.255.0
```

TUNNNEL	SUBNET	HUB1 ISP	BRANCHx ISP	NETWORK ID
TUN1	10.50.1.0/24	ISPA (port1)	ISPA (port1)	1
TUN2	10.50.2.0/24	ISPB (port2)	ISPB (port2)	2
TUN3	10.50.3.0/24	ISPA (port1)	ISPB (port2)	3
TUN4	10.50.4.0/24	ISPB (port2)	ISPA (port1)	4
TUNNNEL	**SUBNET**	**HUB2 ISP**	**BRANCHx ISP**	**NETWORK ID**
TUN5	10.50.5.0/24	ISPA (port1)	ISPA (port1)	5
TUN6	10.50.6.0/24	ISPB (port2)	ISPB (port2)	6
TUN7	10.50.7.0/24	ISPA (port1)	ISPB (port2)	7
TUN8	10.50.8.0/24	ISPB (port2)	ISPA (port1)	8

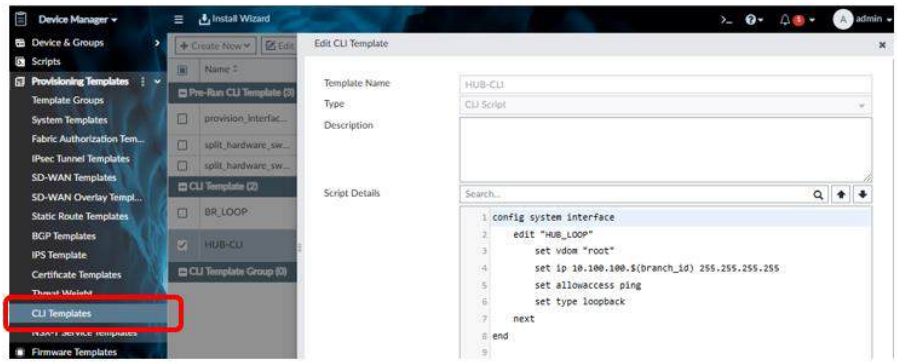
HUB2 Normalized Interface

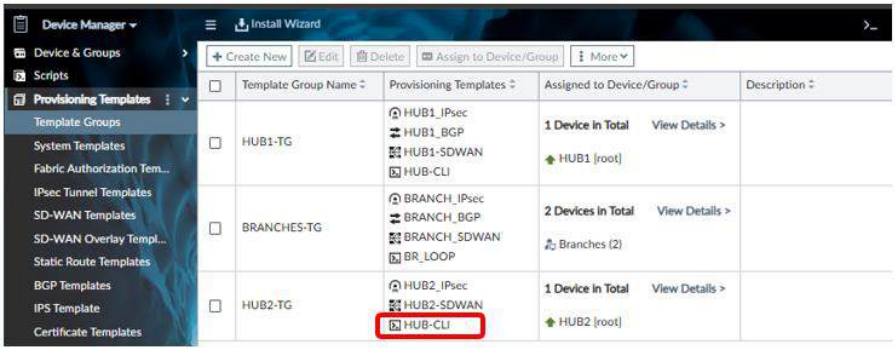
To configure policies by using similar objects, HUB2 is added to the normalized interface referencing the internal interface, LAN. This is port3 as shown on the diagram below.

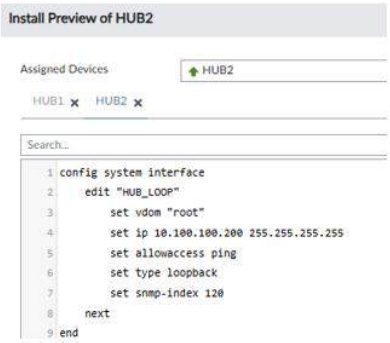


HUB2 LOOPBACK SETUP

A Loopback interface is configured at the HUB2 with **10.100.100.200/32** A firewall policy is required at HUB2 to allow traffic to the loopback. CLI templates are used to achieve this



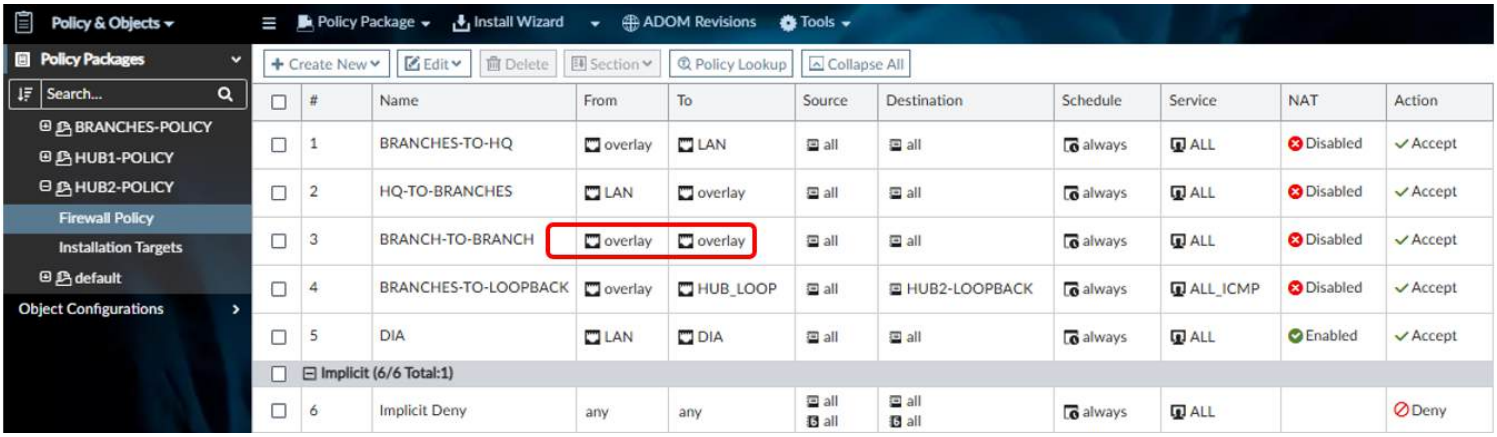




HUB2 Firewall Policies

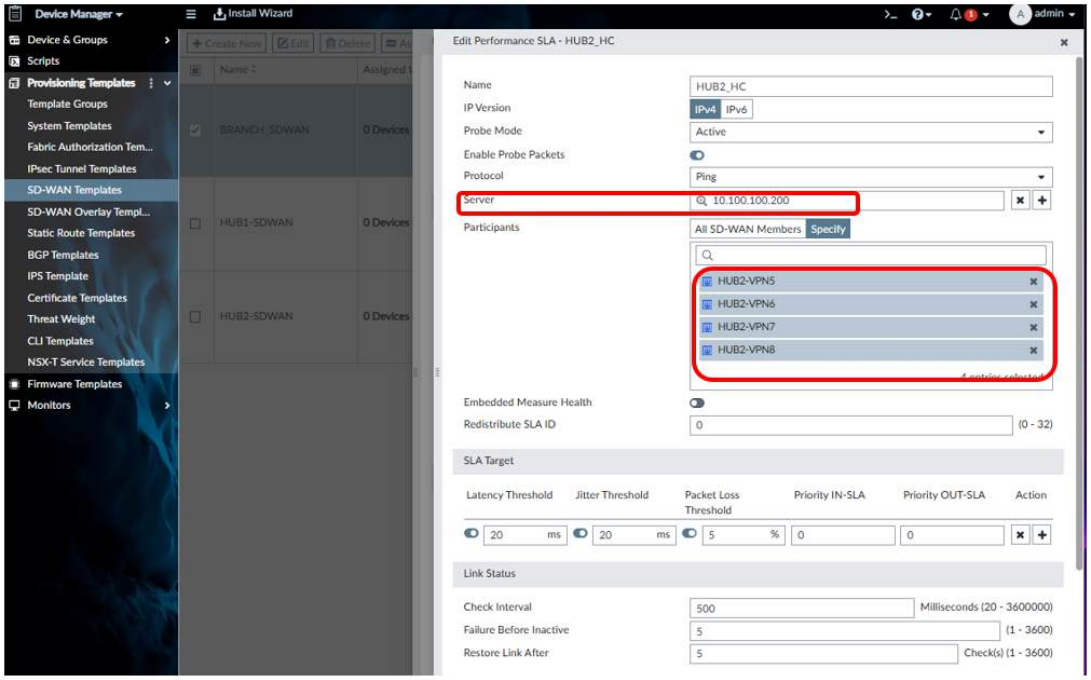
Five key policies are required at the HUB.

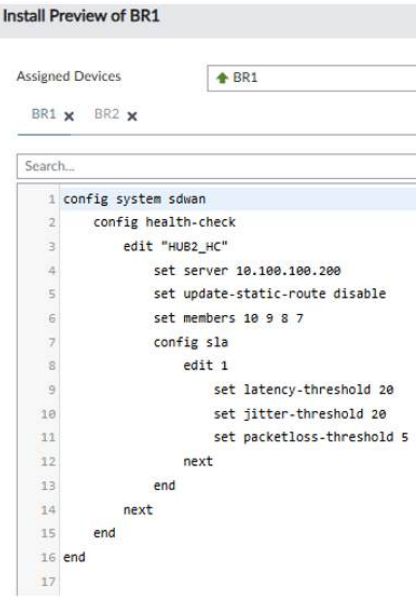
- Branches to HQ LAN inbound on the overlay network
- Outbound rule to allow HQ to access branches on the overlay network
- Branch to Branch traffic hair pinned at the HUB i.e. source and destination interfaces are the overlay network
- Branches to LOOPBACK for Healthcheck on the overlay network
- Direct internet access (DIA) using the underlay network. This will have NAT enabled



HUB2 LOOPBACK Health Check

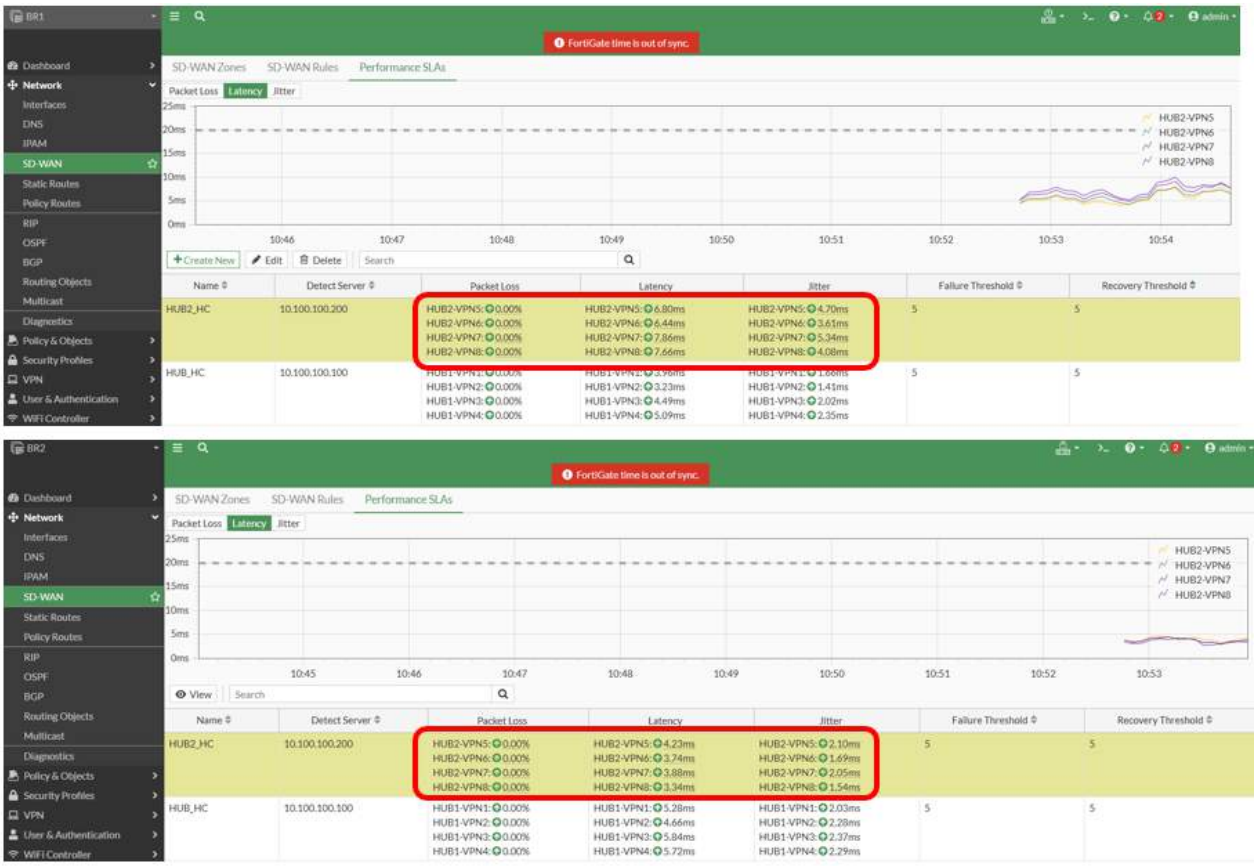
The Healthcheck is configured using the SDWAN template for the branches. This probes the loopback at HUB2 using the overlay2 networks according to defined SLA metrics.





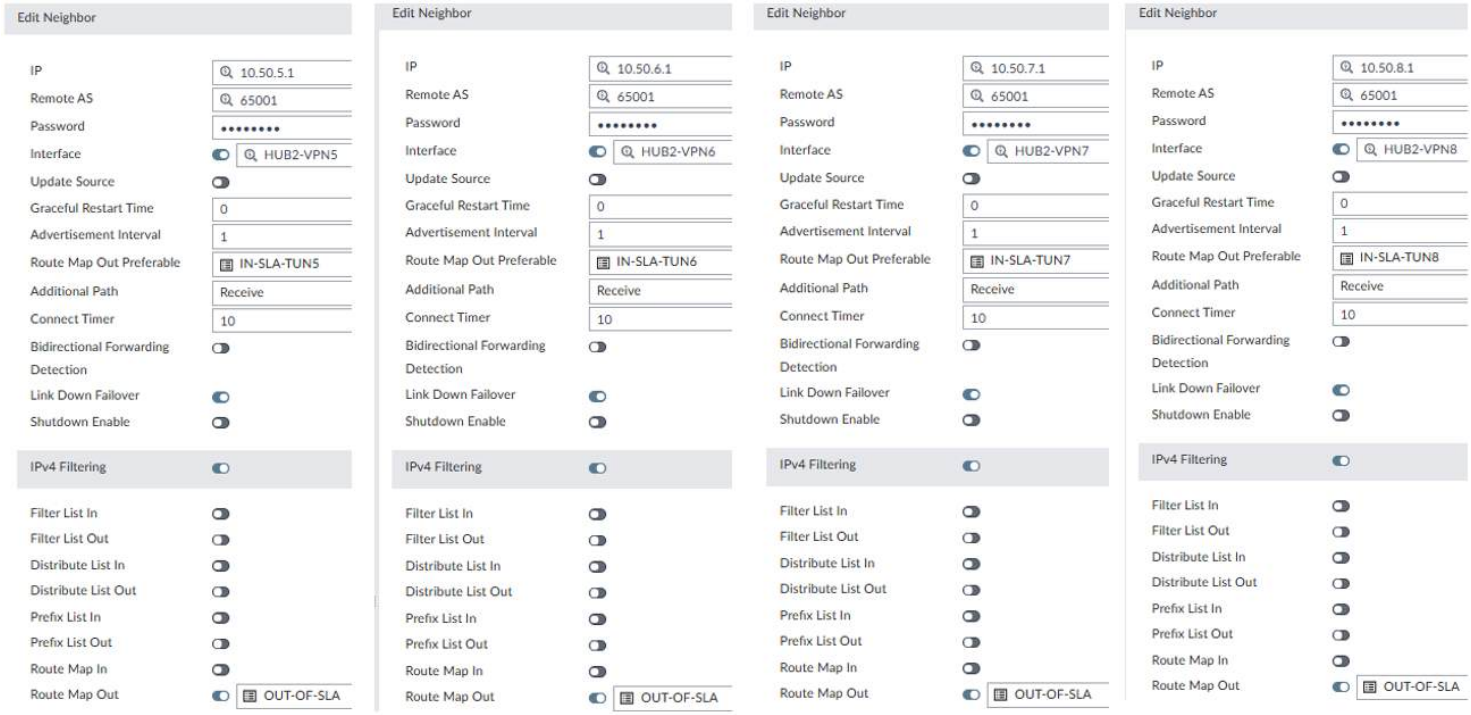
HUB2 LOOPBACK Health Check Verification

With the tunnels up, healthcheck configured and firewall policies set, we proceed to configure BGP.



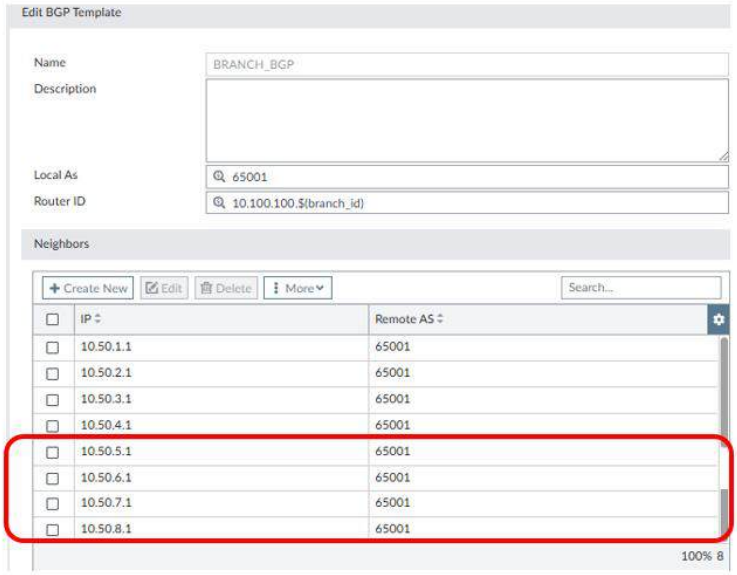
BRANCHES BGP Configuration for HUB2

For the branches, configure additional neighbors by cloning existing neighbors, modifying the Route Map Out and respective community list and remap to the correct interface. The Route Map Out for out of SLA state is still the same as used for all neighbors.



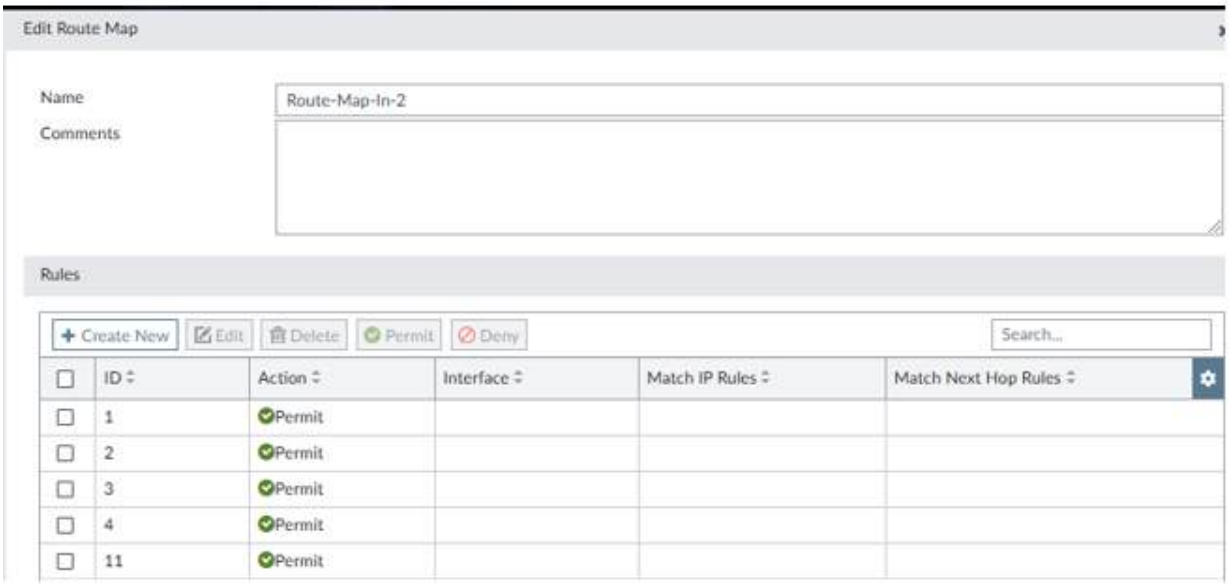
BRANCHES BGP Configuration Preview for HUB2

```
config router route-map
    edit "IN-SLA-TUN5"
        config rule
            edit 1
                set set-community "65001:5"
            next
        end
    next
    edit "IN-SLA-TUN6"
        config rule
            edit 1
                set set-community "65001:6"
            next
        end
    next
    edit "IN-SLA-TUN7"
        config rule
            edit 1
                set set-community "65001:7"
            next
        end
    next
    edit "IN-SLA-TUN8"
        config rule
            edit 1
                set set-community "65001:8"
            next
        end
    next
end
config router bgp
    config neighbor
        edit "10.50.5.1"
            set advertisement-interval 1
            set capability-graceful-restart enable
            set link-down-failover enable
            set soft-reconfiguration enable
            set description "HUB2-VPN5"
            set interface "HUB2-VPN5"
            set remote-as 65001
            set route-map-out "OUT-OF-SLA"
            set connect-timer 10
            set additional-path receive
            set route-map-out-preferable "IN-SLA-TUN5"
        next
        edit "10.50.6.1"
            set advertisement-interval 1
            set capability-graceful-restart enable
            set link-down-failover enable
            set soft-reconfiguration enable
            set description "HUB2-VPN6"
            set interface "HUB2-VPN6"
            set remote-as 65001
            set route-map-out "OUT-OF-SLA"
            set connect-timer 10
            set additional-path receive
            set route-map-out-preferable "IN-SLA-TUN6"
        next
        edit "10.50.7.1"
            set advertisement-interval 1
            set capability-graceful-restart enable
            set link-down-failover enable
            set soft-reconfiguration enable
            set description "HUB2-VPN7"
            set interface "HUB2-VPN7"
            set remote-as 65001
            set route-map-out "OUT-OF-SLA"
            set connect-timer 10
            set additional-path receive
            set route-map-out-preferable "IN-SLA-TUN7"
        next
        edit "10.50.8.1"
            set advertisement-interval 1
            set capability-graceful-restart enable
            set link-down-failover enable
            set soft-reconfiguration enable
            set description "HUB2-VPN8"
            set interface "HUB2-VPN8"
            set remote-as 65001
            set route-map-out "OUT-OF-SLA"
            set connect-timer 10
            set additional-path receive
            set route-map-out-preferable "IN-SLA-TUN8"
        next
```

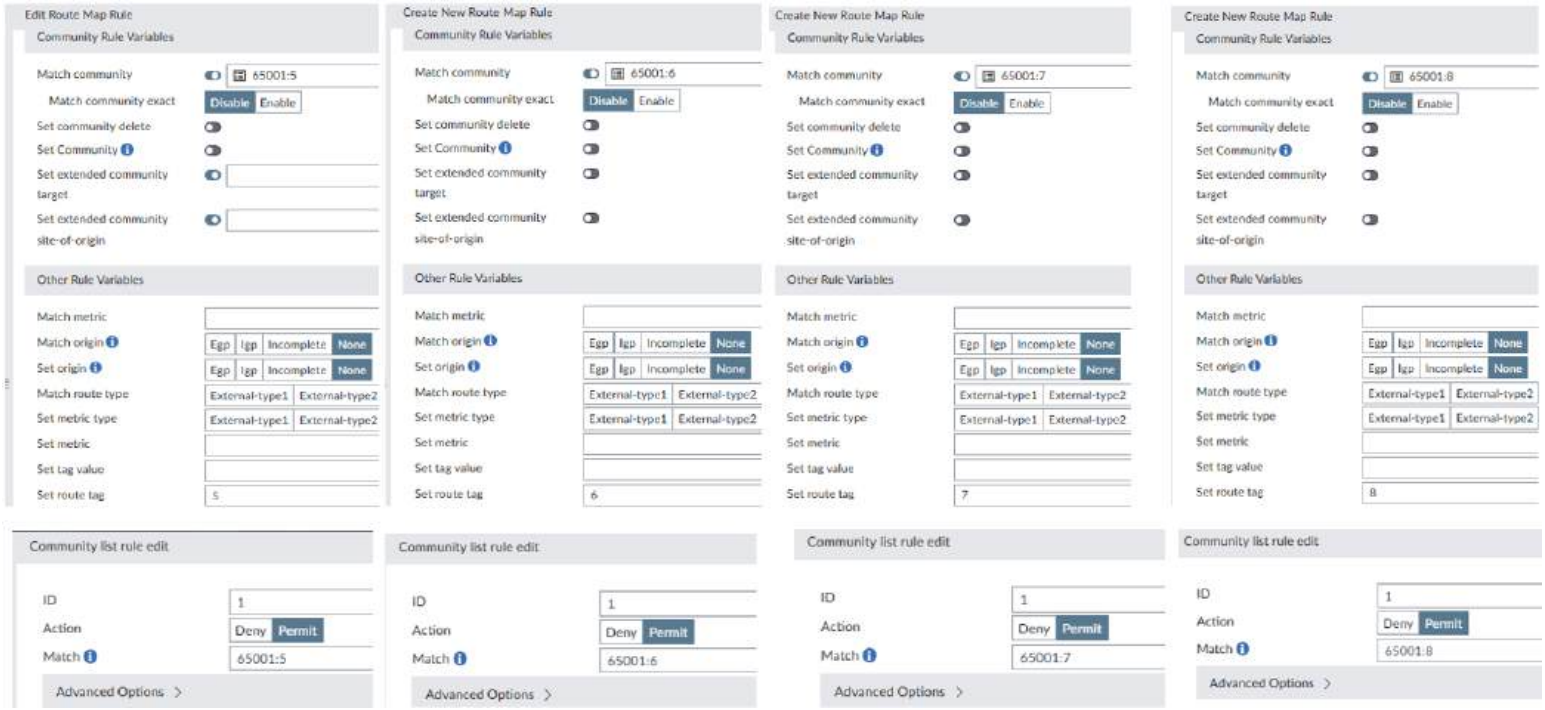


HUB2 BGP Configuration

For HUB2, BGP is configured by cloning the existing template for HUB1. To avoid duplication, a new route-map-in is configured with new Rules to match the IN-SLA state for the new communities 65001:5 to 65001:8 For OUT-OF-SLA state, the same rule is used i.e. 65001:11

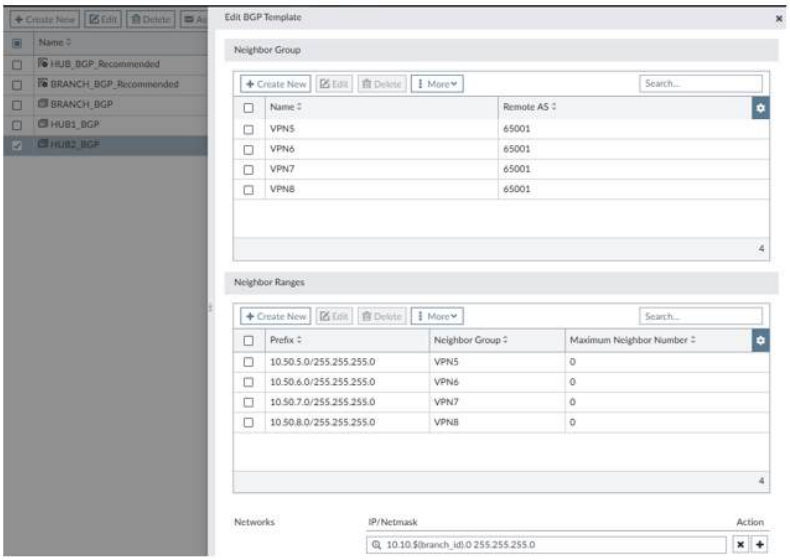


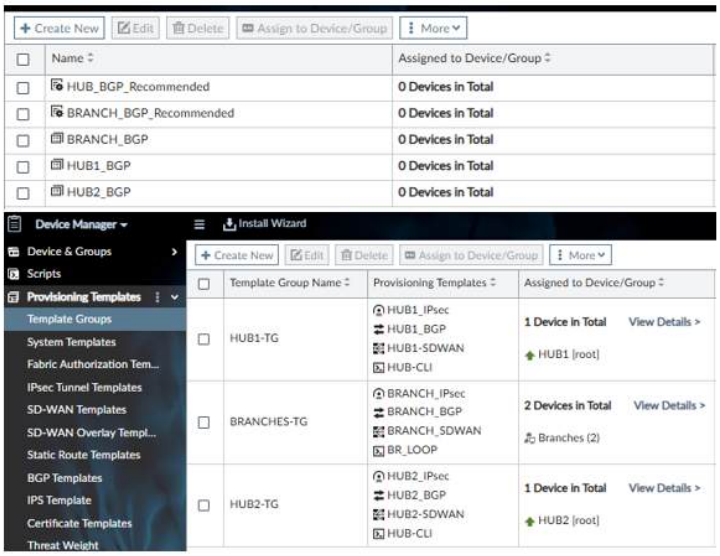
HUB2 BGP Route-Maps Configuration



HUB2 BGP Configuration

New neighbor groups are defined for VPN5-8 and attached to neighbor ranges.





HUB2 BGP Configuration Preview

```
config router community-list
    edit "65001:11"
        config rule
            edit 1
                set action permit
                set match "65001:11"
            next
        end
    next
    edit "65001:5"
        config rule
            edit 1
                set action permit
                set match "65001:5"
            next
        end
    next
    edit "65001:6"
        config rule
            edit 1
                set action permit
                set match "65001:6"
            next
        end
    next
    edit "65001:7"
        config rule
            edit 1
                set action permit
                set match "65001:7"
            next
        end
    next
    edit "65001:8"
        config rule
            edit 1
                set action permit
                set match "65001:8"
            next
        end
    next
end
```

```
config router route-map
    edit "Route-Map-In-2"
        config rule
            edit 1
                set match-community "65001:5"
                set set-route-tag "5"
            next
            edit 2
                set match-community "65001:6"
                set set-route-tag "6"
            next
            edit 3
                set match-community "65001:7"
                set set-route-tag "7"
            next
            edit 4
                set match-community "65001:8"
                set set-route-tag "8"
            next
            edit 11
                set match-community "65001:11"
                set set-route-tag "11"
            next
        end
    next
end
config router bgp
    set as 65001
    set router-id 10.100.100.200
    set ibgp-multipath enable
    set additional-path enable
    set graceful-restart enable
    set additional-path-select 255
    config neighbor-group
        edit "VPN5"
            set capability-graceful-restart enable
            set link-down-failover enable
            set next-hop-self enable
            set soft-reconfiguration enable
            set remote-as 65001
            set route-map-in "Route-Map-In-2"
            set additional-path send
            set route-reflector-client enable
        next
        edit "VPN6"
            set capability-graceful-restart enable
            set link-down-failover enable
            set next-hop-self enable
            set soft-reconfiguration enable
            set remote-as 65001
            set route-map-in "Route-Map-In-2"
            set additional-path send
            set route-reflector-client enable
        next
        edit "VPN7"
            set capability-graceful-restart enable
            set link-down-failover enable
            set next-hop-self enable
            set soft-reconfiguration enable
            set remote-as 65001
            set route-map-in "Route-Map-In-2"
            set additional-path send
            set route-reflector-client enable
        next
        edit "VPN8"
            set capability-graceful-restart enable
            set link-down-failover enable
            set next-hop-self enable
            set soft-reconfiguration enable
            set remote-as 65001
            set route-map-in "Route-Map-In-2"
            set additional-path send
            set route-reflector-client enable
        next
    end
    config neighbor-range
        edit 1
            set prefix 10.50.5.0 255.255.255.0
            set neighbor-group "VPN5"
        next
        edit 2
            set prefix 10.50.6.0 255.255.255.0
            set neighbor-group "VPN6"
        next
        edit 3
            set prefix 10.50.7.0 255.255.255.0
            set neighbor-group "VPN7"
        next
        edit 4
            set prefix 10.50.8.0 255.255.255.0
            set neighbor-group "VPN8"
        next
    end
    config network
        edit 1
            set prefix 10.10.200.0 255.255.255.0
        next
    end
end
```

HUB2 SDWAN Configuration

For the new neighbors to get information on link performance, the SLA for HUB2 is defined for each neighbor towards HUB2 from the branch SDWAN template.

Neighbor

+ Create New | Edit | Delete | Search...

	Neighbor	Role	Interface Member	Performance SLA	SLA
☐	10.50.3.1	Standalone	HUB1-VPN3	HUB_HC	1
☐	10.50.4.1	Standalone	HUB1-VPN4	HUB_HC	1
☐	10.50.5.1	Standalone	HUB2-VPN5	HUB2_HC	1
☐	10.50.6.1	Standalone	HUB2-VPN6	HUB2_HC	1
☐	10.50.7.1	Standalone	HUB2-VPN7	HUB2_HC	1
☐	10.50.8.1	Standalone	HUB2-VPN8	HUB2_HC	1

98% 8

Install Preview of BR1

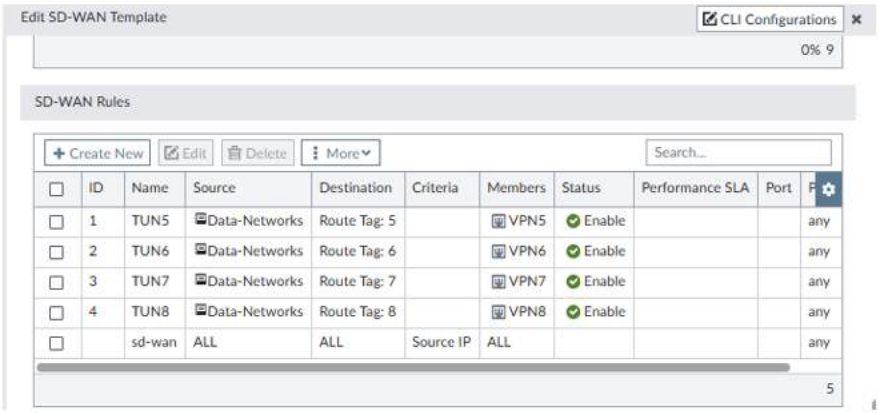
Assigned Devices: BR1

BR1 × BR2 ×

Search...

```
config system sdwan
    config neighbor
        edit "10.50.5.1"
            set health-check "HUB2_HC"
            set sla-id 1
            set member 7
        next
        edit "10.50.6.1"
            set health-check "HUB2_HC"
            set sla-id 1
            set member 8
        next
        edit "10.50.7.1"
            set health-check "HUB2_HC"
            set sla-id 1
            set member 9
        next
        edit "10.50.8.1"
            set health-check "HUB2_HC"
            set sla-id 1
            set member 10
        next
    end
end
```

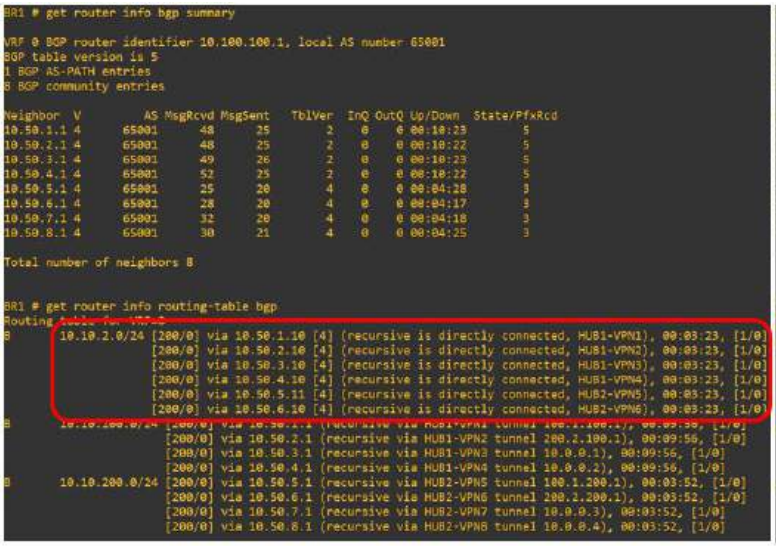
SDWAN rules are created at HUB2 referencing the route tags as destination to steer traffic sourced from the HUB towards branches.



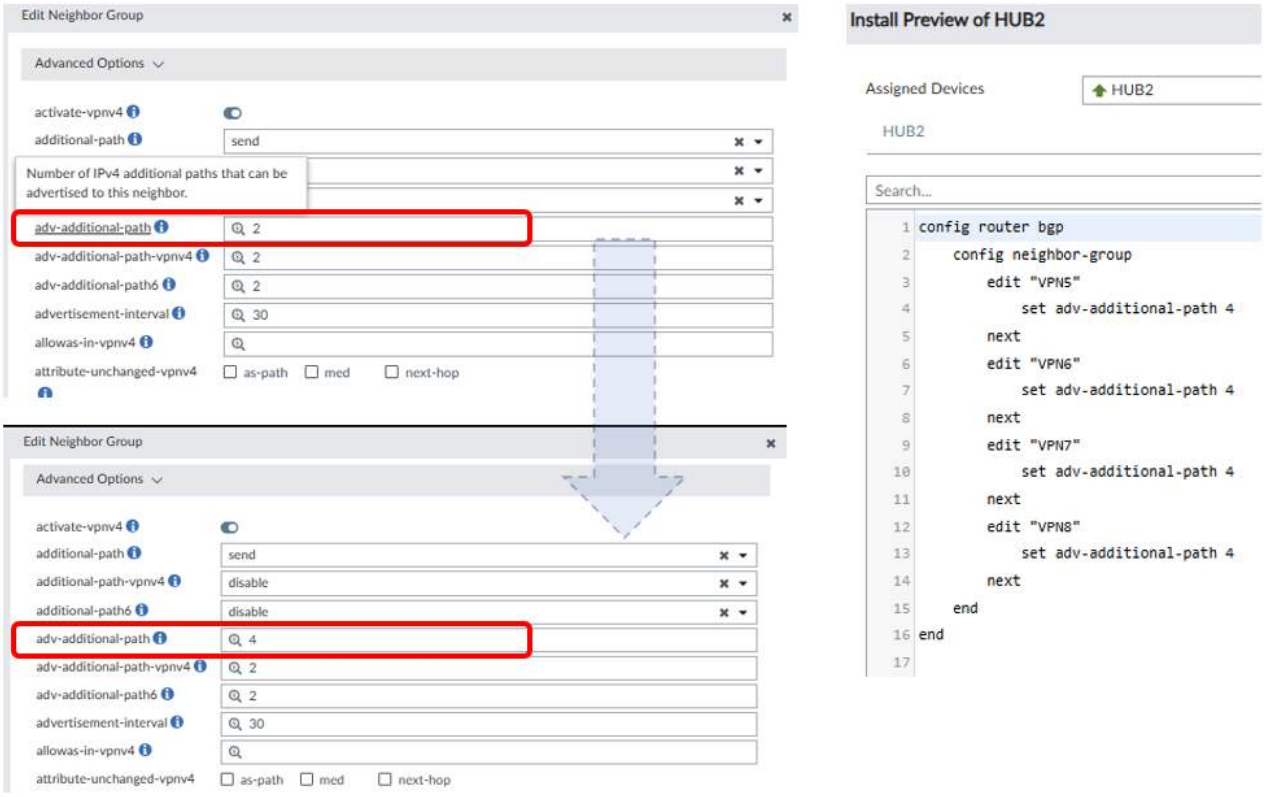

```
config firewall address
    edit "Data-Networks"
        set uuid 53bf5a3e-9541-51ef-5bd8-1ca0cce2e343
        set subnet 10.1.0.0 255.255.0.0
    next
end
config system sdwan
    config service
        edit 1
            set name "TUN5"
            set route-tag 5
            set src "Data-Networks"
            set priority-members 3
        next
        edit 2
            set name "TUN6"
            set route-tag 6
            set src "Data-Networks"
            set priority-members 4
        next
        edit 3
            set name "TUN7"
            set route-tag 7
            set src "Data-Networks"
            set priority-members 5
        next
        edit 4
            set name "TUN8"
            set route-tag 8
            set src "Data-Networks"
            set priority-members 6
        next
    end
end
```

Branch 1 & 2 BGP Routing Verification after 8 Tunnels are Established

After establishing eight neighbors, the branches still have 6 paths learnt from each other. This limitation is on the number of routes HUB2 can advertise downstream to branches which is limited to 2.



As done for HUB1, configure HUB2 to advertise 4 best paths to branches. Each neighbor group is configured to advertise 4 paths downstream to each spoke.



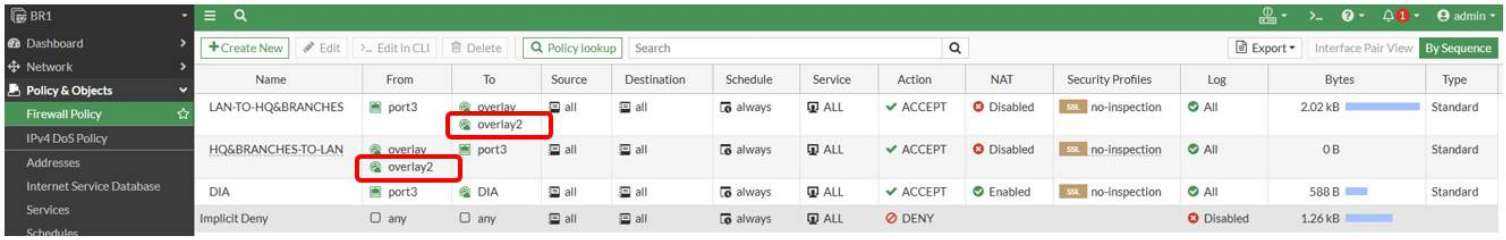
With the additional paths raised to 4, the spokes/branches now have 8 paths to each other over the available tunnels for better redundancy.

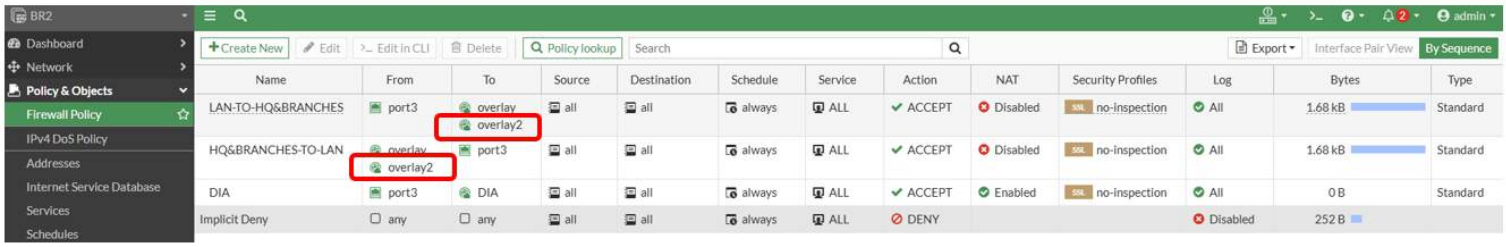
```
BR1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.2.0/24 [200/0] via 10.50.1.10 [4] (recursive is directly connected, HUB1-VPN1), 00:00:15, [1/0]
                     [200/0] via 10.50.2.10 [4] (recursive is directly connected, HUB1-VPN2), 00:00:15, [1/0]
                     [200/0] via 10.50.3.10 [4] (recursive is directly connected, HUB1-VPN3), 00:00:15, [1/0]
                     [200/0] via 10.50.4.10 [4] (recursive is directly connected, HUB1-VPN4), 00:00:15, [1/0]
                     [200/0] via 10.50.5.11 [4] (recursive is directly connected, HUB2-VPN5), 00:00:15, [1/0]
                     [200/0] via 10.50.6.10 [4] (recursive is directly connected, HUB2-VPN6), 00:00:15, [1/0]
                     [200/0] via 10.50.7.10 [4] (recursive is directly connected, HUB2-VPN7), 00:00:15, [1/0]
                     [200/0] via 10.50.8.10 [4] (recursive is directly connected, HUB2-VPN8), 00:00:15, [1/0]
B       10.10.100.0/24 [200/0] via 10.50.1.1 (recursive via HUB1-VPN1 tunnel 100.1.100.1), 00:19:56, [1/0]
                       [200/0] via 10.50.2.1 (recursive via HUB1-VPN2 tunnel 200.2.100.1), 00:19:56, [1/0]
                       [200/0] via 10.50.3.1 (recursive via HUB1-VPN3 tunnel 10.0.0.1), 00:19:56, [1/0]
                       [200/0] via 10.50.4.1 (recursive via HUB1-VPN4 tunnel 10.0.0.2), 00:19:56, [1/0]
B       10.10.200.0/24 [200/0] via 10.50.5.1 (recursive via HUB2-VPN5 tunnel 100.1.200.1), 00:00:43, [1/0]
                       [200/0] via 10.50.6.1 (recursive via HUB2-VPN6 tunnel 200.2.200.1), 00:00:43, [1/0]
                       [200/0] via 10.50.7.1 (recursive via HUB2-VPN7 tunnel 10.0.0.3), 00:00:43, [1/0]
                       [200/0] via 10.50.8.1 (recursive via HUB2-VPN8 tunnel 10.0.0.4), 00:00:43, [1/0]
```

```
BR2 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 10.50.1.11 [4] (recursive is directly connected, HUB1-VPN1), 00:00:06, [1/0]
                     [200/0] via 10.50.2.11 [4] (recursive is directly connected, HUB1-VPN2), 00:00:06, [1/0]
                     [200/0] via 10.50.3.11 [4] (recursive is directly connected, HUB1-VPN3), 00:00:06, [1/0]
                     [200/0] via 10.50.4.11 [4] (recursive is directly connected, HUB1-VPN4), 00:00:06, [1/0]
                     [200/0] via 10.50.5.10 [4] (recursive is directly connected, HUB2-VPN5), 00:00:06, [1/0]
                     [200/0] via 10.50.6.11 [4] (recursive is directly connected, HUB2-VPN6), 00:00:06, [1/0]
                     [200/0] via 10.50.7.11 [4] (recursive is directly connected, HUB2-VPN7), 00:00:06, [1/0]
                     [200/0] via 10.50.8.11 [4] (recursive is directly connected, HUB2-VPN8), 00:00:06, [1/0]
B       10.10.100.0/24 [200/0] via 10.50.1.1 (recursive via HUB1-VPN1 tunnel 100.1.100.1), 00:19:31, [1/0]
                       [200/0] via 10.50.2.1 (recursive via HUB1-VPN2 tunnel 200.2.100.1), 00:19:31, [1/0]
                       [200/0] via 10.50.3.1 (recursive via HUB1-VPN3 tunnel 10.0.0.1), 00:19:31, [1/0]
                       [200/0] via 10.50.4.1 (recursive via HUB1-VPN4 tunnel 10.0.0.2), 00:19:31, [1/0]
B       10.10.200.0/24 [200/0] via 10.50.5.1 (recursive via HUB2-VPN5 tunnel 100.1.200.1), 00:00:34, [1/0]
                       [200/0] via 10.50.6.1 (recursive via HUB2-VPN6 tunnel 200.2.200.1), 00:00:34, [1/0]
                       [200/0] via 10.50.7.1 (recursive via HUB2-VPN7 tunnel 10.0.0.3), 00:00:34, [1/0]
                       [200/0] via 10.50.8.1 (recursive via HUB2-VPN8 tunnel 10.0.0.4), 00:00:34, [1/0]
```

Branch 1 & 2 Firewall Policies for HUB2

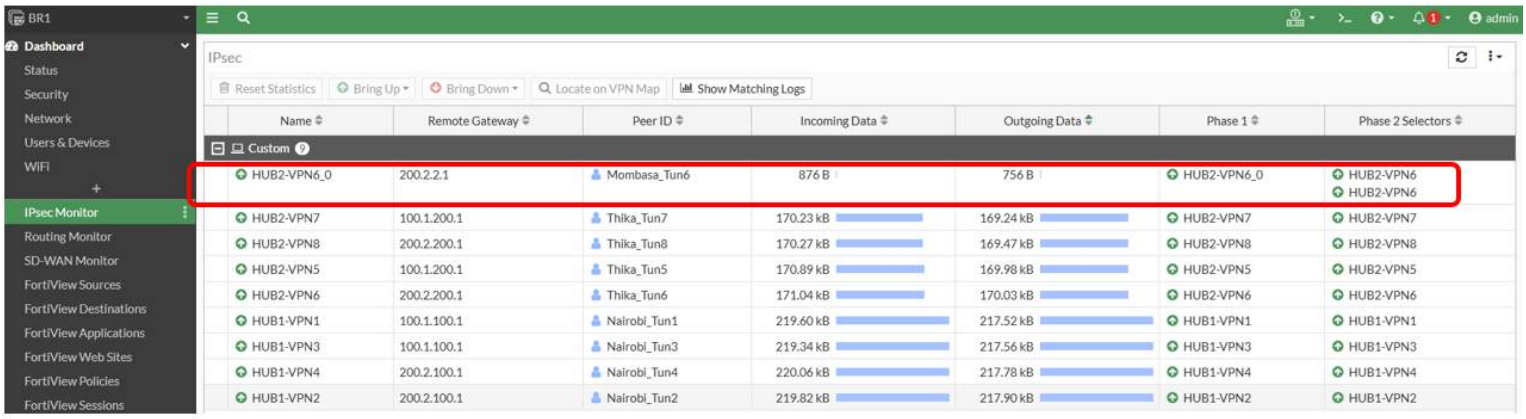
The new sdwan zone are added to the policies that use the overlay tunnels. Separate policies could be configured for overlay2 or Mult interface policies created as shown below.

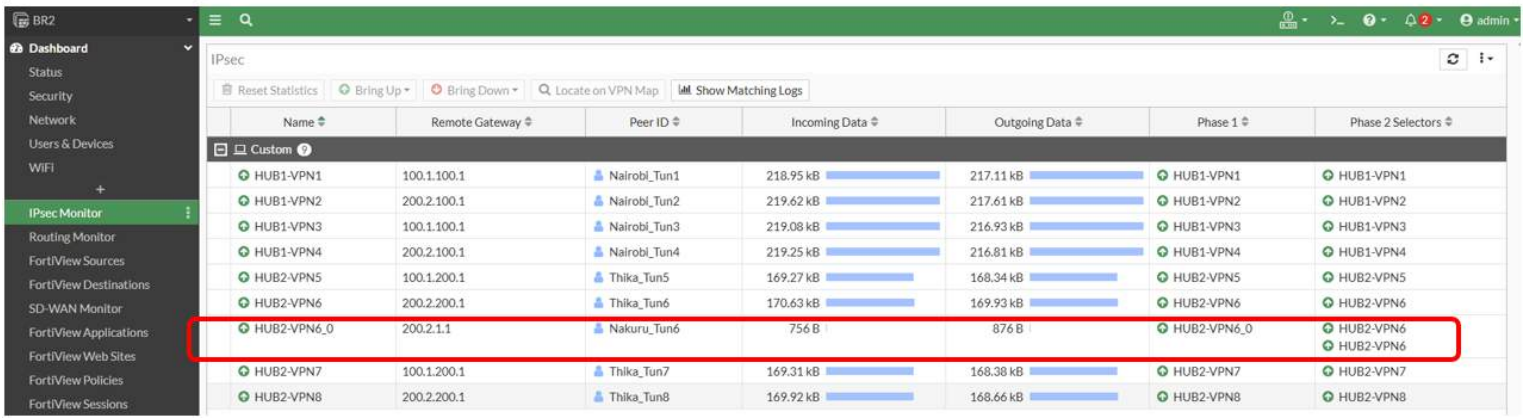




ADVPN with 8 Tunnels

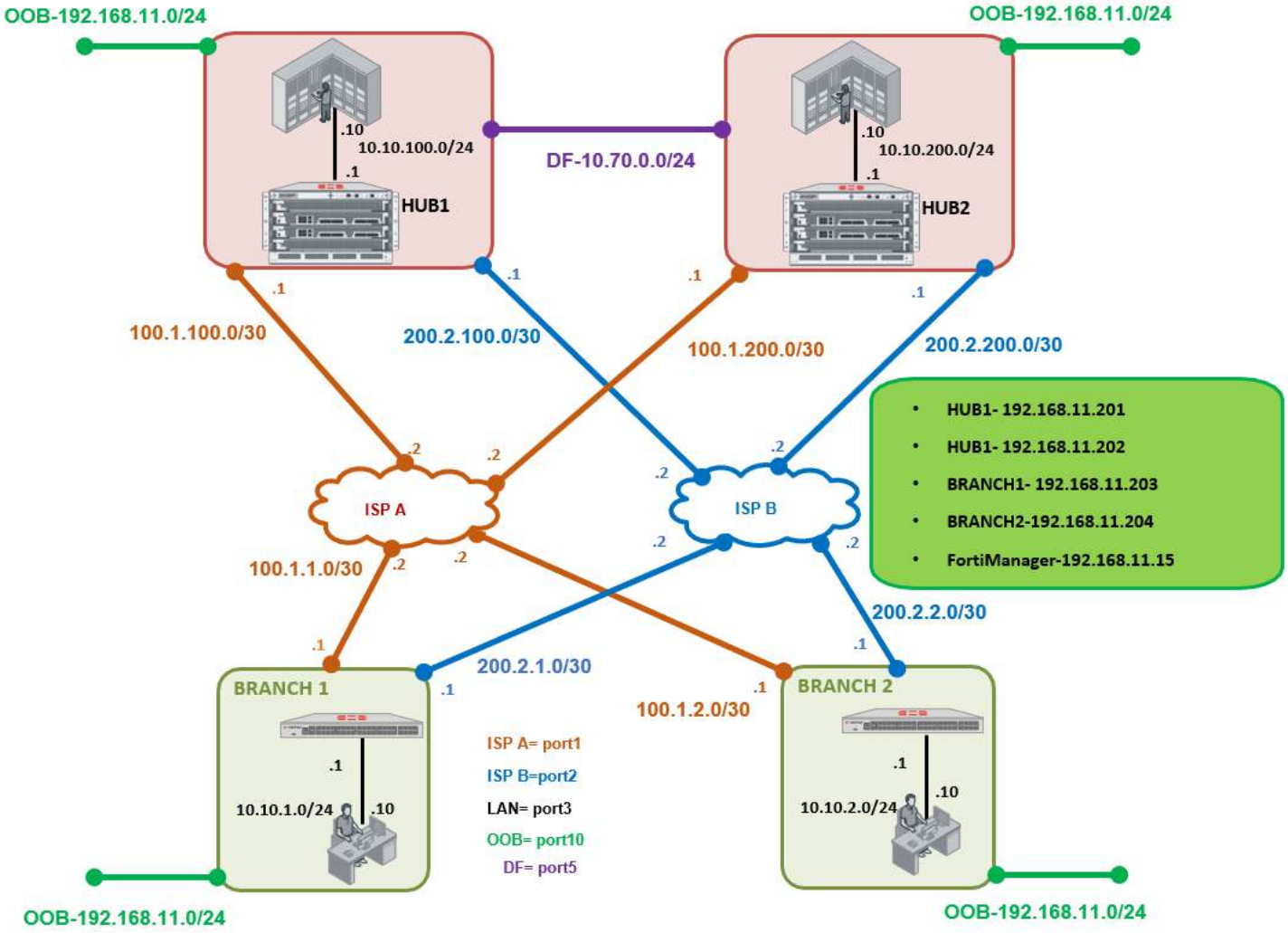
The configurations done at IPSec and BGP levels were ADVPN aware, branch to branch traffic will activate an ADVPN session, in the case shown below over HUB2-VPN6.



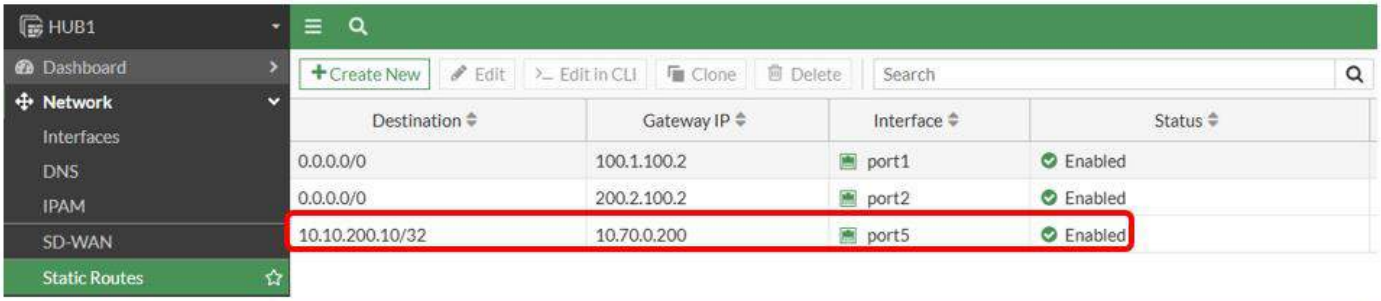


HUB1 TO HUB2 Dark Fiber/Private Circuit

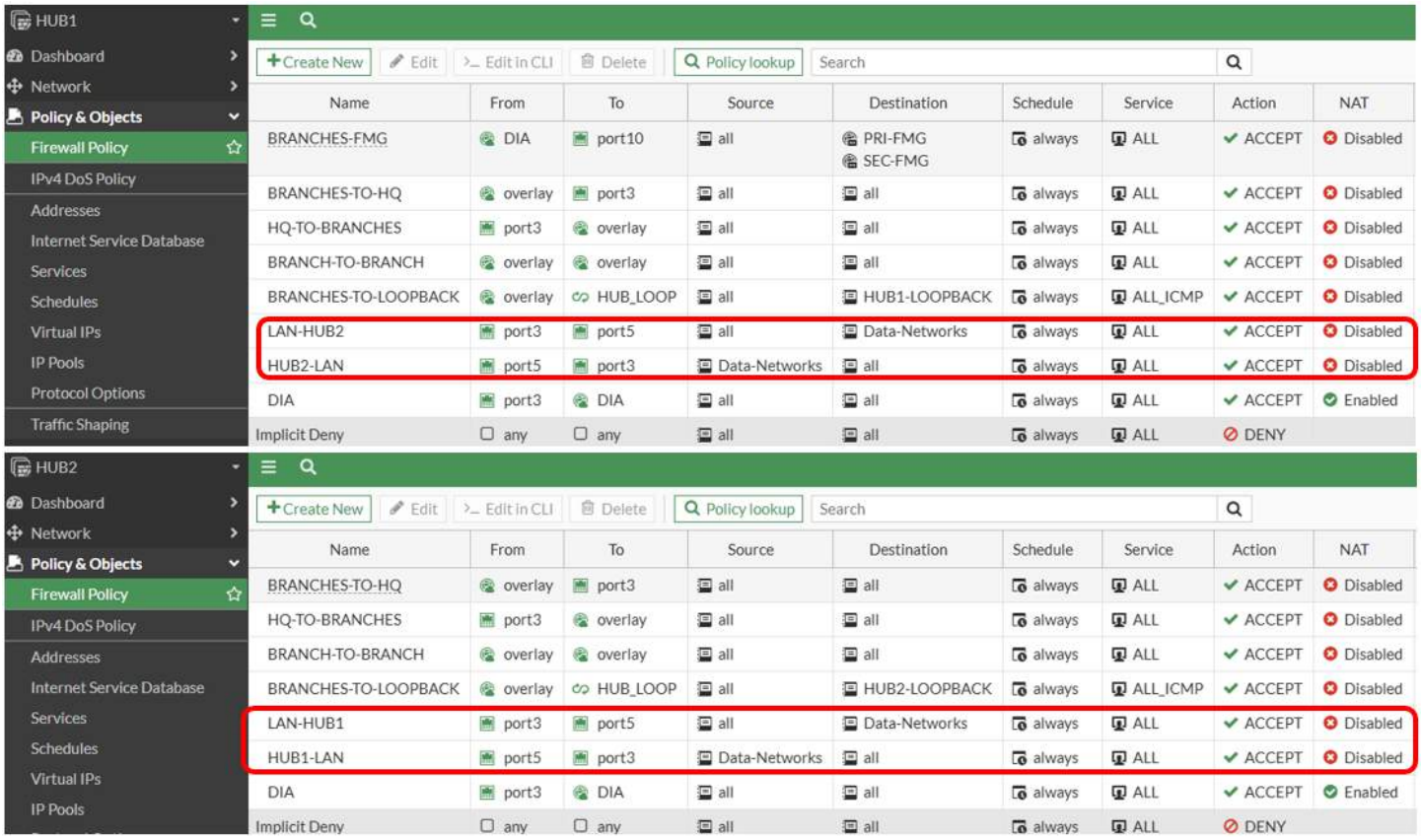
The easiest way to interconnect the HUBs is to use a private circuit between them. This could be MPLS, Dark Fiber or other layer2 circuits provided by the ISPs



Static routes are configured to route traffic over the direct path between the HUBs. If multiple networks exist between them, a dynamic routing protocol is recommended, port5 is used on both HUBs as the interconnecting link for the LAB.

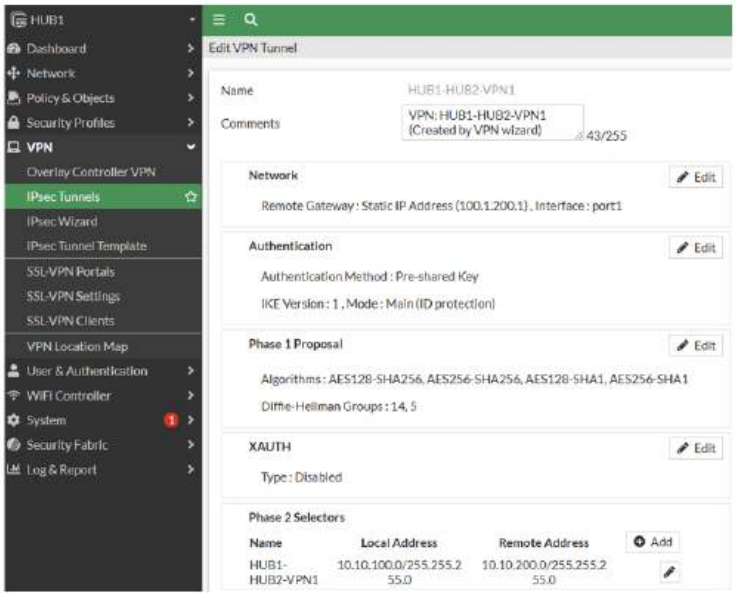


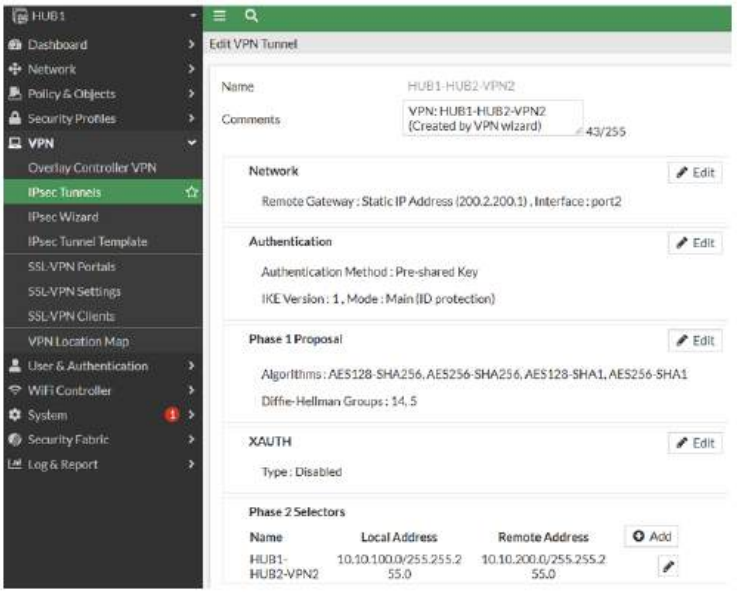
Firewall policies are configured between the sites to allow traffic flow between the them.

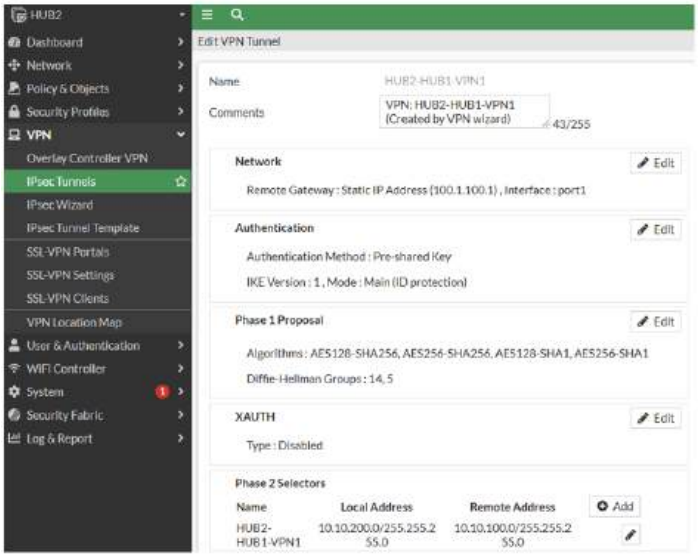


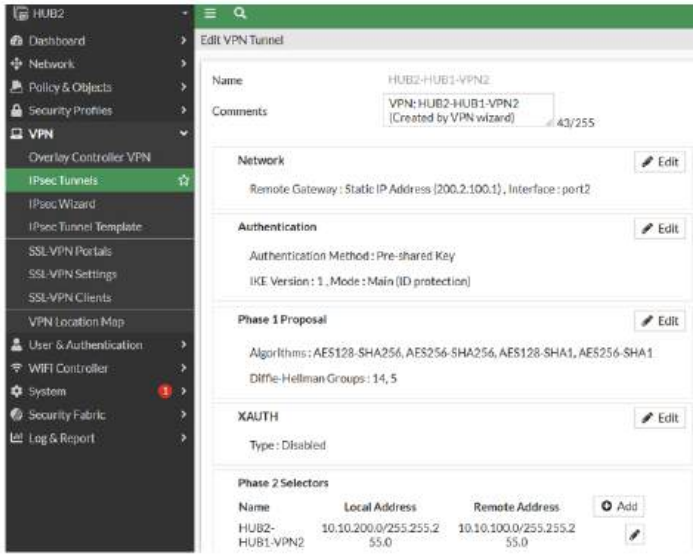
HUB1 to HUB2 Connection via IPSEC

Two Site to Site IPsec tunnels are configured to connect the HUBs. Interesting traffic is identified to define the phase2 parameters.

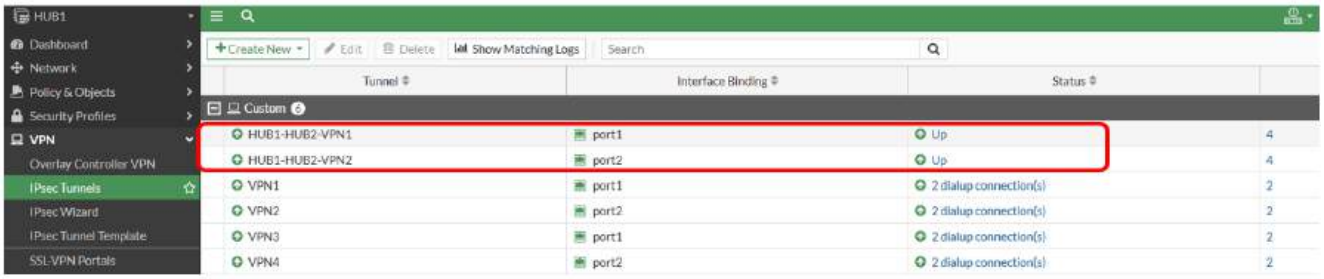


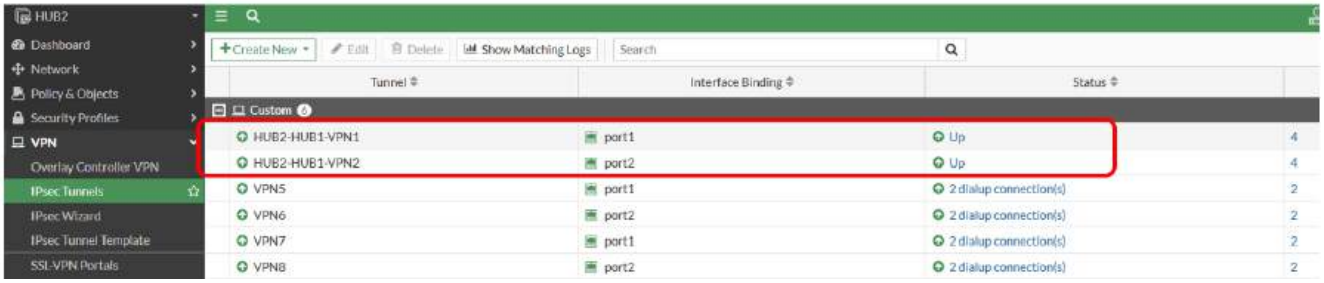




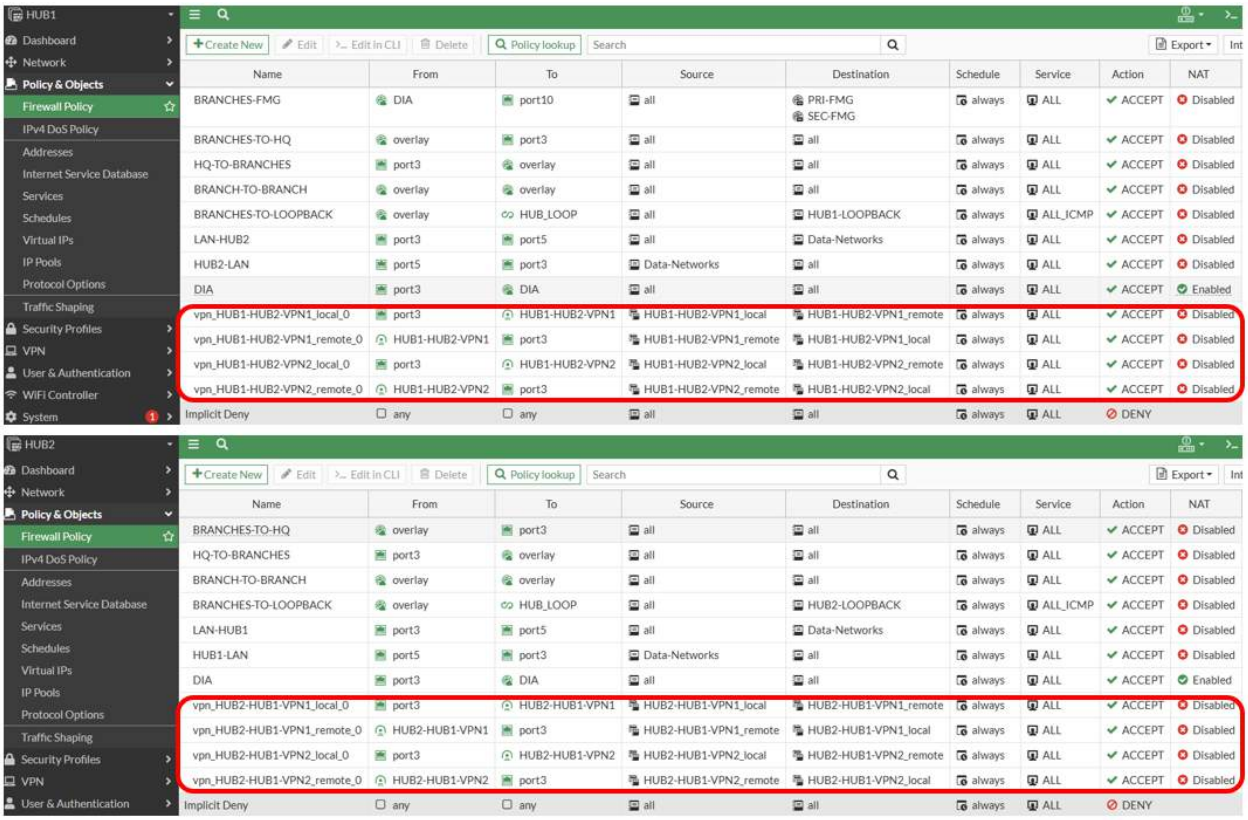


Two tunnels are configured between the HUBs for simplicity but more could be established for redundancy.

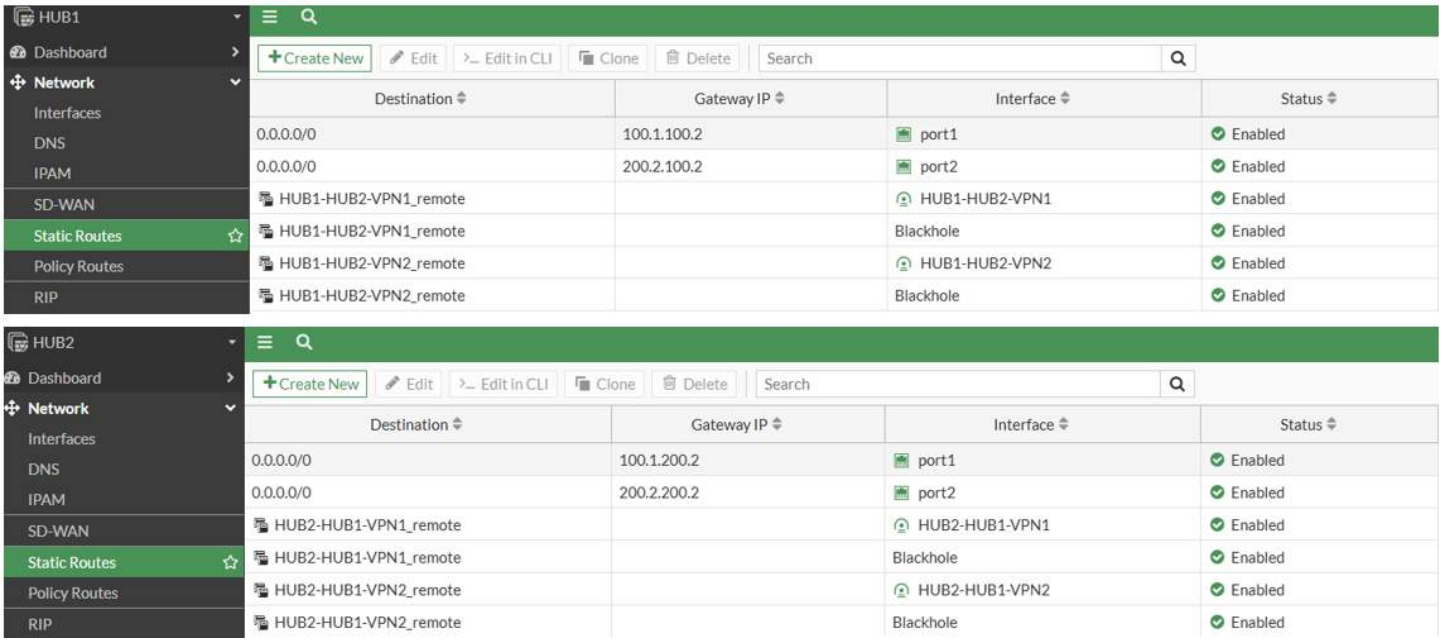




Firewall policies are then configured between the sites to allow traffic flow between the sites.



Static routing is used between the sites to direct traffic to the tunnels.



Chapter 4 Questions

1. After configuring 4 redundant paths to HUB1 and HUB2 each branch only learns two paths while the HUB has four paths. What is the likely issue if the deployment is done by FortiManager.

    a) IBGP is limited to allow 2 paths
    b) The BGP template at the HUB only allows advertising 2 paths to branches by default
    c) The Branch is limited to learning 2 paths only

2. When HUB2 is created, what needs to be unique at the branch for the new tunnels to be established.
    a) Local_ID
    b) Branch_ID
    c) Network_ID

3. Once two Hubs are configured i.e. HUB1 and HUB2, with two ISP links at all sites and redundant cross-connect links, how many tunnels are available per branch
    a) 4
    b) 6
    c) 8
4. After configuring the new loopback interface for HUB2, it was not available under Policy and Objects for firewall rules configuration. What is required to resolve this in FortiManager.

    a) Interface normalization
    b) Metadata variable definition.
    c) IP address assignment to the configured loopback interface.

5. After configuring IPsec tunnels and BGP, it is observed that redundant routes are learnt between branches and the HUB. However, traffic could only traverse between branches to HUBs and vice versa, branch to branch traffic is not successful. What is required to facilitate this?

    a) Outbound policy for branch-to-branch traffic at the branches.
    b) Inbound policy for branch-to-branch traffic at the branches
    c) Overlay-to-overlay rule for branch prefixes at the HUB.
    d) All the above.

Chapter 5: SD-WAN overlay Orchestration

The SD-WAN overlay template is a wizard that automates and simplifies SD-WAN provisioning. It creates the necessary **IPsec, BGP and CLI** templates required for the creation of SD-WAN overlays. Different templates are created for the HUBs and branches reducing the time required to setup the solution. **SD-WAN Templates** are not part of the provisioning process and must be configured manually.

To use the overlay template, the following requirements should be met:

i. The HUB and branch devices must be imported and registered in FortiManager.
ii. The underlay interfaces must be configured; the respective IP addresses are used in the provisioning process.
iii. Branches must be attached to a device group which will be referenced in the wizard.
iv. Allocate an overlay network address space. Defaults to 10.10.0.0/16.
v. Allocate an address space for loopback interfaces. Defaults to 172.16.0.0/16.
vi. The iBGP AS number to use. Defaults to 65000.
vii. The **branch_id** Metadata variable is automatically created by the template. Assignment of ids can happen automatically if selected within the Wizard or manually for granular control.

The Overlay Template does not create any Policy package for the devices. Firewall policies should be configured and assigned to the HUB and branches accordingly.

Overlay Templates support the following Topologies:

- Single HUB
- Dual HUB (Primary/Secondary)
- Dual HUB (Primary/Primary)
- Multi HUB-Support 3 or 4 hubs

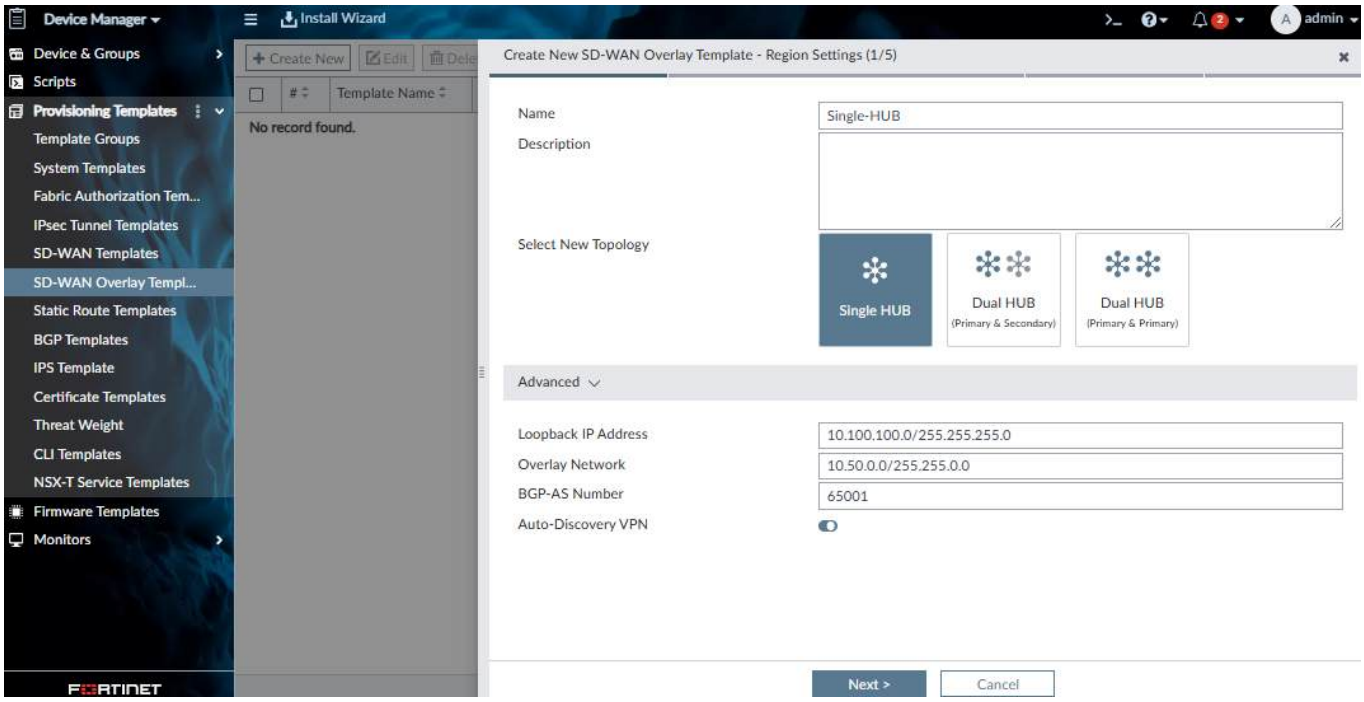
Primary/Secondary and Primary/Primary Dual HUB options are essentially the same configuration. In a Primary/Secondary deployment, the Secondary hub is given a higher cost than the Primary while in the Primary/Primary they have the same cost. The cost is controlled by the SDWAN rules at the branches.

It should be noted that the overlay Template simply expedites provisioning of required **CLI,BGP& IPsec** Templates but extra customization using the knowledge in previous chapters is required to have a functional setup.

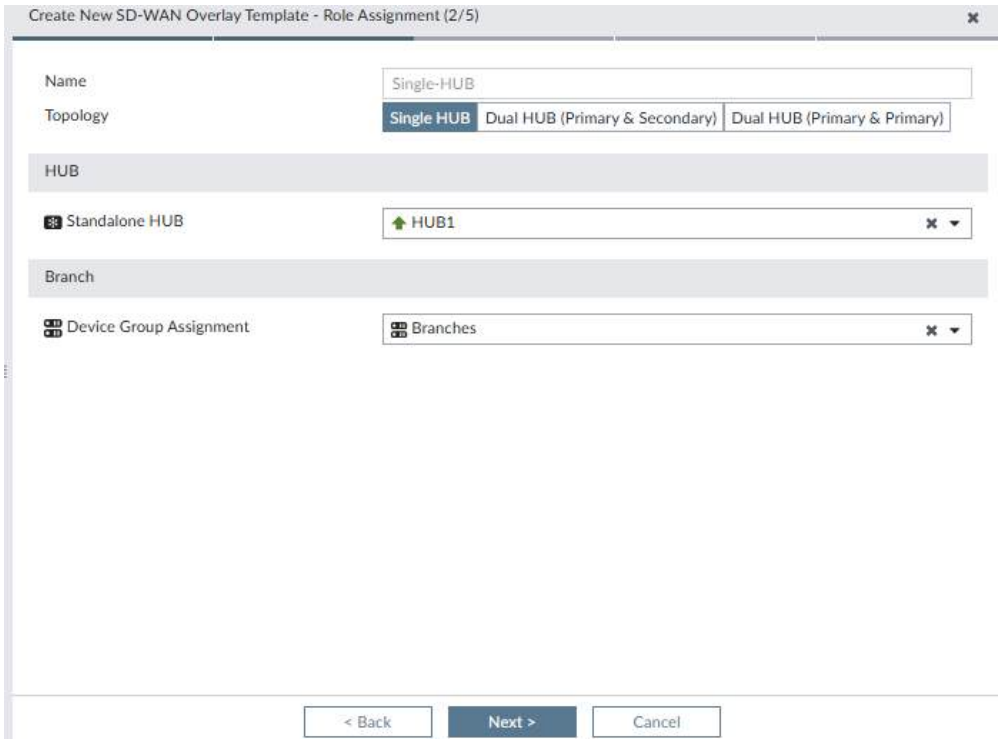
The LAB demos will include single-hub and dual-hub scenarios with customization to match the topology used in earlier demos.

SDWAN OVERLAY Templates- SINGLE HUB

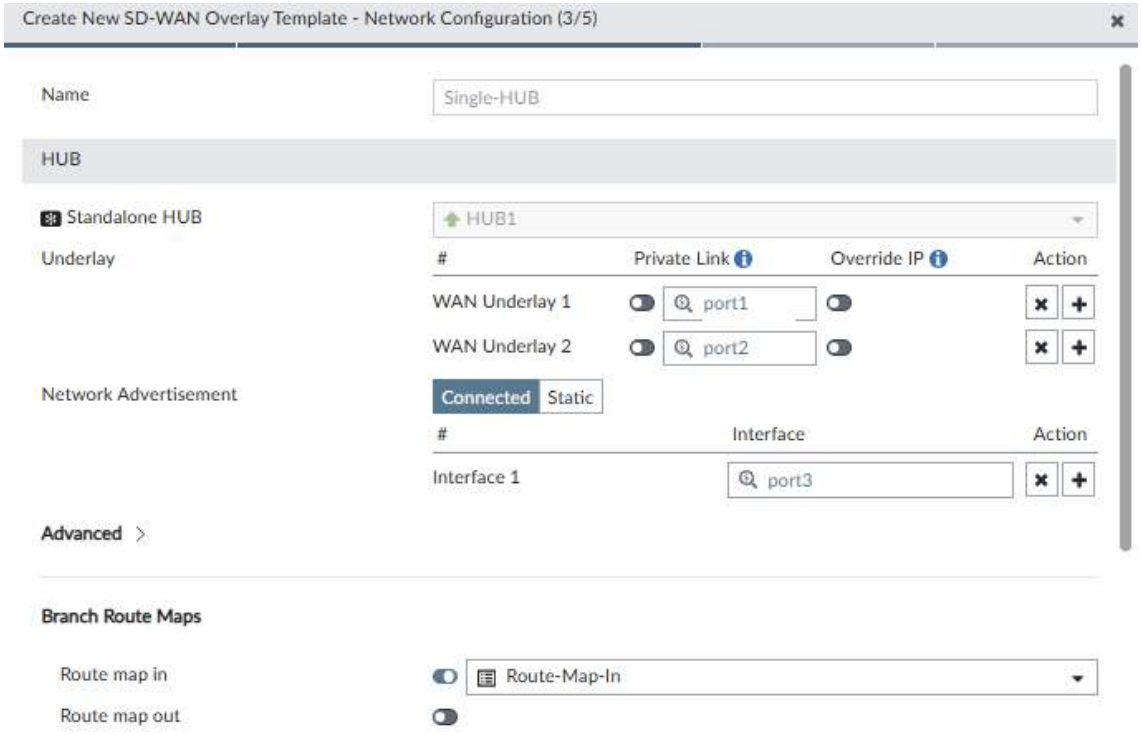
Start by selecting the appropriate topology (Single-Hub) and specify the loopback ip addresses, overlay network, AS number and activate ADVPN.



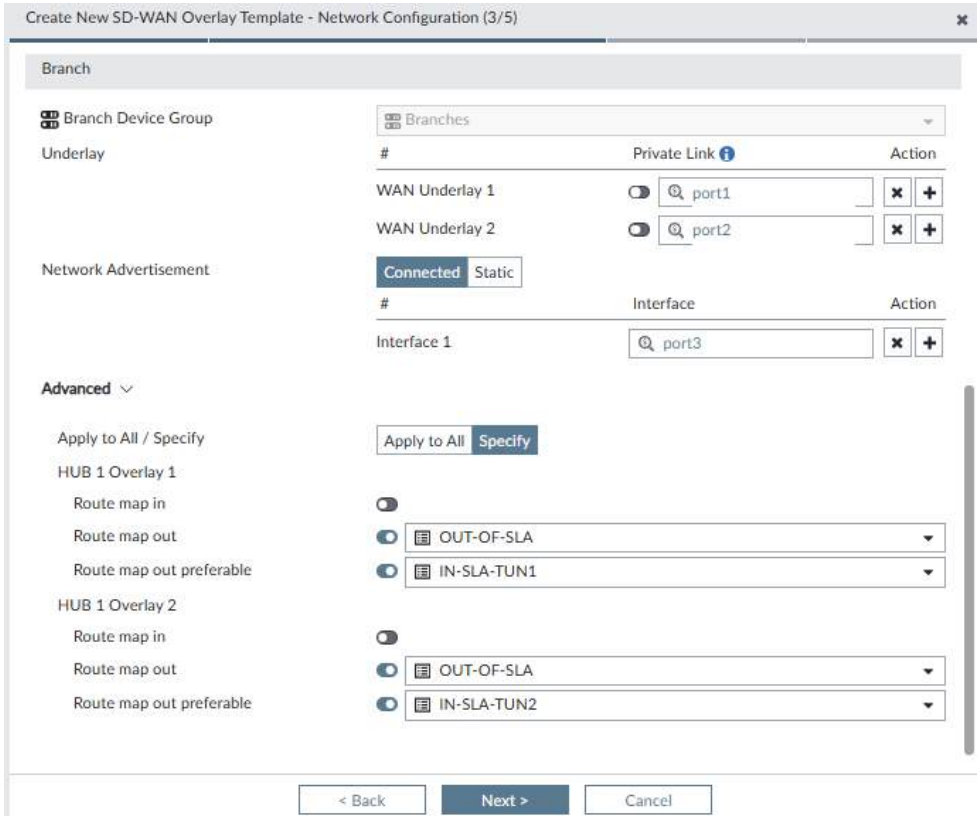
For the single-hub, HUB1 and the branch group are attached.



Port assignment for the underlay networks, LAN and assignment of route-maps is done at stage 3. For the HUB, Rout-map-in is configured. The template allows inline configuration of route-maps. It is recommended to create empty route-maps and customize then once the wizard is complete.

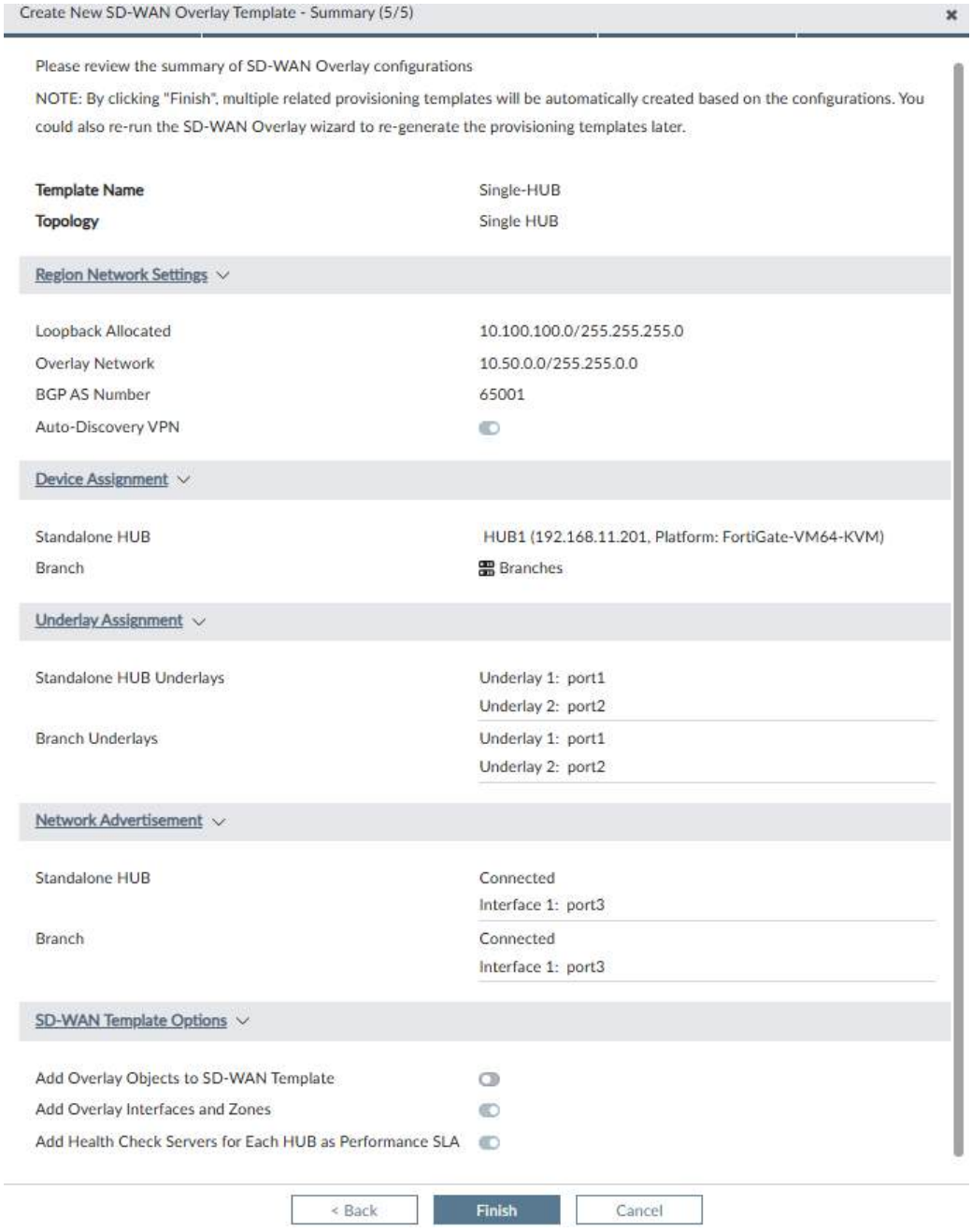


For the Branches, create route-map-out for out of SLA state for all branches and custom route maps for compliant state.

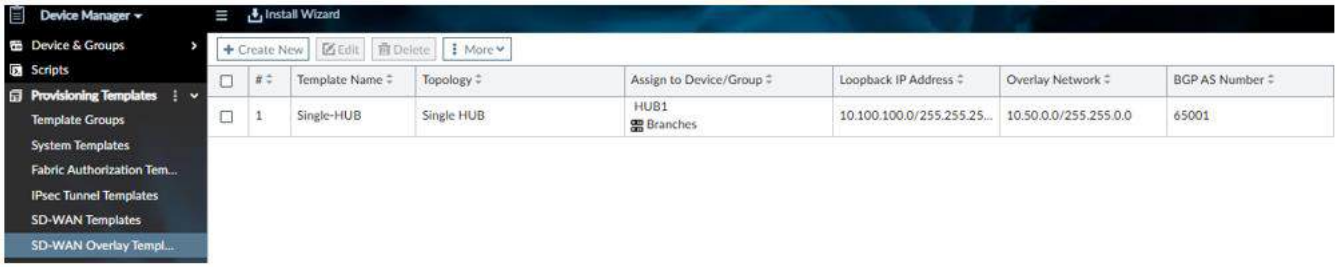


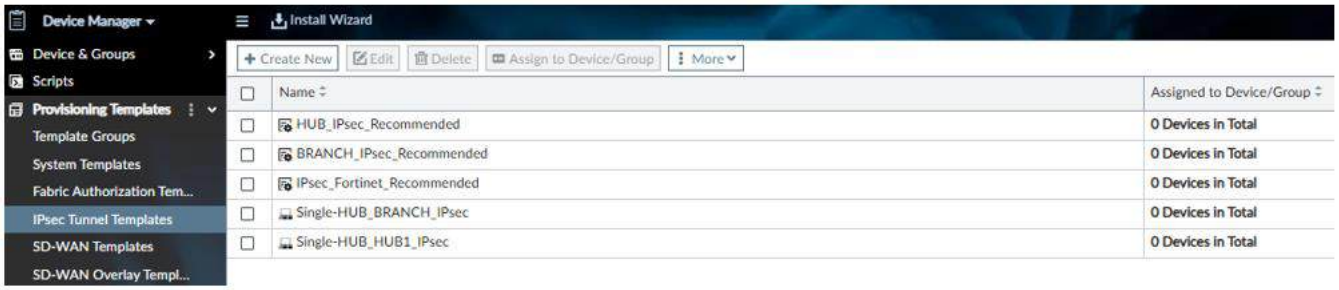
Finally add overlay interfaces and zones.

The summary of the settings done is displayed below:

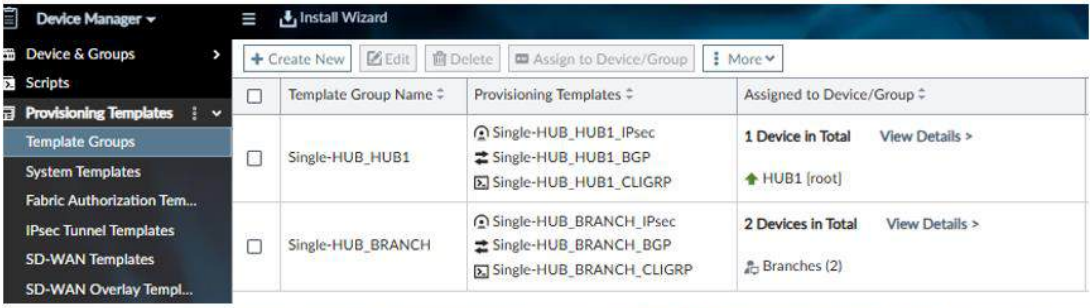


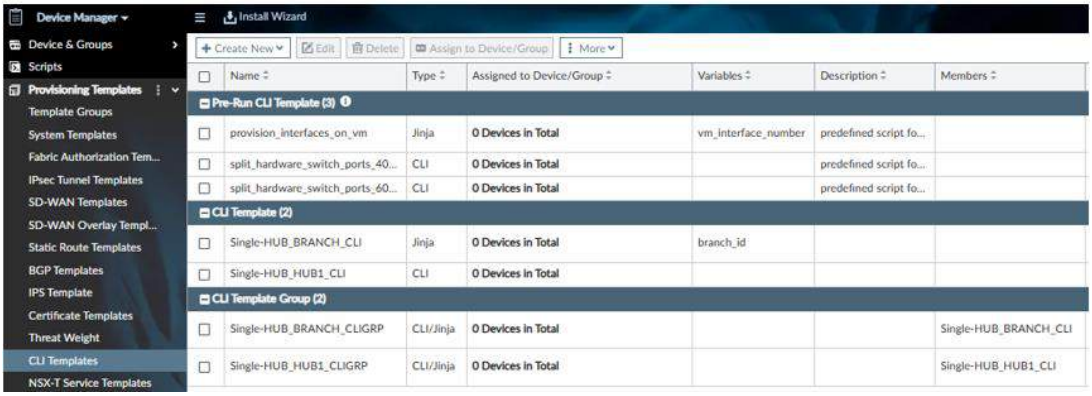
The wizard creates the IPsec and BGP templates for the branches and HUB



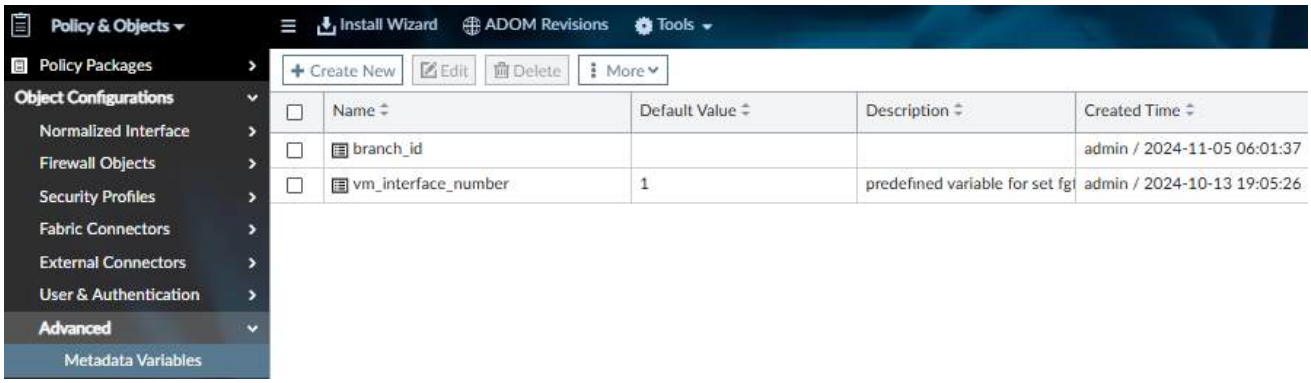


It also creates Template Groups and CLI templates:

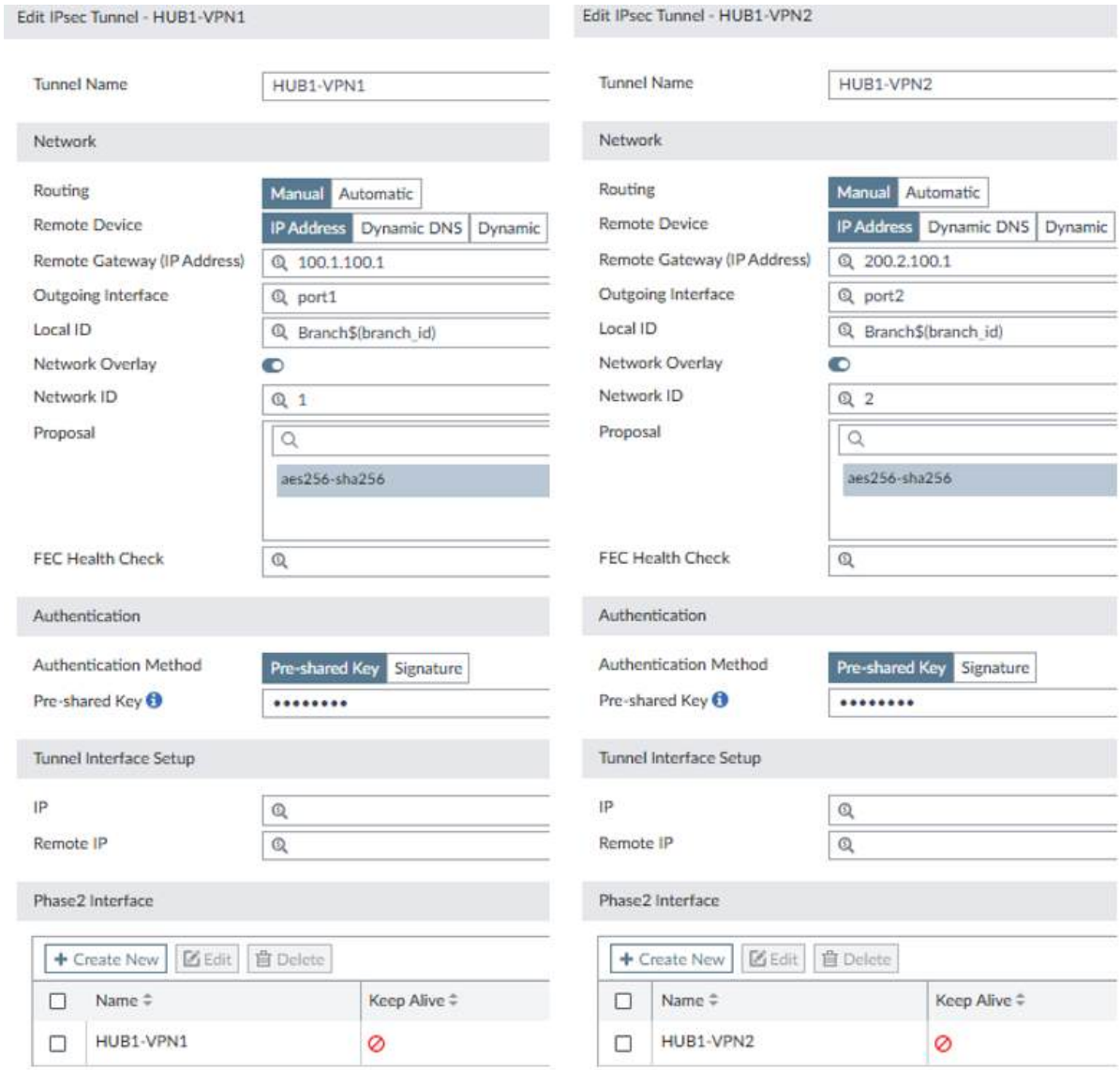




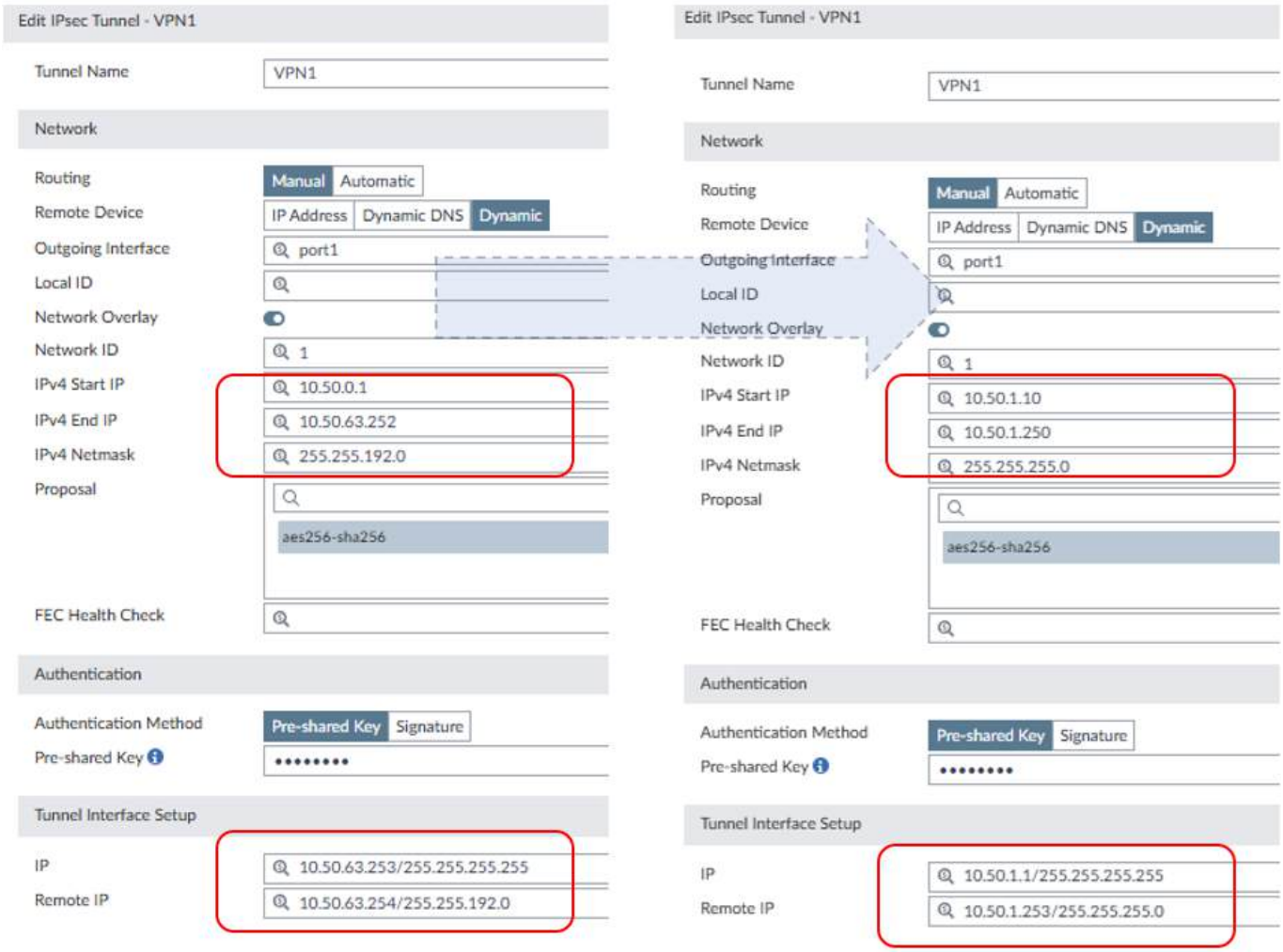
Under policies and objects, a metadata variable of branch_id is created.

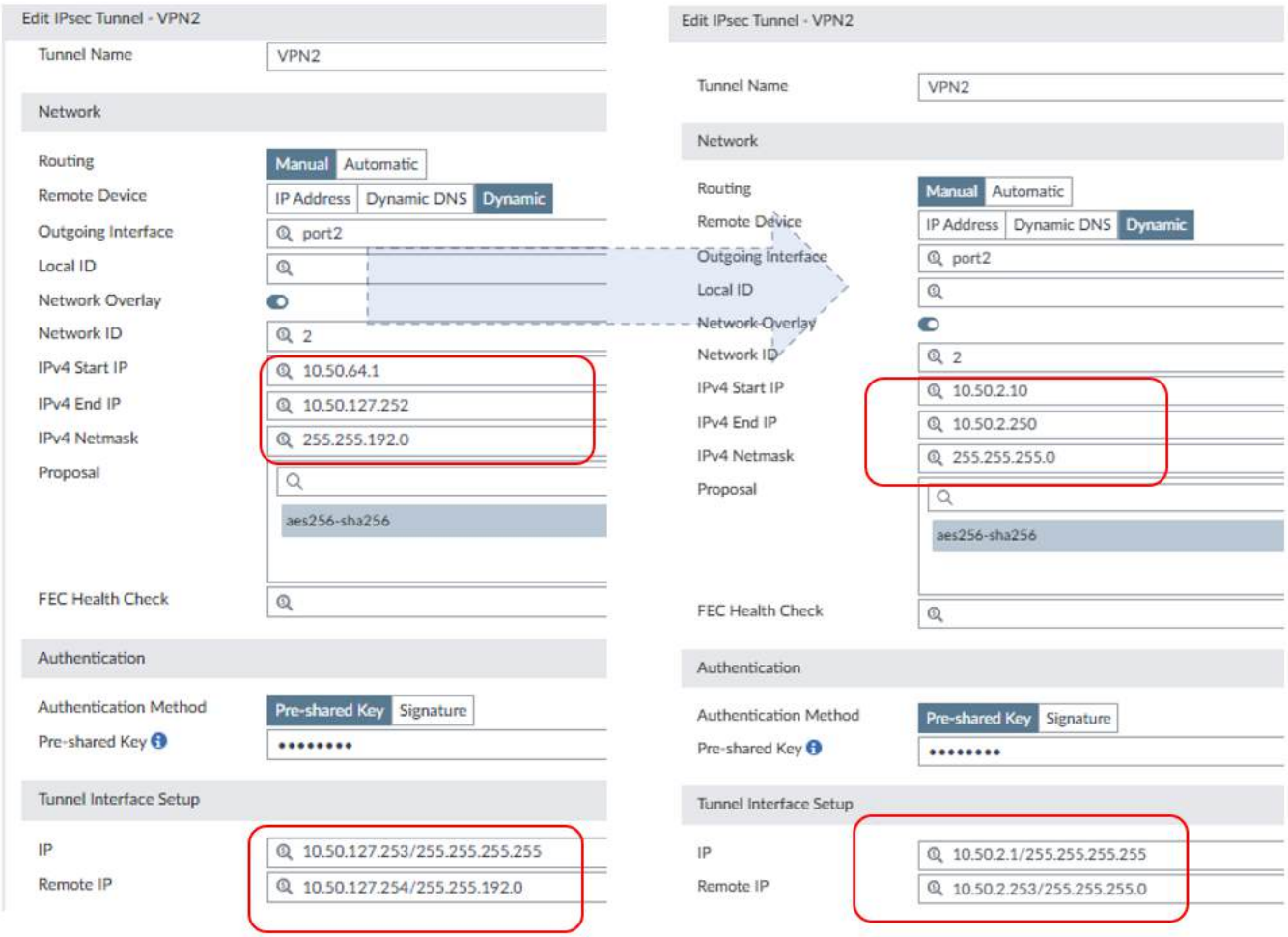


For Branch IPsec, HUB1-VPN1 and HUB1-VPN2 are created with the remote gateway ip address automatically captured from the HUB underlay interfaces. If you require additional tunnels for redundancy, they must be configured manually

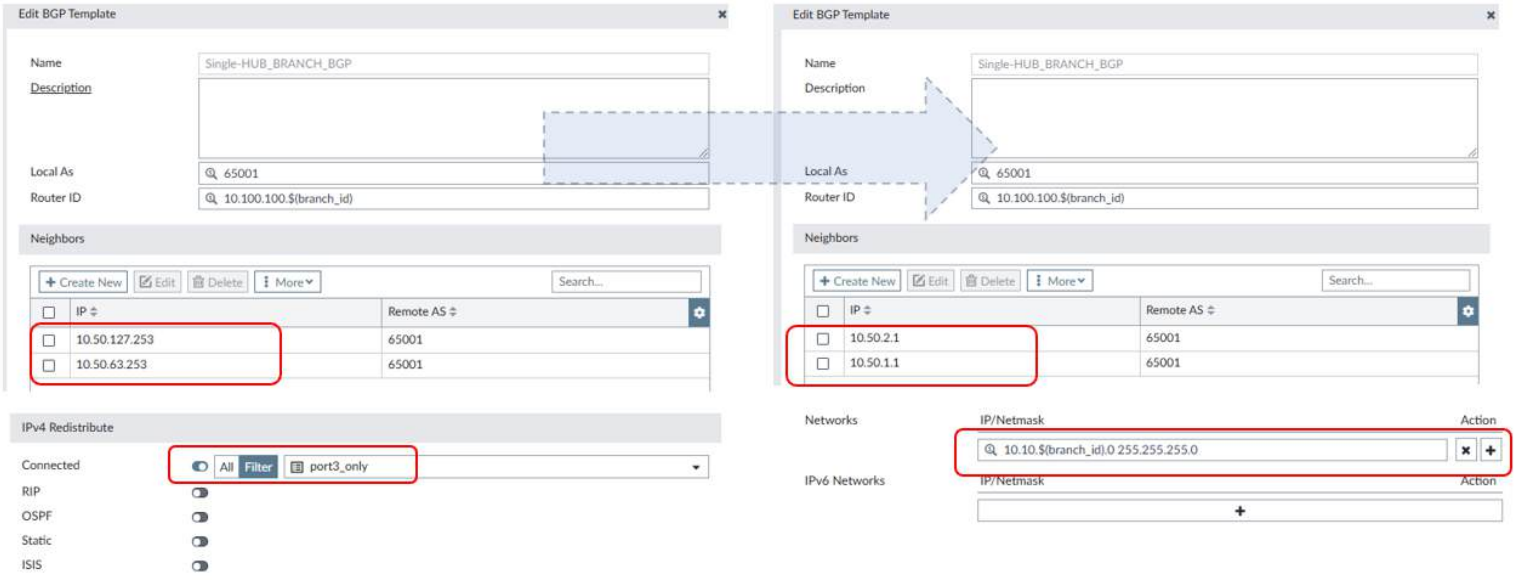


The HUB IPsec VPN interfaces are customized from the defaults to match with the settings at the branch.

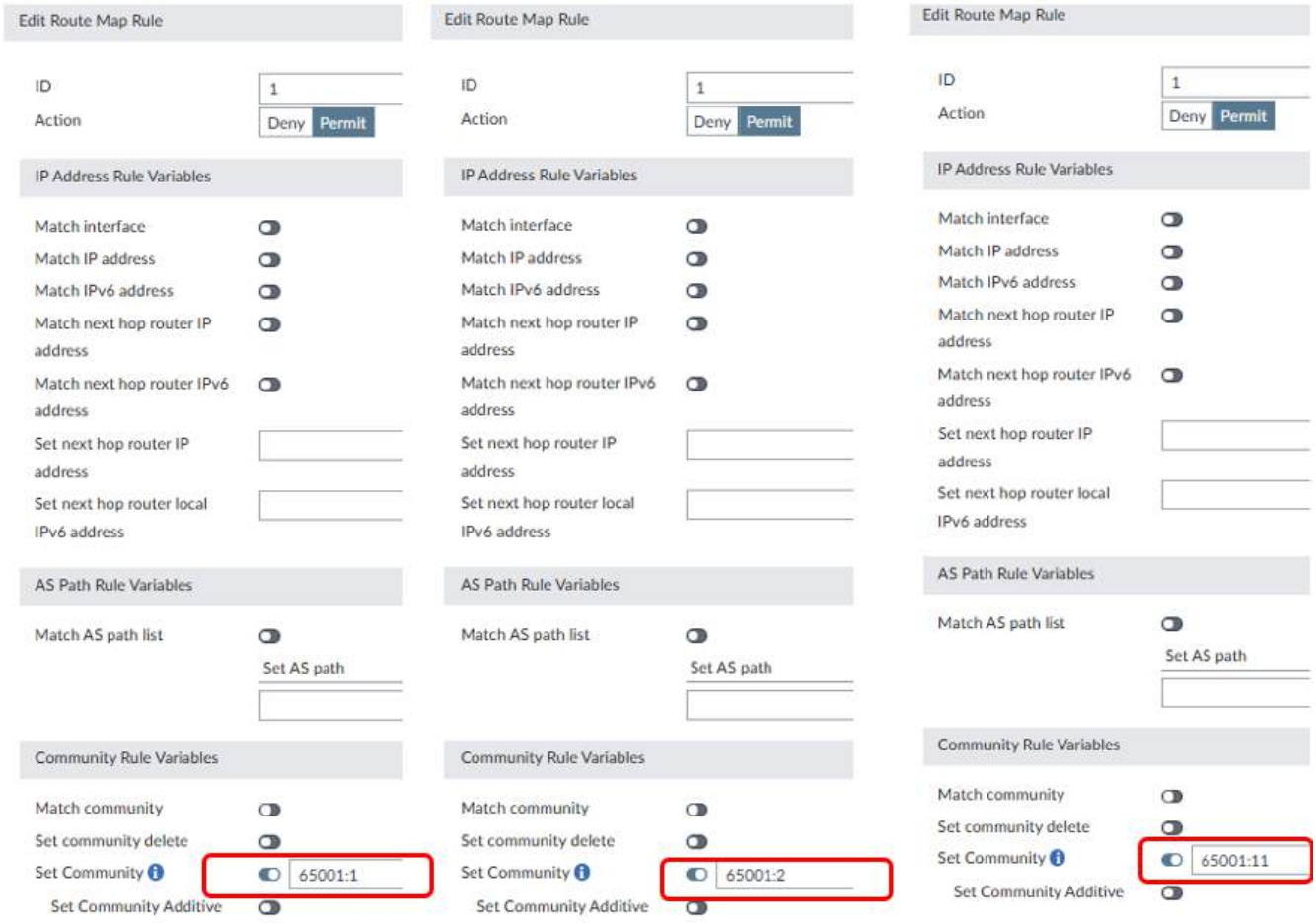




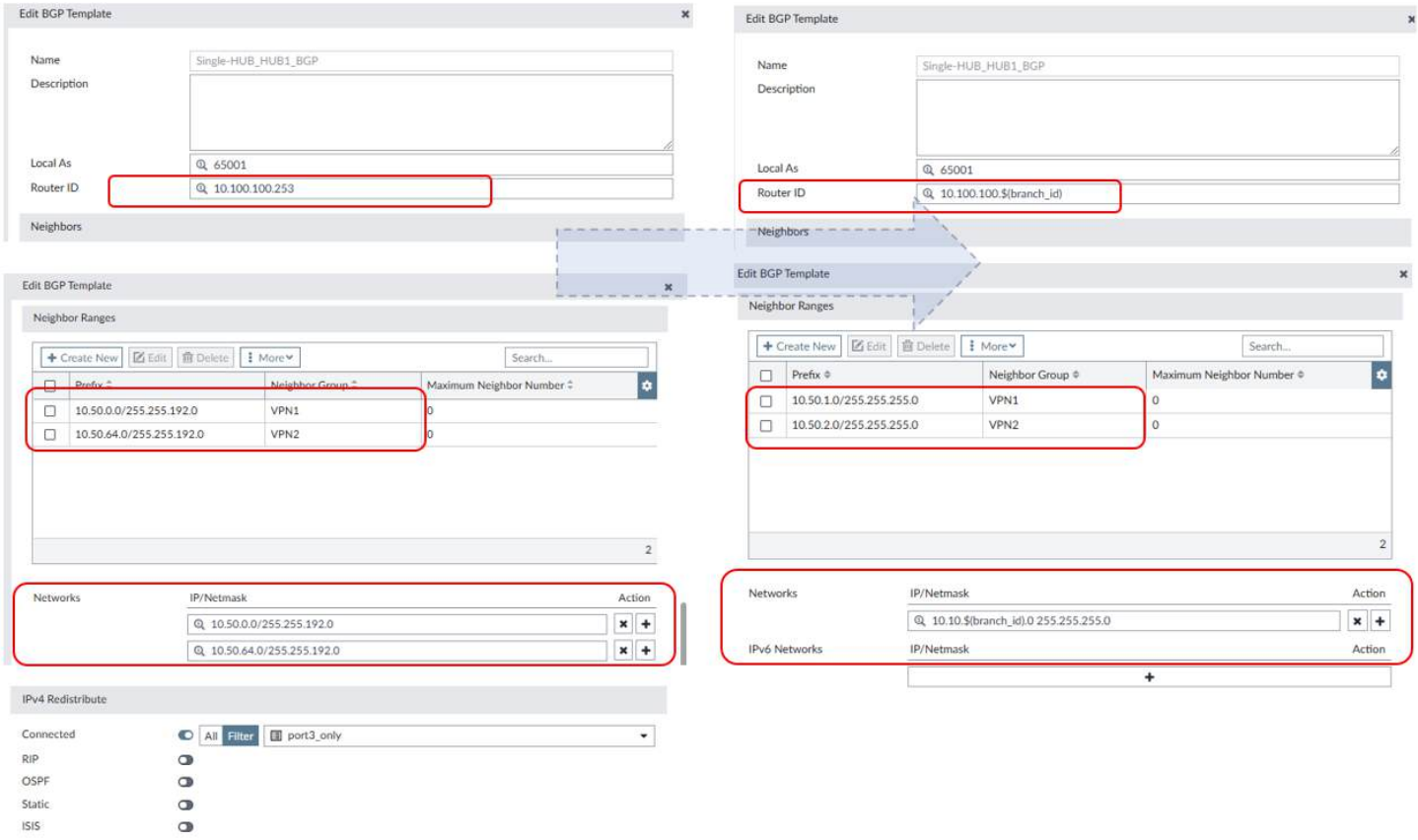
For the Branches, BGP neighbors are created randomly from the Wizard assigned scope, these should be customized to the required ip addresses. The routing is also changed to match LAN networks and redistribution disable. This is not mandatory but it is implemented to match earlier setups and give granular control compared to random assignment.



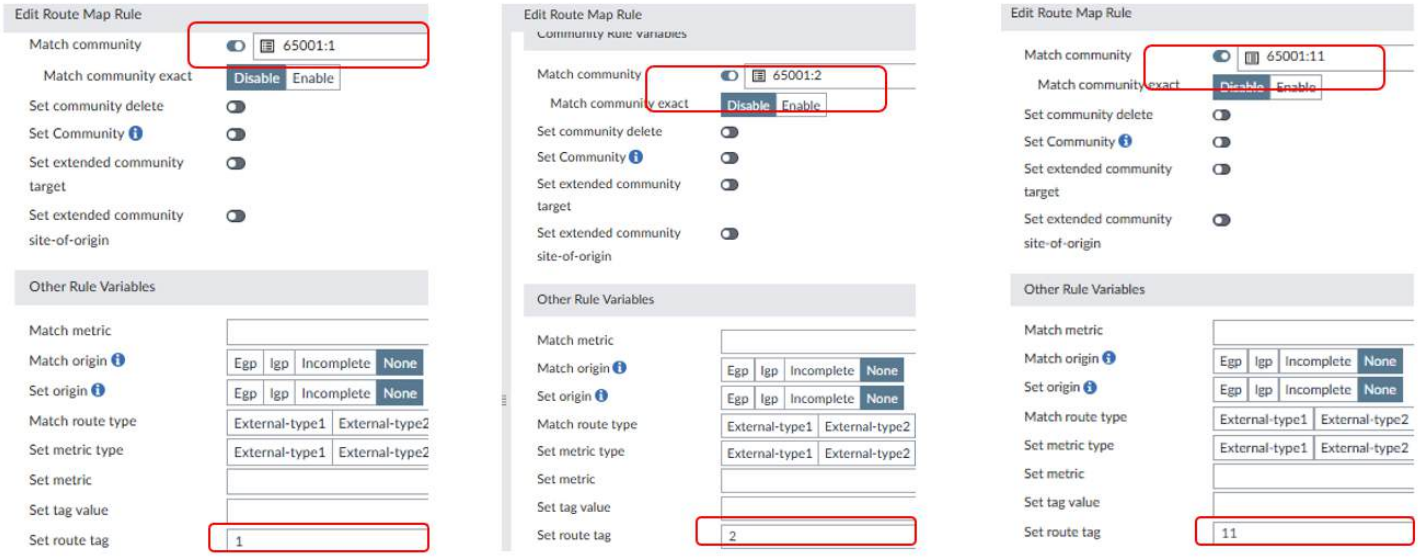
The empty route-maps configured with the Wizard are populated to set required communities at the branch.



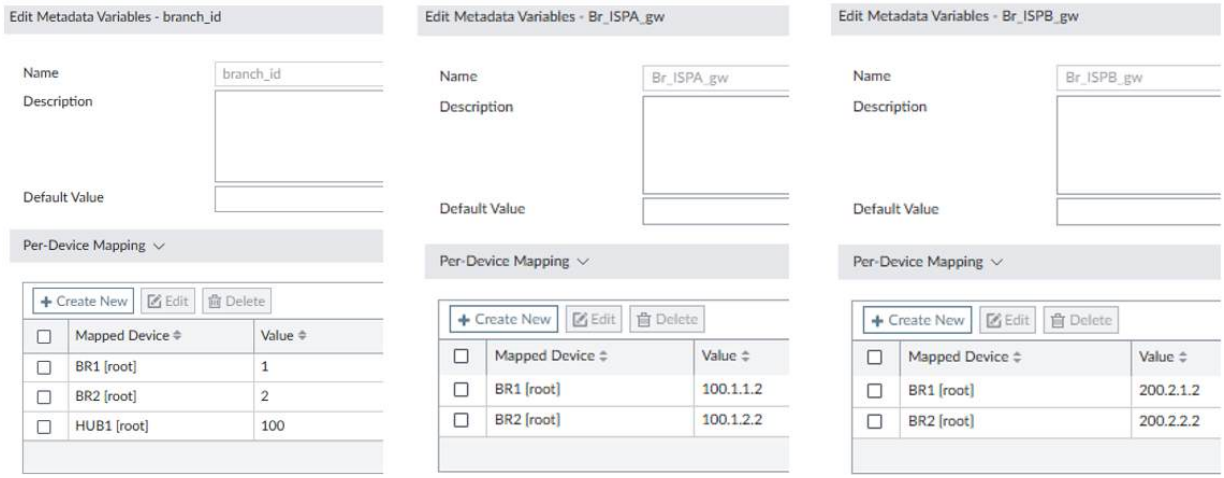
At the HUB, customize the Router ID and neighbor ranges to fit the design from the initial randomly generated scopes.



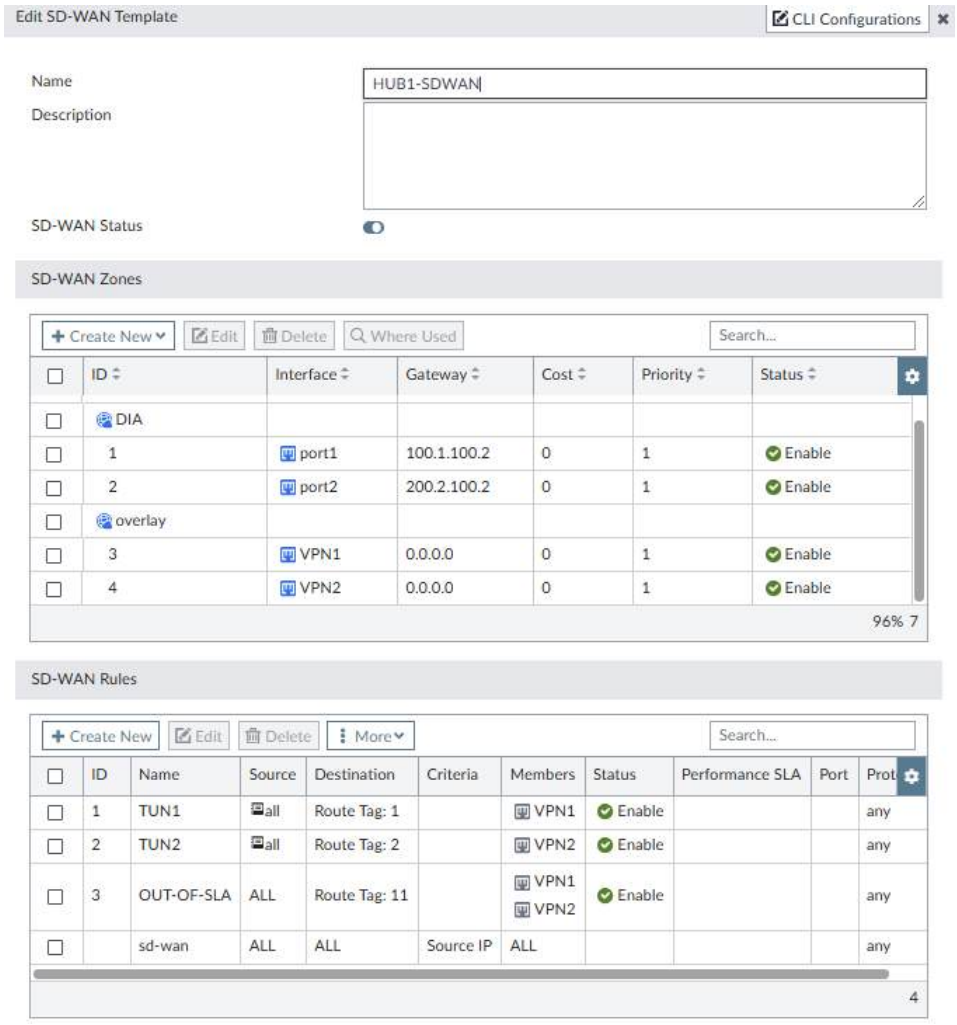
The route-map at the HUB is configured to match incoming communities and set route tags based on the diagram below:



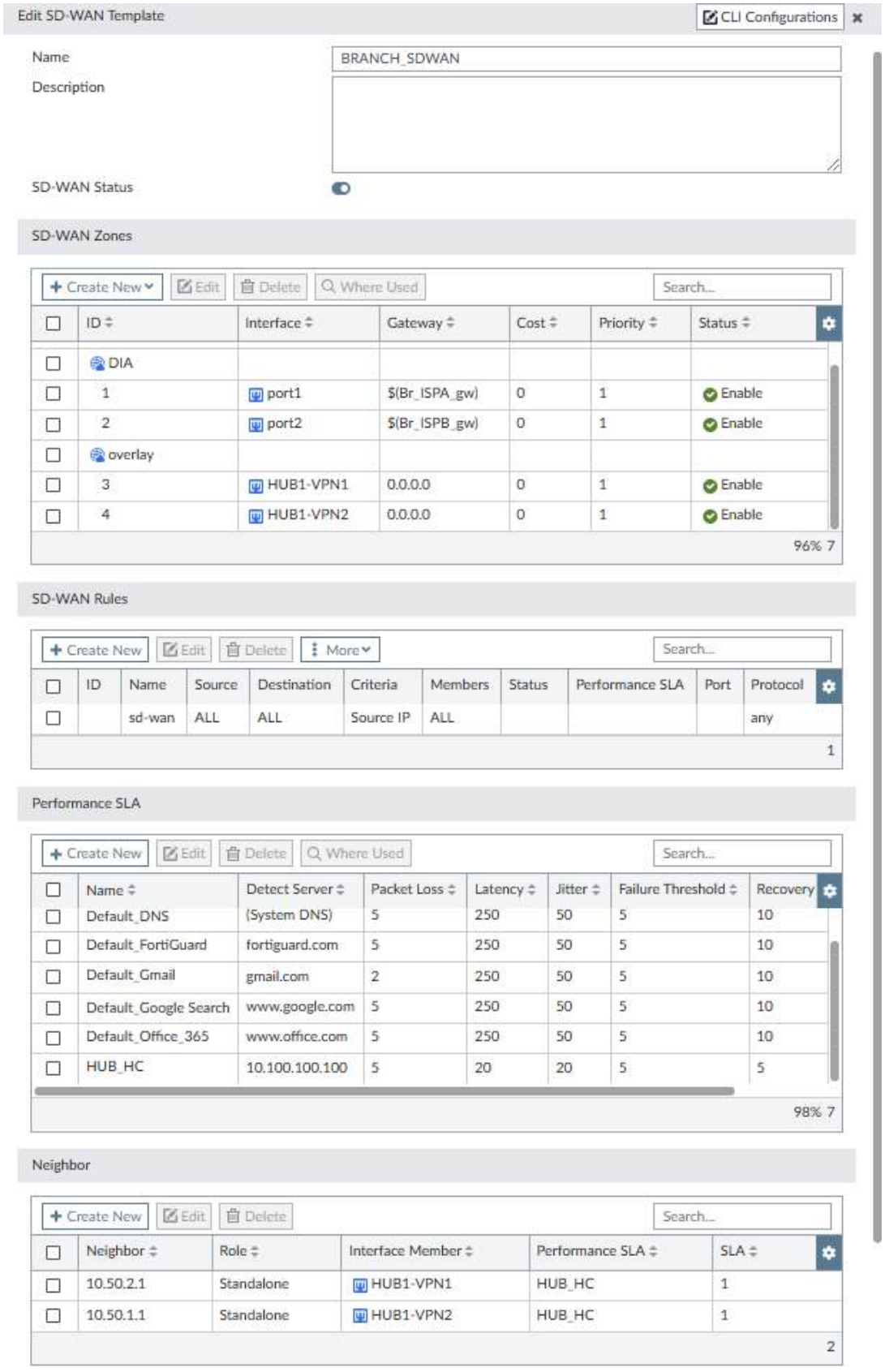
The Metadata Variables are configured as shown below:



The overlay template doesn't create any SDWAN template. New SD-WAN templates are configured for the HUB and Branches.



When defining the underlay (DIA) interfaces as SD-WAN members, add the gateways to avoid losing the default routes and remote access to the devices.



Branch Configuration Preview

```
config vpn ipsec phase1-interface
    edit "HUB1-VPN1"
        set interface "port1"
        set ike-version 2
        set comments "VPN: HUB1-VPN1 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set localid "Branch1"
        set remote-gw 100.1.100.1
        set idle-timeout enable
        set net-device enable
        set add-route disable
        set auto-discovery-receiver enable
        set psksecret ENC Z8Zpc/bwU2j1HxCFWzO/XkWz1iP/WK4qAGVHE9oazICl
        set network-overlay enable
        set network-id 1
        set auto-discovery-shortcuts dependent
    next
end
config system interface
    edit "HUB1-VPN1"
        set vdom "root"
        set type tunnel
        set snmp-index 116
        set interface "port1"
    next
end
config vpn ipsec phase1-interface
    edit "HUB1-VPN2"
        set interface "port2"
        set ike-version 2
        set comments "VPN: HUB1-VPN2 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set localid "Branch1"
        set remote-gw 200.2.100.1
        set idle-timeout enable
        set net-device enable
        set add-route disable
        set auto-discovery-receiver enable
        set psksecret ENC Z8Zpc/bwU2j1HxCFWzO/XkWz1iP/WK4qAGVHE9oazICl
        set network-overlay enable
        set network-id 2
        set auto-discovery-shortcuts dependent
    next
end
```

```
config system interface
    edit "HUB1-VPN2"
        set vdom "root"
        set allowaccess ping
        set type tunnel
        set snmp-index 117
        set interface "port2"
    next
    edit "BR-Lo"
        set vdom "root"
        set ip 10.100.100.1 255.255.255.255
        set allowaccess ping
        set type loopback
        set snmp-index 118
    next
end
config vpn ipsec phase2-interface
    edit "HUB1-VPN1"
        set phase1name "HUB1-VPN1"
        set proposal aes256-sha256
        set auto-negotiate enable
        set comments "VPN: HUB1-VPN1 [Created by IPSEC Template]"
    next
    edit "HUB1-VPN2"
        set phase1name "HUB1-VPN2"
        set proposal aes256-sha256
        set auto-negotiate enable
        set comments "VPN: HUB1-VPN2 [Created by IPSEC Template]"
    next
end
config system sdwan
    set status enable
    config zone
        edit "DIA"
        next
        edit "overlay"
        next
    end
    config members
        edit 1
            set interface "port1"
            set zone "DIA"
            set gateway 100.1.1.2
        next
        edit 2
            set interface "port2"
            set zone "DIA"
            set gateway 200.2.1.2
```

```
        next
        edit 3
            set interface "HUB1-VPN1"
            set zone "overlay"
        next
        edit 4
            set interface "HUB1-VPN2"
            set zone "overlay"
        next
    end
    config health-check
        edit "HUB_HC"
            set server 10.100.100.100
            set update-static-route disable
            set members 0
            config sla
                edit 1
                    set latency-threshold 20
                    set jitter-threshold 20
                    set packetloss-threshold 5
                next
            end
        next
    end
end
config router route-map
    edit "OUT-OF-SLA"
        config rule
            edit 1
                set set-community "65001:11"
            next
        end
    next
    edit "IN-SLA-TUN1"
        config rule
            edit 1
                set set-community "65001:1"
            next
        end
    next
end
config router bgp
    config neighbor
        edit "10.50.1.1"
            set advertisement-interval 1
            set capability-graceful-restart enable
            set link-down-failover enable
            set soft-reconfiguration enable
```

```
            set description "HUB1-VPN1"
            set interface "HUB1-VPN1"
            set remote-as 65001
            set route-map-out "OUT-OF-SLA"
            set connect-timer 10
            set additional-path receive
            set route-map-out-preferable "IN-SLA-TUN1"
        next
    end
end
config system sdwan
    config neighbor
        edit "10.50.1.1"
            set health-check "HUB_HC"
            set sla-id 1
            set member 4
        next
    end
end
config router route-map
    edit "IN-SLA-TUN2"
        config rule
            edit 1
                set set-community "65001:2"
            next
        end
    next
end
config router bgp
    config neighbor
        edit "10.50.2.1"
            set advertisement-interval 1
            set capability-graceful-restart enable
            set link-down-failover enable
            set soft-reconfiguration enable
            set description "HUB1-VPN2"
            set interface "HUB1-VPN2"
            set remote-as 65001
            set route-map-out "OUT-OF-SLA"
            set connect-timer 10
            set additional-path receive
            set route-map-out-preferable "IN-SLA-TUN2"
        next
    end
end
config system sdwan
    config neighbor
        edit "10.50.2.1"
            set health-check "HUB_HC"
            set sla-id 1
            set member 3
        next
    end
end
config router prefix-list
    edit "all_prefixes"
        config rule
            edit 1
                set prefix any
                unset ge
                unset le
            next
        end
    next
end
config router route-map
    edit "port3_only"
        config rule
            edit 1
                set match-interface "port3"
            next
            edit 2
                set action deny
                set match-ip-address "all_prefixes"
            next
        end
    next
end
config router bgp
    set as 65001
    set router-id 10.100.100.1
    set ibgp-multipath enable
    set additional-path enable
    set graceful-restart enable
    set additional-path-select 255
    config network
        edit 1
            set prefix 10.10.1.0 255.255.255.0
        next
    end
    set recursive-next-hop enable
    config redistribute "connected"
        set route-map "port3_only"
    end
end
```

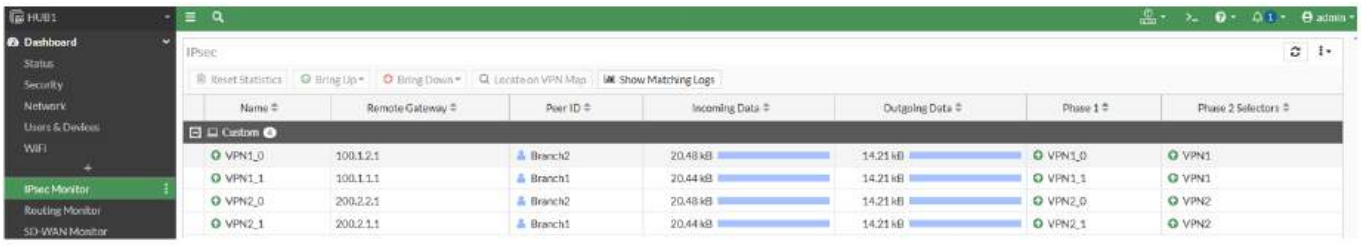
HUB Configuration Preview

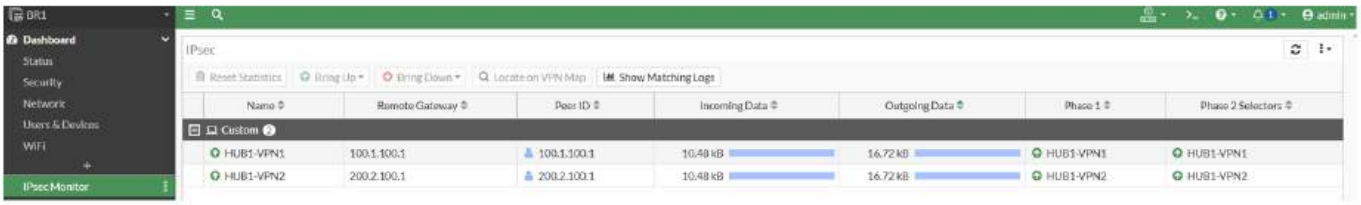
```
config vpn ipsec phase1-interface
    edit "VPN1"
        set type dynamic
        set interface "port1"
        set ike-version 2
        set dpd on-idle
        set comments "VPN: VPN1 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set dpd-retryinterval 60
        set net-device disable
        set add-route disable
        set auto-discovery-sender enable
        set ipv4-start-ip 10.50.1.10
        set ipv4-end-ip 10.50.1.250
        set ipv4-netmask 255.255.255.0
        set psksecret ENC Un6Une1vtiGFiBsCvmESNhp6LVgclszcHC8MEeCa
        set network-overlay enable
        set network-id 1
    next
end
config system interface
    edit "VPN1"
        set vdom "root"
        set ip 10.50.1.1 255.255.255.255
        set type tunnel
        set remote-ip 10.50.1.253 255.255.255.0
        set snmp-index 116
        set interface "port1"
    next
end
config vpn ipsec phase1-interface
    edit "VPN2"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set dpd on-idle
        set comments "VPN: VPN2 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set dpd-retryinterval 60
        set net-device disable
        set add-route disable
        set auto-discovery-sender enable
        set ipv4-start-ip 10.50.2.10
        set ipv4-end-ip 10.50.2.250
        set ipv4-netmask 255.255.255.0
        set psksecret ENC Un6Une1vtiGFiBsCvmESNhp6LVgclszcHC8MEeCa
        set network-overlay enable
        set network-id 2
    next
end
config system interface
    edit "VPN2"
        set vdom "root"
        set ip 10.50.2.1 255.255.255.255
        set type tunnel
        set remote-ip 10.50.2.253 255.255.255.0
        set snmp-index 117
        set interface "port2"
    next
    edit "HUB1-Lo"
        set vdom "root"
        set ip 10.100.100.100 255.255.255.255
        set allowaccess ping
        set type loopback
        set snmp-index 118
    next
end
config vpn ipsec phase2-interface
    edit "VPN1"
        set phase1name "VPN1"
        set proposal aes256-sha256
        set comments "VPN: VPN1 [Created by IPSEC Template]"
    next
    edit "VPN2"
        set phase1name "VPN2"
        set proposal aes256-sha256
        set comments "VPN: VPN2 [Created by IPSEC Template]"
    next
end
config system sdwan
    config zone
        edit "overlay"
        next
    end
    config members
        edit 3
            set interface "VPN1"
            set zone "overlay"
        next
        edit 4
            set interface "VPN2"
            set zone "overlay"
        next
    end
    config service
        edit 1
            set name "TUN1"
            set route-tag 1
            set src "all"
            set priority-members 3
        next
        edit 2
            set name "TUN2"
            set route-tag 2
            set src "all"
            set priority-members 4
        next
        edit 3
            set name "OUT-OF-SLA"
            set route-tag 11
            set priority-members 3 4
        next
    end
end
config router prefix-list
    edit "all_prefixes"
        config rule
            edit 1
                set prefix any
                unset ge
                unset le
            next
        end
    next
end
config router community-list
    edit "65001:1"
        config rule
            edit 1
                set action permit
                set match "65001:1"
            next
        end
    next
    edit "65001:11"
        config rule
            edit 1
                set action permit
                set match "65001:11"
            next
```

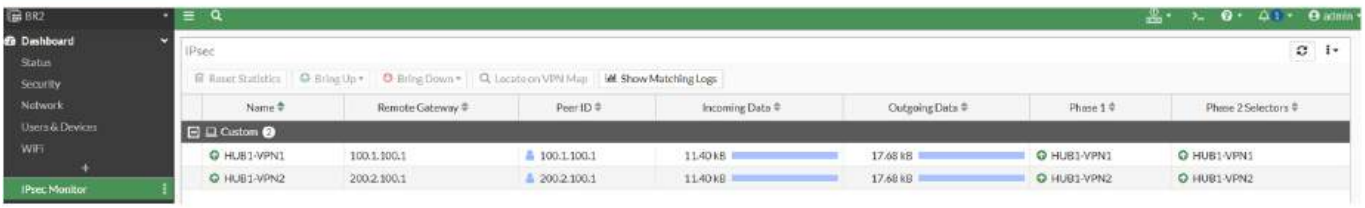
```
        end
    next
    edit "65001:2"
        config rule
            edit 1
                set action permit
                set match "65001:2"
            next
        end
    next
end
config router route-map
    edit "Route-Map-In"
        config rule
            edit 1
                set match-community "65001:1"
                set set-route-tag "1"
            next
            edit 2
                set match-community "65001:2"
                set set-route-tag "2"
            next
            edit 3
                set match-community "65001:11"
                set set-route-tag "11"
            next
        end
    next
    edit "port3_only"
        config rule
            edit 1
                set match-interface "port3"
            next
            edit 2
                set action deny
                set match-ip-address "all_prefixes"
            next
        end
    next
end
config router bgp
    set as 65001
    set router-id 10.100.100.100
    set ibgp-multipath enable
    set additional-path enable
    set graceful-restart enable
    set additional-path-select 255
    config neighbor-group
        edit "VPN1"
            set capability-graceful-restart enable
            set link-down-failover enable
            set next-hop-self enable
            set soft-reconfiguration enable
            set remote-as 65001
            set route-map-in "Route-Map-In"
            set additional-path send
            set route-reflector-client enable
        next
        edit "VPN2"
            set capability-graceful-restart enable
            set link-down-failover enable
            set next-hop-self enable
            set soft-reconfiguration enable
            set remote-as 65001
            set route-map-in "Route-Map-In"
            set additional-path send
            set route-reflector-client enable
        next
    end
    config neighbor-range
        edit 1
            set prefix 10.50.1.0 255.255.255.0
            set neighbor-group "VPN1"
        next
        edit 2
            set prefix 10.50.2.0 255.255.255.0
            set neighbor-group "VPN2"
        next
    end
    config network
        edit 1
            set prefix 10.10.100.0 255.255.255.0
        next
    end
    config redistribute "connected"
        set route-map "port3_only"
    end
end
```

IPsec Verification

Four Tunnels are established at the HUB, two at each branch.







Hub routing verification

Four neighbors are formed, two per branch with appropriate tags.

```
HUB1 # get router info bgp summary

VRF 0 BGP router identifier 10.100.100.100, local AS number 65001
BGP table version is 2
1 BGP AS-PATH entries
2 BGP community entries
Next peer check timer due in 6 seconds

Neighbor   V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.50.1.10 4      65001      12      17        2    0    0 00:06:36        1
10.50.1.11 4      65001      12      17        2    0    0 00:06:36        1
10.50.2.10 4      65001      12      17        2    0    0 00:06:36        1
10.50.2.11 4      65001      12      17        2    0    0 00:06:37        1

Total number of neighbors 4


HUB1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 10.50.1.11 (recursive is directly connected, VPN1), 00:04:32, [1/0]
                     [200/0] via 10.50.2.11 (recursive is directly connected, VPN2), 00:04:32, [1/0]
B       10.10.2.0/24 [200/0] via 10.50.1.10 (recursive is directly connected, VPN1), 00:04:33, [1/0]
                     [200/0] via 10.50.2.10 (recursive is directly connected, VPN2), 00:04:33, [1/0]


HUB1 # get router info bgp network
VRF 0 BGP table version is 2, local router ID is 10.100.100.100
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.10.1.0/24     10.50.2.11      0              100      0        2 i <-/2>
*>i                 10.50.1.11      0              100      0        1 i <-/1>
*>i10.10.2.0/24     10.50.1.10      0              100      0        1 i <-/1>
*>i                 10.50.2.10      0              100      0        2 i <-/2>
*> 10.10.100.0/24   0.0.0.0                        100  32768        0 i <-/1>

Total number of prefixes 3
```

Branch routing verification

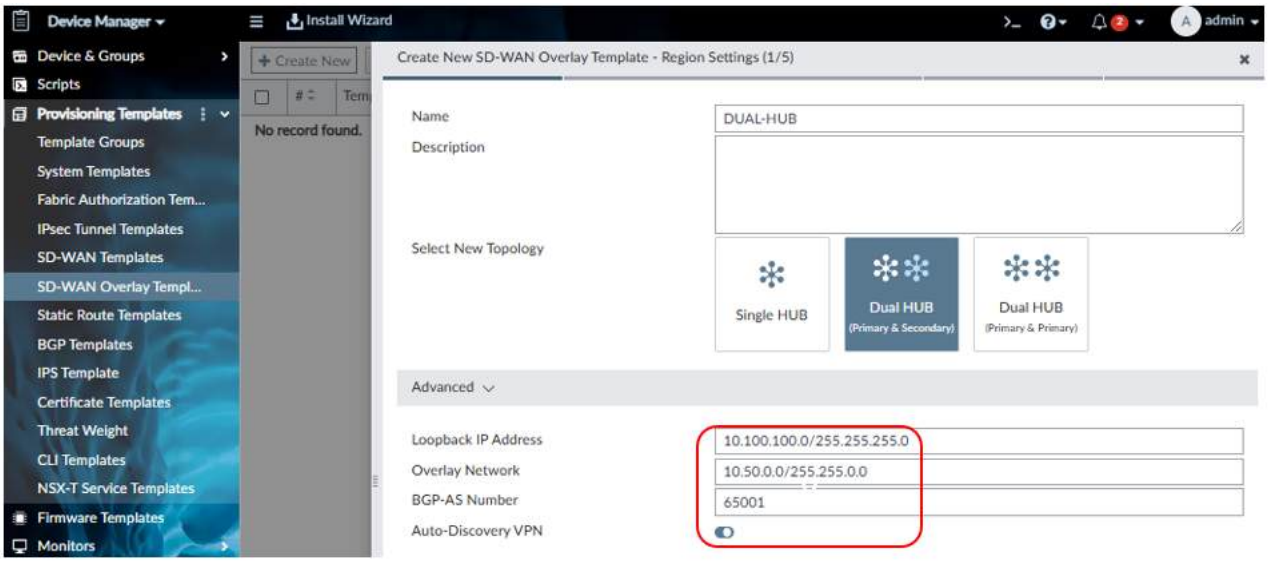
Each branch has two neighborships to HUB1 over the two tunnels.

```
BR1 # get router info bgp summary

VRF 0 BGP router identifier 10.100.100.1, local AS number 65001
BGP table version is 3
1 BGP AS-PATH entries
2 BGP community entries

Neighbor  V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.50.1.1 4      65001      18      12        2    0    0 00:06:58        3
10.50.2.1 4      65001      17      12        2    0    0 00:06:59        3

Total number of neighbors 2


BR1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.2.0/24 [200/0] via 10.50.1.10 [2] (recursive is directly connected, HUB1-VPN1), 00:04:56, [1/0]
                     [200/0] via 10.50.2.10 [2] (recursive is directly connected, HUB1-VPN2), 00:04:56, [1/0]
B       10.10.100.0/24 [200/0] via 10.50.1.1 (recursive via HUB1-VPN1 tunnel 100.1.100.1), 00:06:58, [1/0]
                       [200/0] via 10.50.2.1 (recursive via HUB1-VPN2 tunnel 200.2.100.1), 00:06:58, [1/0]
```

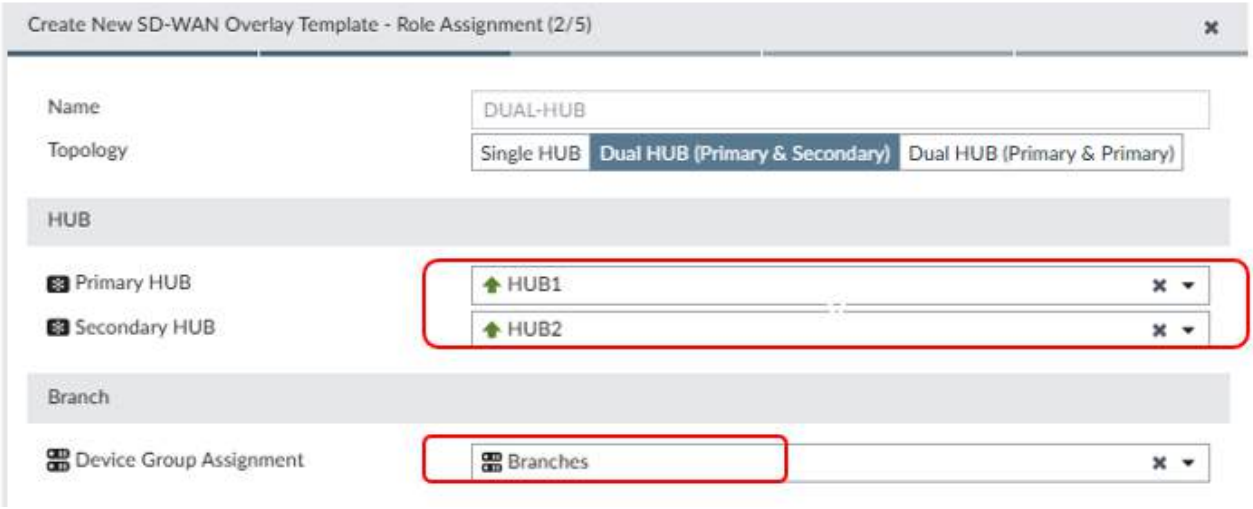
```
BR2 # get router info bgp summary

VRF 0 BGP router identifier 10.100.100.2, local AS number 65001
BGP table version is 3
1 BGP AS-PATH entries
2 BGP community entries

Neighbor  V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.50.1.1 4      65001      18      13        2    0    0 00:07:09        3
10.50.2.1 4      65001      18      13        2    0    0 00:07:09        3

Total number of neighbors 2


BR2 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 10.50.1.11 [2] (recursive is directly connected, HUB1-VPN1), 00:04:41, [1/0]
                     [200/0] via 10.50.2.11 [2] (recursive is directly connected, HUB1-VPN2), 00:04:41, [1/0]
B       10.10.100.0/24 [200/0] via 10.50.1.1 (recursive via HUB1-VPN1 tunnel 100.1.100.1), 00:06:47, [1/0]
                       [200/0] via 10.50.2.1 (recursive via HUB1-VPN2 tunnel 200.2.100.1), 00:06:47, [1/0]
```

SDWAN OVERLAY Templates- DUAL HUB

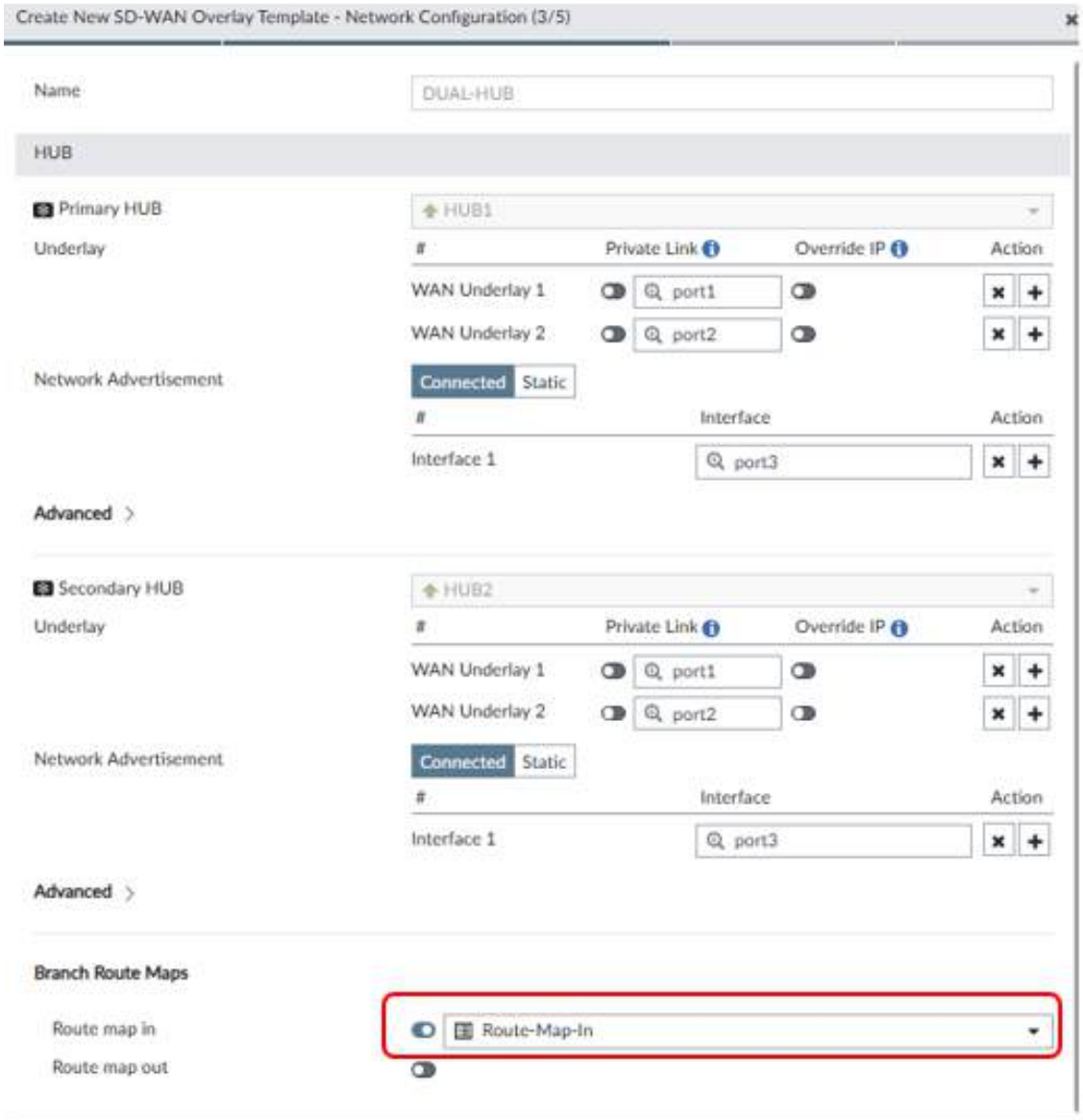
For a Dual hub setup, ensure all devices are registered and the branches are grouped before provisioning the overlay template. Select the Dual HUB option and fill the IP addressing, AS number and activate ADVPN.

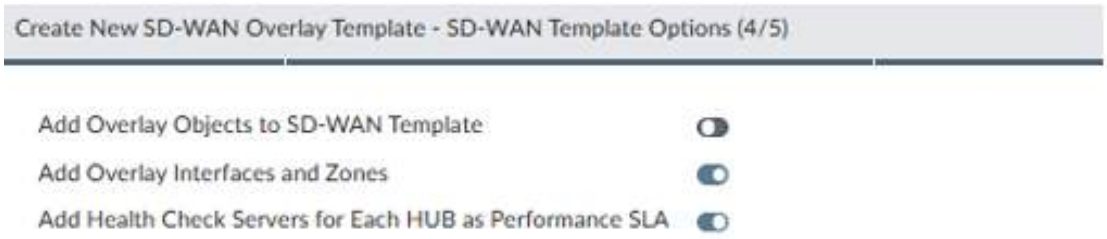


Specify the HUBs (primary/secondary) and branches.

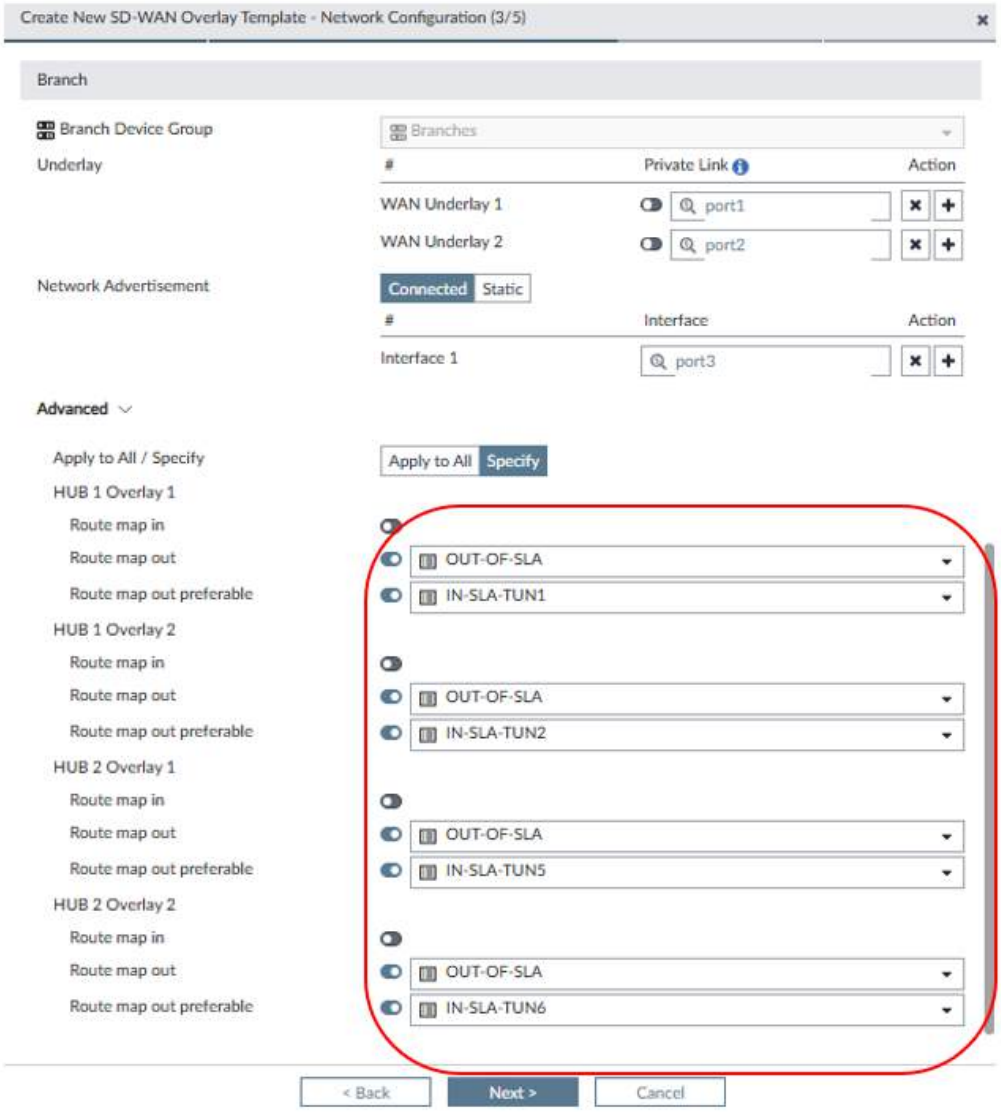


For the HUB, Rout-map-in is configured for both HUBs.

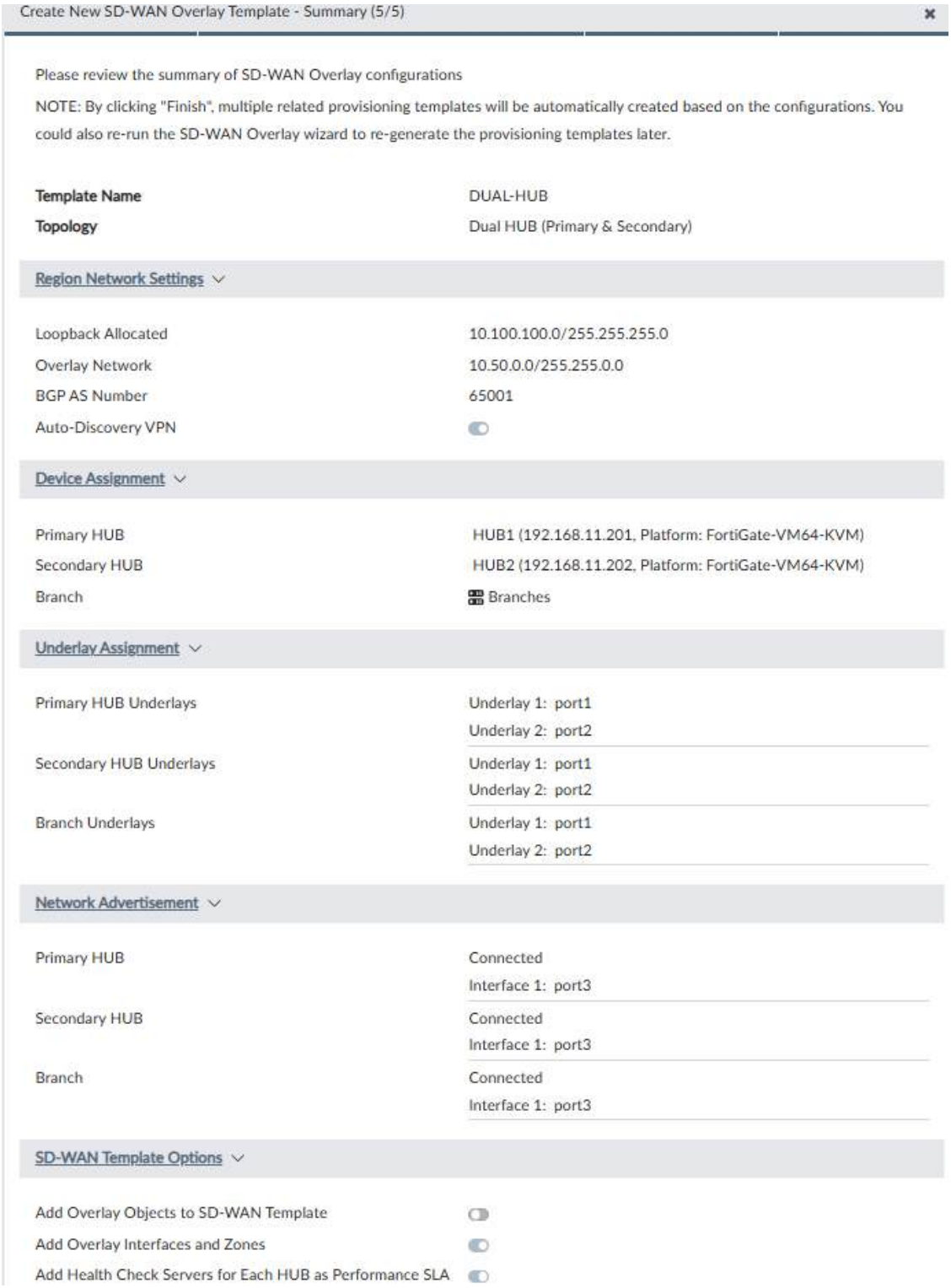




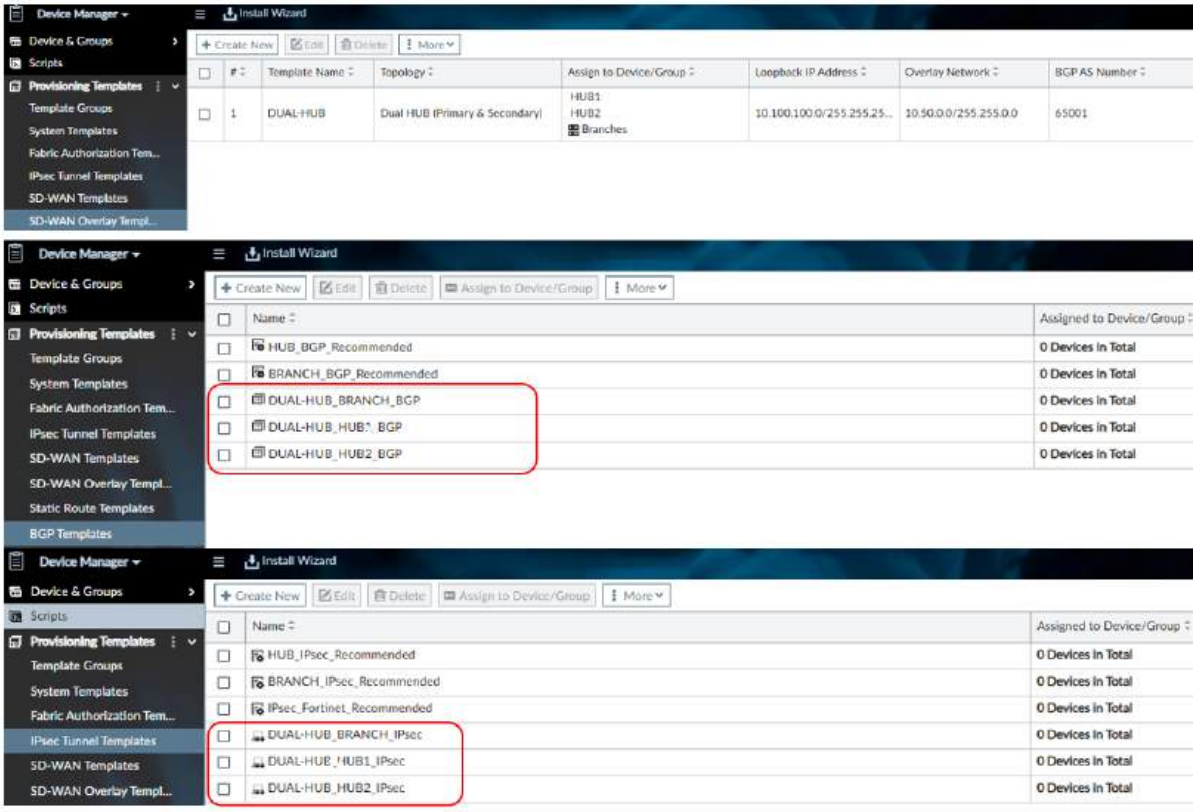
Branches have the tunnels configured for both HUBs with respective route-maps based on SLA state.



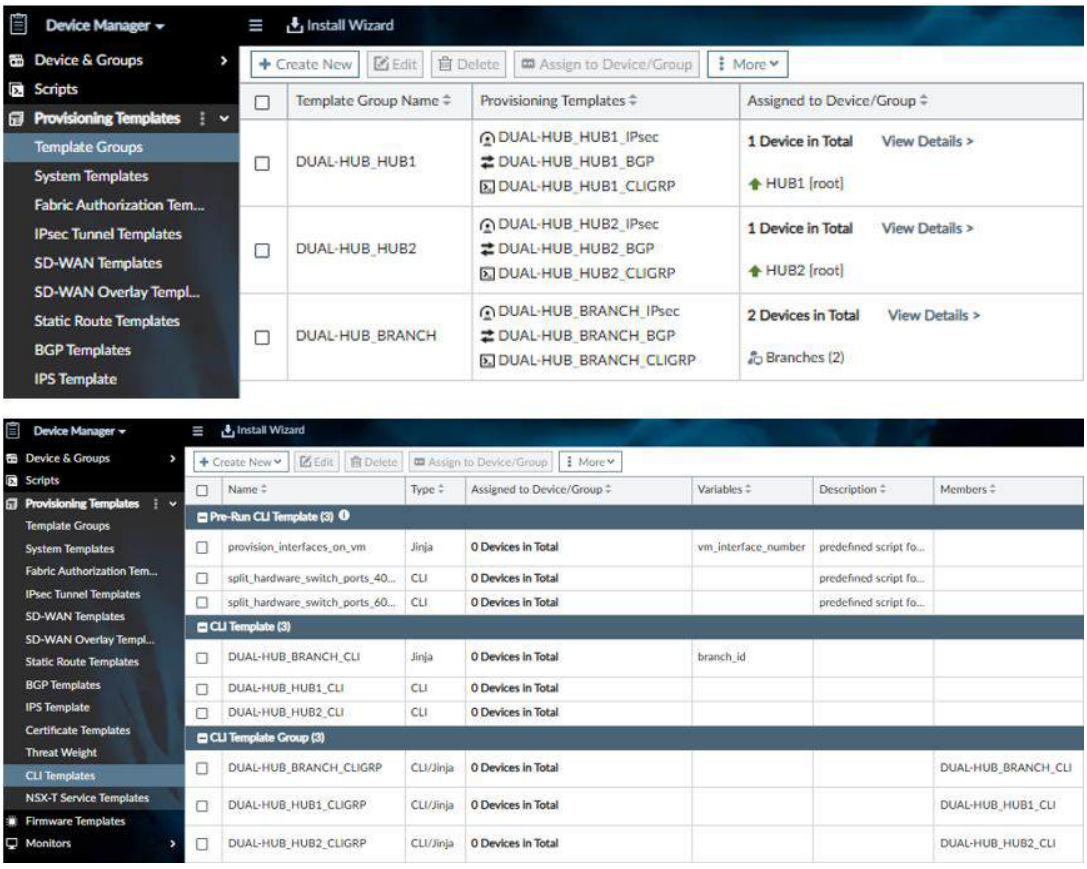
The configuration is summarized below for the dual-hub overlay template:



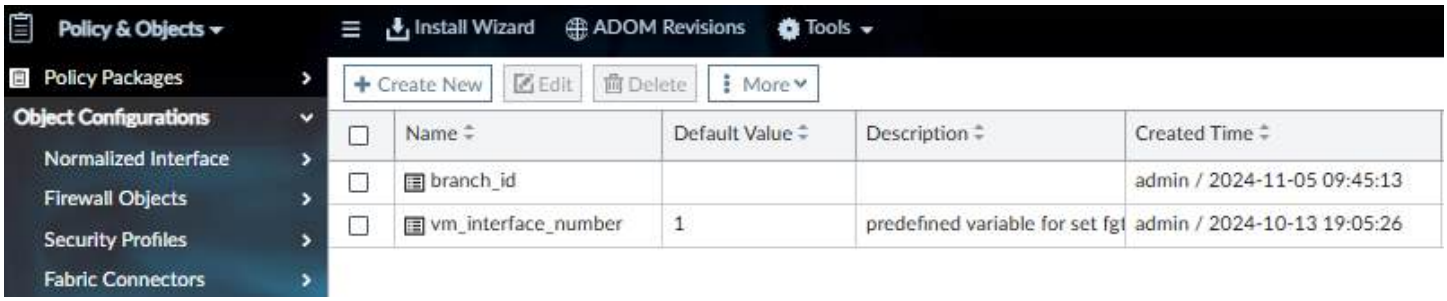
The wizard creates the IPsec and BGP templates for the branches and HUBs.



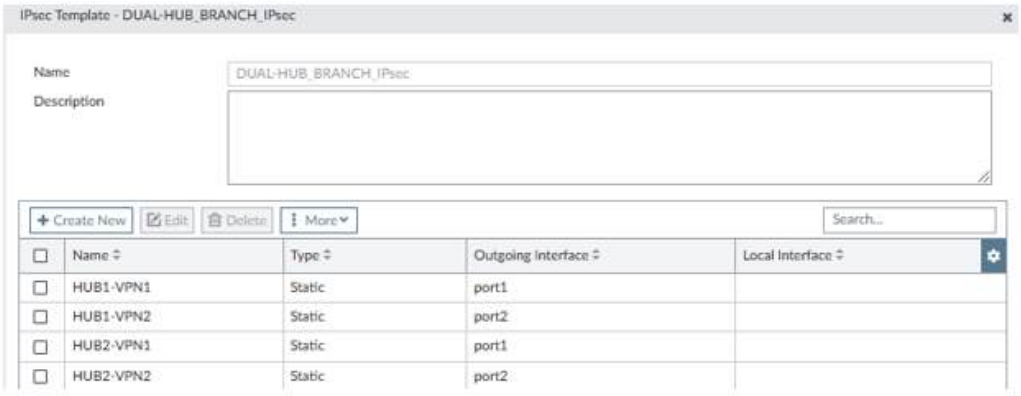
Also created are Template Groups and CLI templates.

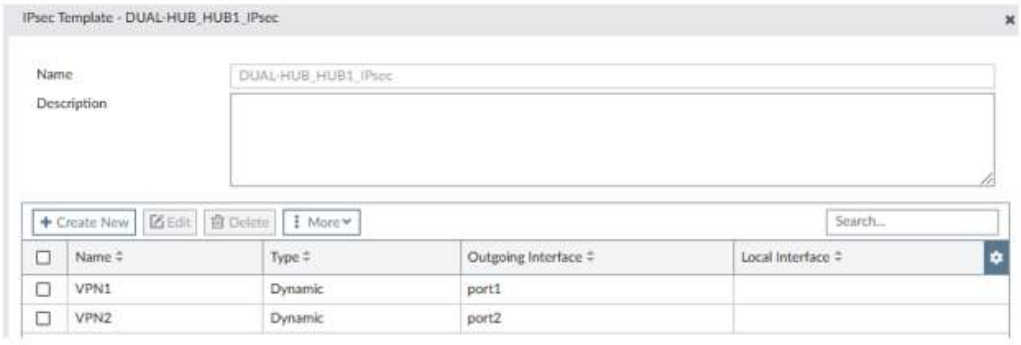


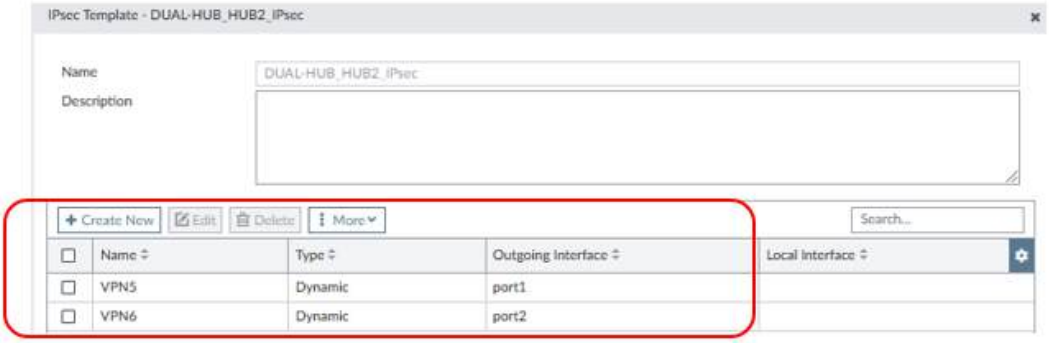
The metadata variable branch_id is generated.



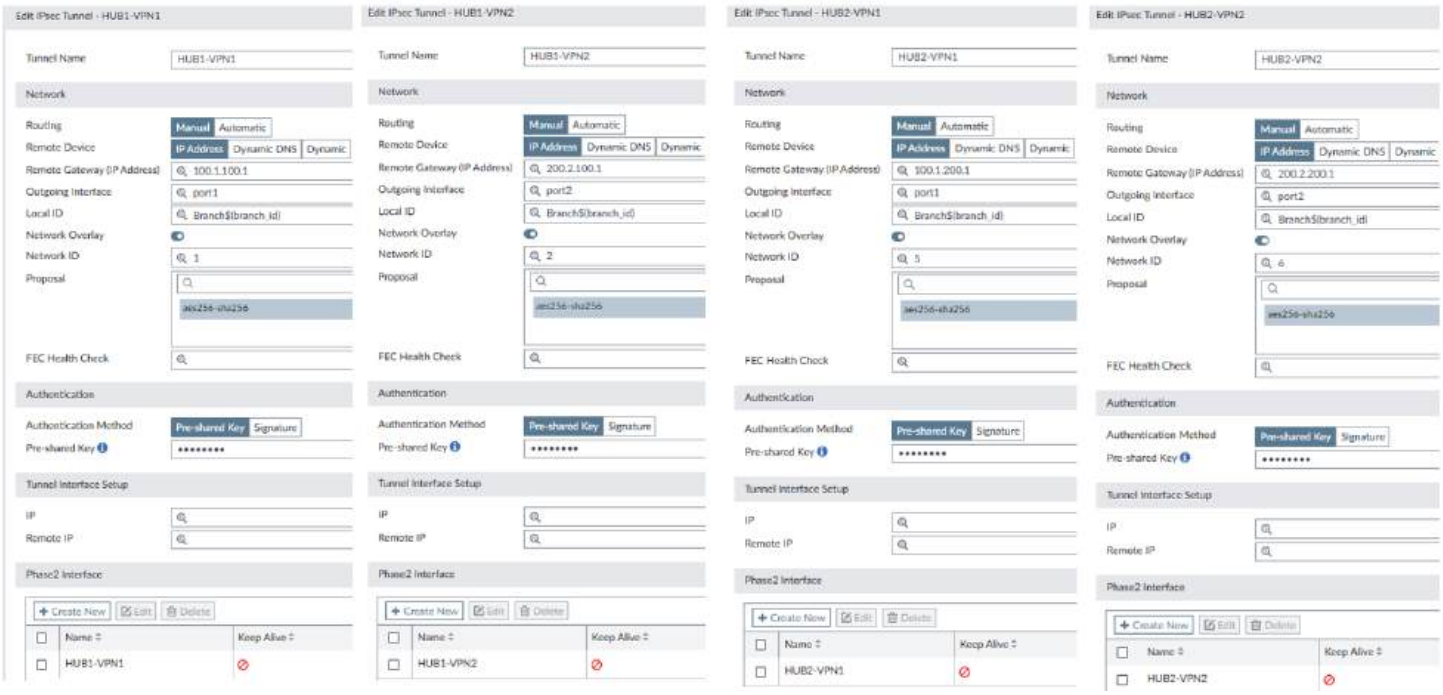
Customization is done on HUB2 IPsec to match the tunnel numbers for uniformity.



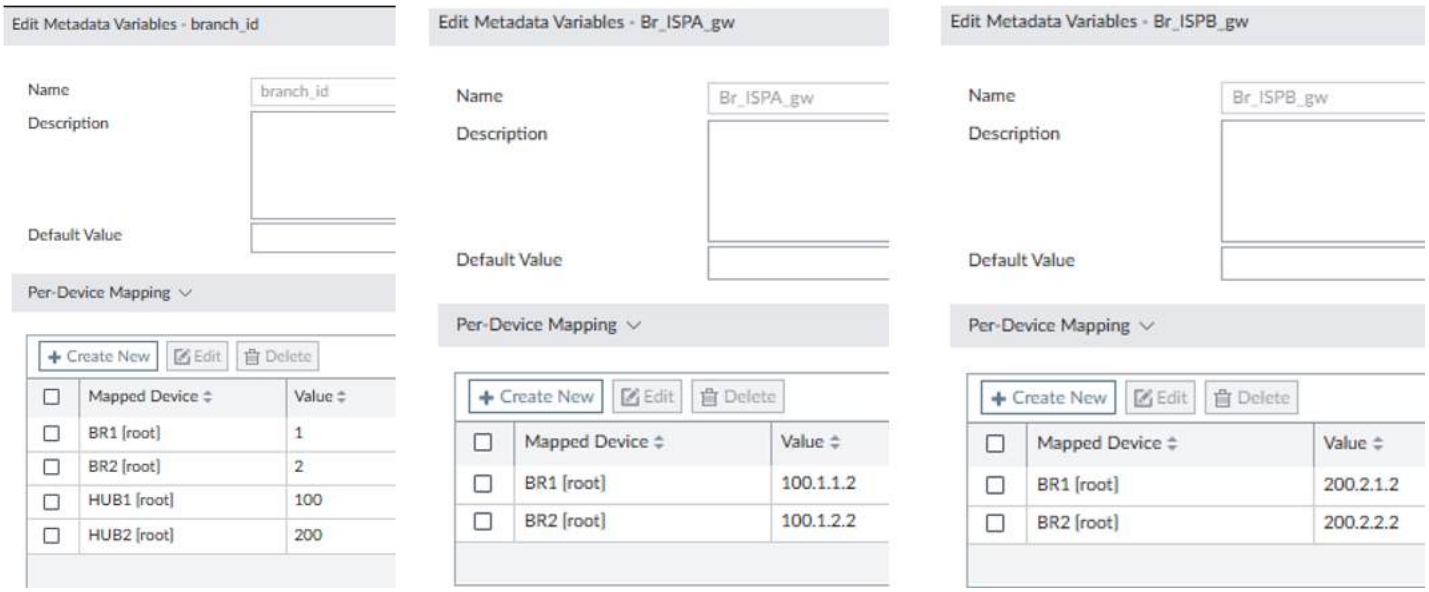




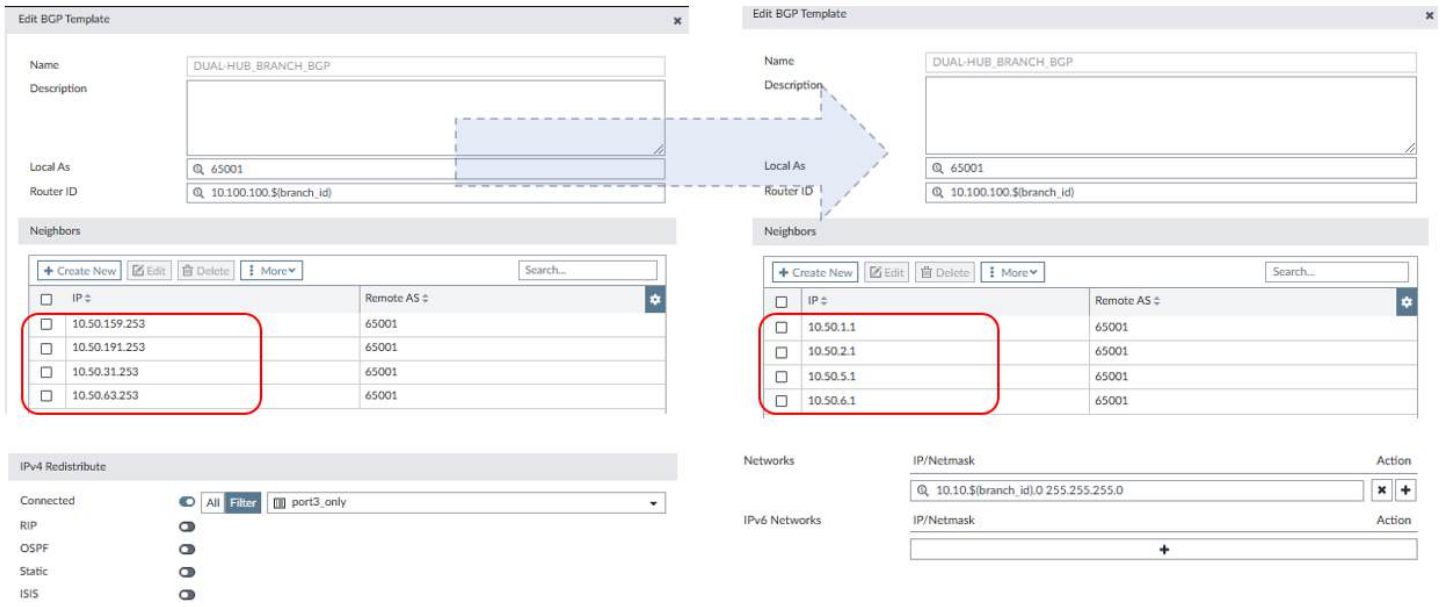
No change is required on branch IPsec Templates for the default tunnels created. If additional tunnels are required, the defined tunnels are cloned.



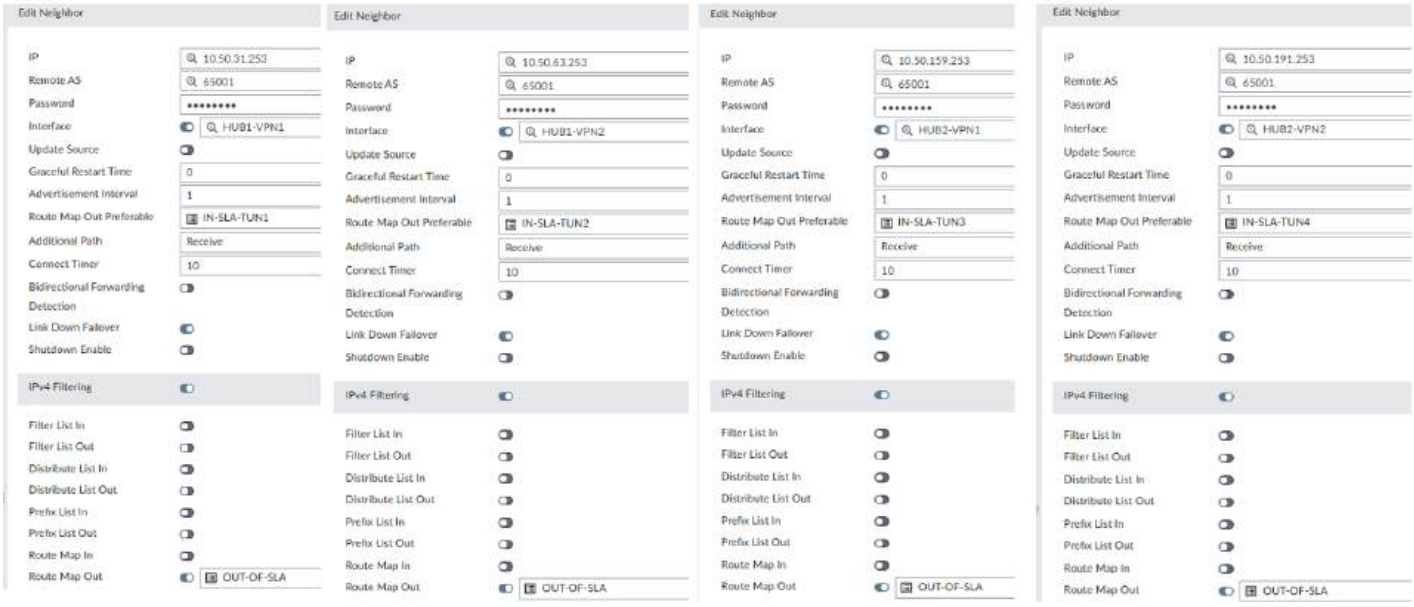
Metadata Variables are defined for all units.



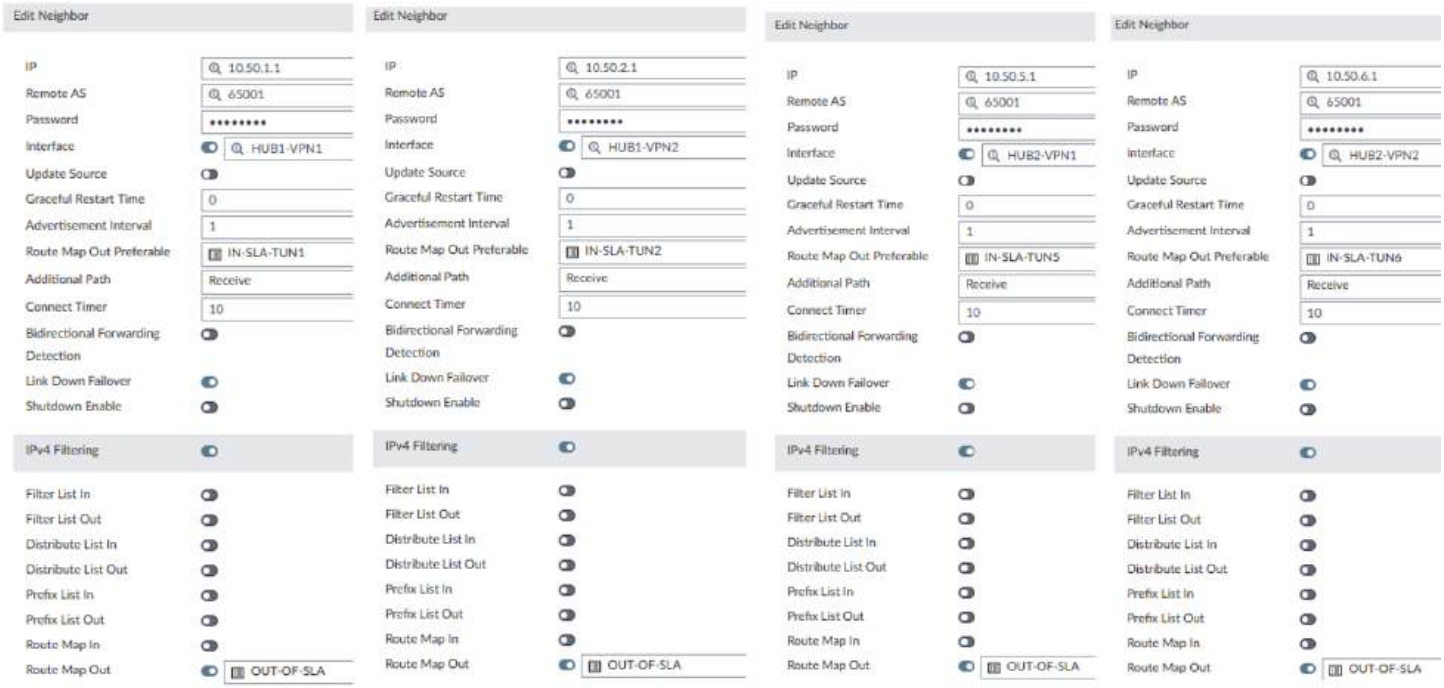
For the branches, BGP neighbors are changed from the default random template addressing to tunnel specific ip addresses.



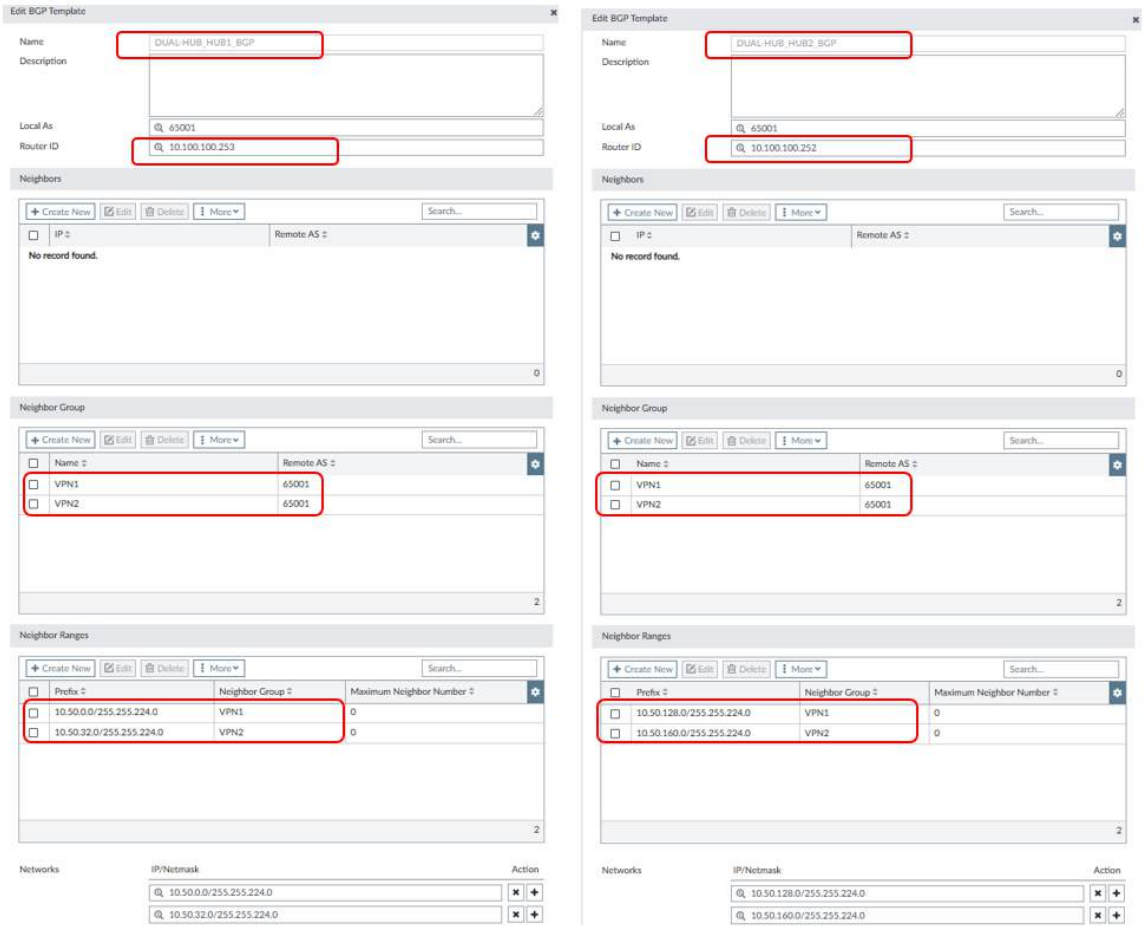
The default template neighbor settings are shown below:



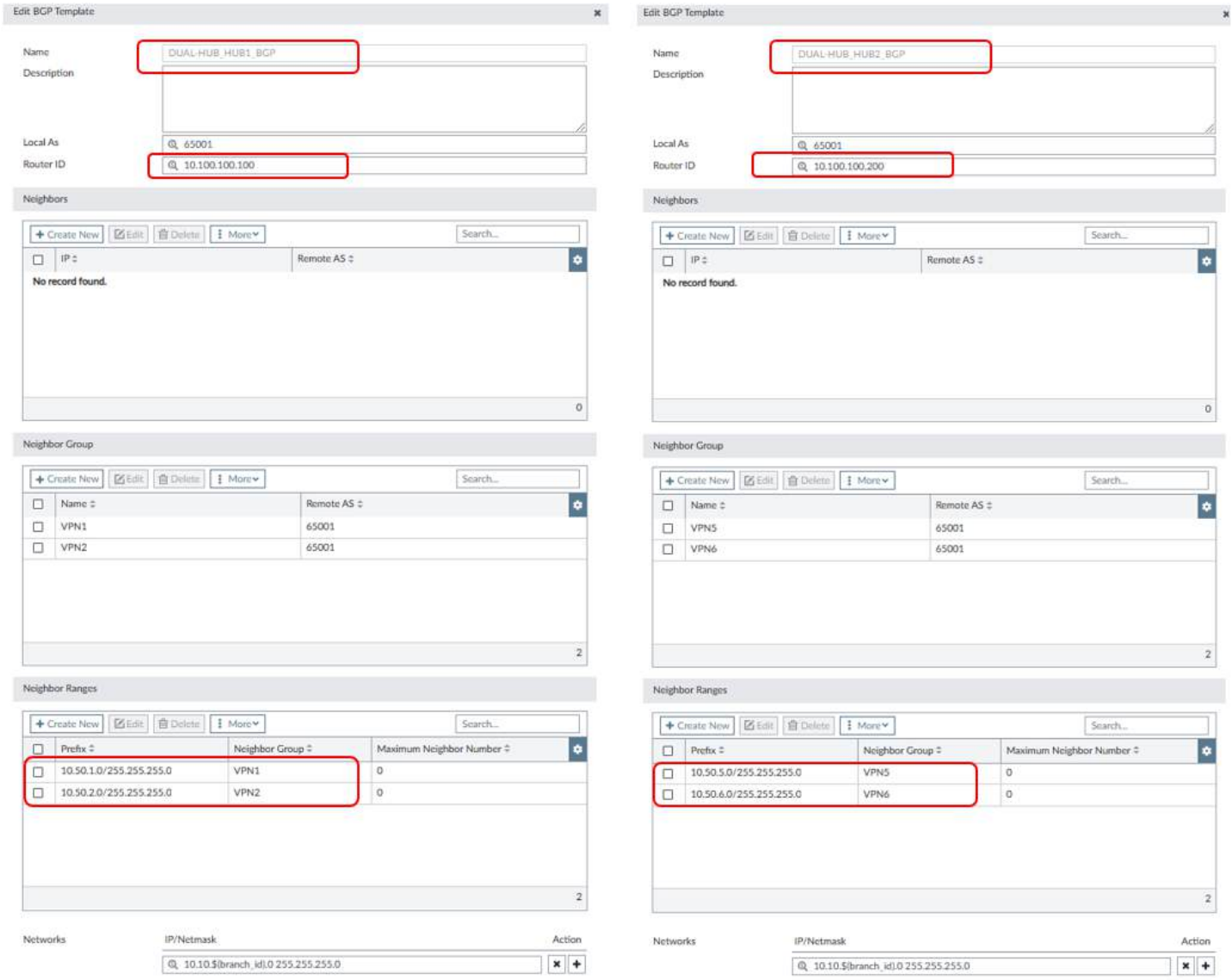
After customization to reference the LAB network the neighborships are shown on the diagram below.



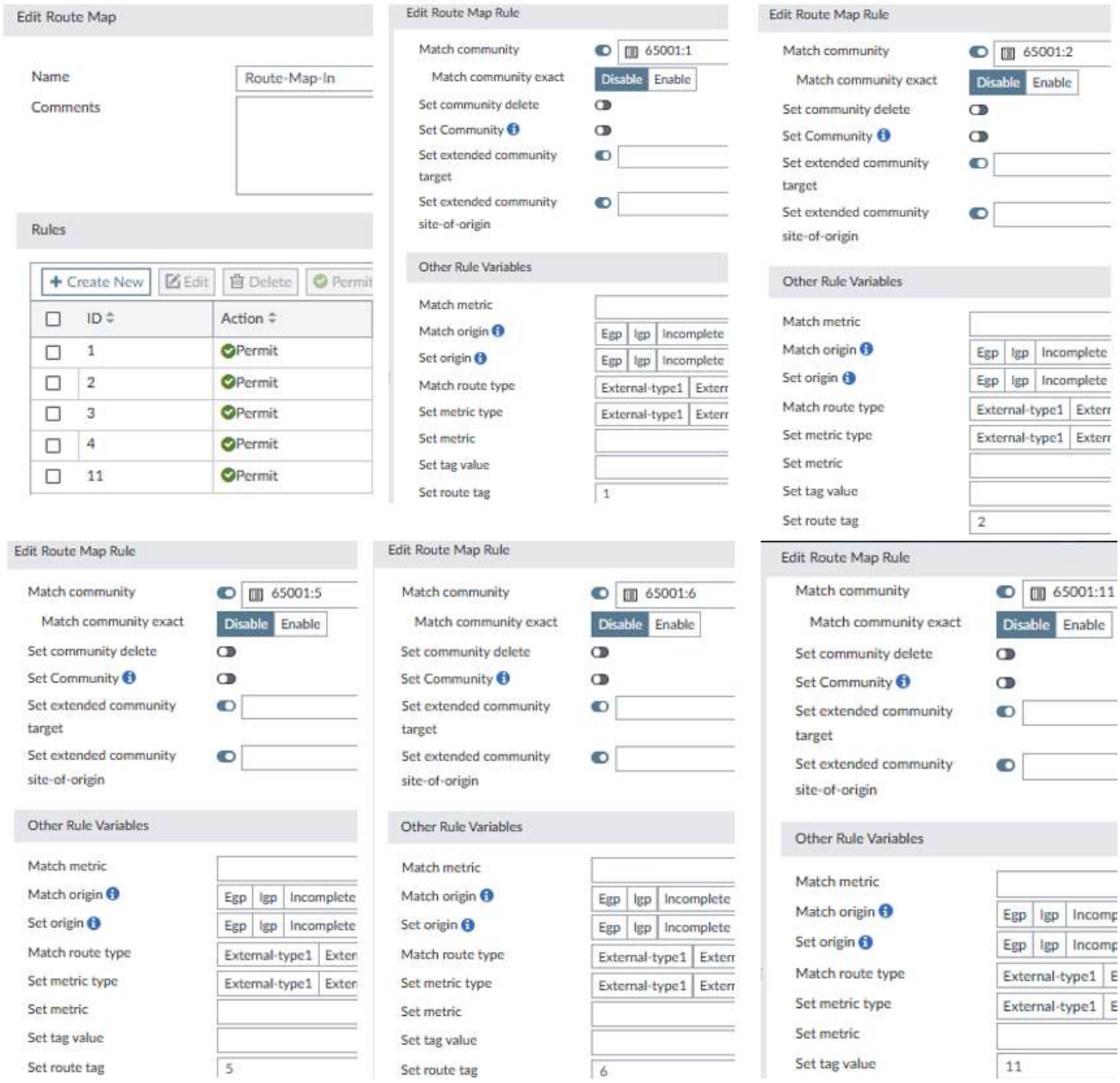
HUB1 and HUB2 default BGP settings from the overlay template are presented below:



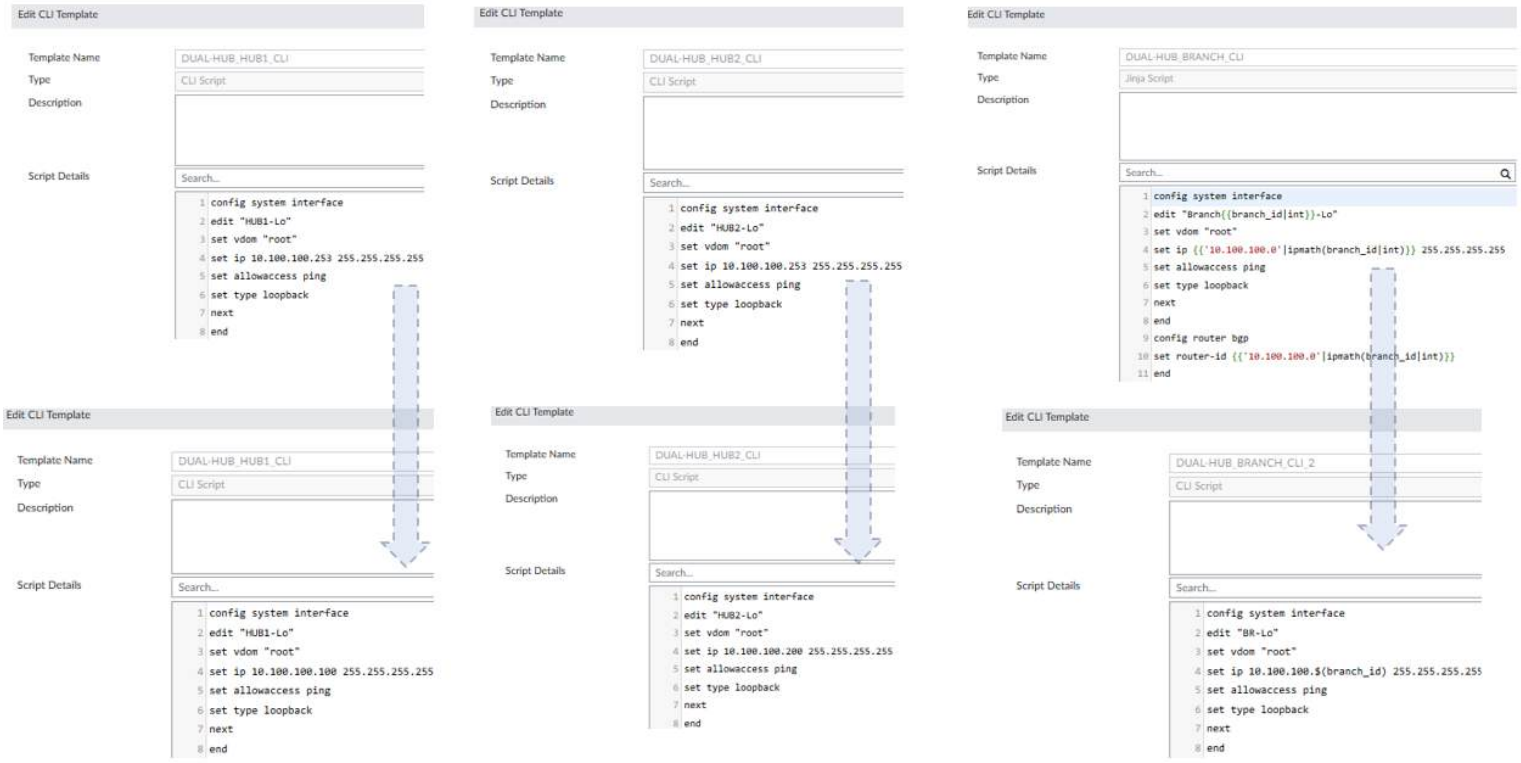
The BGP router-ID and neighbor groups are customized to the LAB settings.



The route-map-in at the HUB is configured to match inbound communities and set required tags.



Finally, the CLI templates are customized accordingly.



Branch 2 Configuration Preview

```
config vpn ipsec phase1-interface
    edit "HUB1-VPN1"
        set interface "port1"
        set ike-version 2
        set comments "VPN: HUB1-VPN1 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set localid "Branch2"
        set remote-gw 100.1.100.1
        set idle-timeout enable
        set net-device enable
        set add-route disable
        set auto-discovery-receiver enable
        set psksecret ENC RjZSBztK7ZzFo5D3gV36uNuoyqfRaOx6RZvIQh5XQa
        set network-overlay enable
        set network-id 1
        set auto-discovery-shortcuts dependent
    next
end
config system interface
    edit "HUB1-VPN1"
        set vdom "root"
        set type tunnel
        set snmp-index 116
        set interface "port1"
    next
end
config vpn ipsec phase1-interface
    edit "HUB1-VPN2"
        set interface "port2"
        set ike-version 2
        set comments "VPN: HUB1-VPN2 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set localid "Branch2"
        set remote-gw 200.2.100.1
        set idle-timeout enable
        set net-device enable
        set add-route disable
        set auto-discovery-receiver enable
        set psksecret ENC RjZSBztK7ZzFo5D3gV36uNuoyqfRaOx6RZvIQh5XQa
        set network-overlay enable
        set network-id 2
        set auto-discovery-shortcuts dependent
    next
end
```

```
config system interface
    edit "HUB1-VPN2"
        set vdom "root"
        set type tunnel
        set snmp-index 117
        set interface "port2"
    next
end
config vpn ipsec phase1-interface
    edit "HUB2-VPN1"
        set interface "port1"
        set ike-version 2
        set comments "VPN: HUB2-VPN1 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set localid "Branch2"
        set remote-gw 100.1.200.1
        set idle-timeout enable
        set net-device enable
        set add-route disable
        set auto-discovery-receiver enable
        set psksecret ENC RjZSBztK7ZzFo5D3gV36uNuoyqfRaOx6RZvIQh5XQav
        set network-overlay enable
        set network-id 5
        set auto-discovery-shortcuts dependent
    next
end
config system interface
    edit "HUB2-VPN1"
        set vdom "root"
        set type tunnel
        set snmp-index 118
        set interface "port1"
    next
end
config vpn ipsec phase1-interface
    edit "HUB2-VPN2"
        set interface "port2"
        set ike-version 2
        set comments "VPN: HUB2-VPN2 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set localid "Branch2"
        set remote-gw 200.2.200.1
        set idle-timeout enable
        set net-device enable
```

```
        set add-route disable
        set auto-discovery-receiver enable
        set psksecret ENC RjZSBztK7ZzFo5D3gV36uNuoyqfRaOx6RZvIQh5XQav
        set network-overlay enable
        set network-id 6
        set auto-discovery-shortcuts dependent
    next
end
config system interface
    edit "HUB2-VPN2"
        set vdom "root"
        set type tunnel
        set snmp-index 119
        set interface "port2"
    next
    edit "BR-Lo"
        set vdom "root"
        set ip 10.100.100.2 255.255.255.255
        set allowaccess ping
        set type loopback
        set snmp-index 120
    next
end
config vpn ipsec phase2-interface
    edit "HUB1-VPN1"
        set phase1name "HUB1-VPN1"
        set proposal aes256-sha256
        set auto-negotiate enable
        set comments "VPN: HUB1-VPN1 [Created by IPSEC Template]"
    next
    edit "HUB1-VPN2"
        set phase1name "HUB1-VPN2"
        set proposal aes256-sha256
        set auto-negotiate enable
        set comments "VPN: HUB1-VPN2 [Created by IPSEC Template]"
    next
    edit "HUB2-VPN1"
        set phase1name "HUB2-VPN1"
        set proposal aes256-sha256
        set auto-negotiate enable
        set comments "VPN: HUB2-VPN1 [Created by IPSEC Template]"
    next
    edit "HUB2-VPN2"
        set phase1name "HUB2-VPN2"
        set proposal aes256-sha256
        set auto-negotiate enable
        set comments "VPN: HUB2-VPN2 [Created by IPSEC Template]"
    next
```

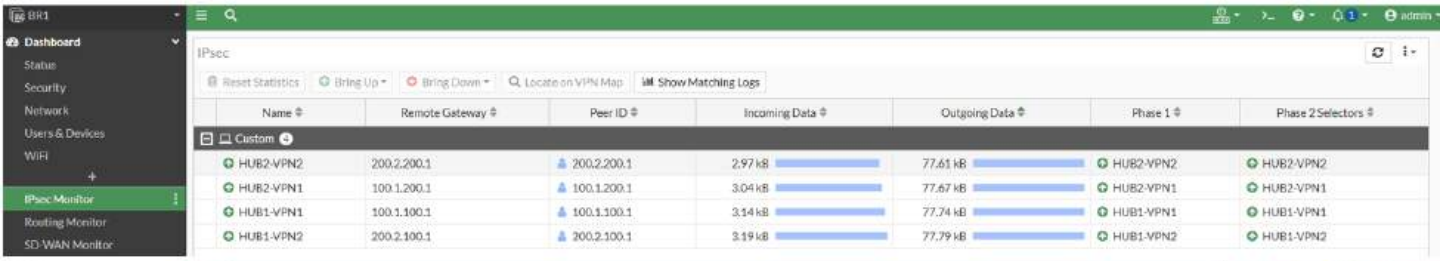
```
end
config system sdwan
    set status enable
    config zone
        edit "overlay"
        next
        edit "overlay2"
        next
        edit "DIA"
        next
    end
    config members
        edit 1
            set interface "port1"
            set zone "DIA"
            set gateway 100.1.2.2
        next
        edit 2
            set interface "port2"
            set zone "DIA"
            set gateway 200.2.2.2
        next
        edit 3
            set interface "HUB1-VPN1"
            set zone "overlay"
        next
        edit 4
            set interface "HUB1-VPN2"
            set zone "overlay"
        next
        edit 5
            set interface "HUB2-VPN1"
            set zone "overlay2"
        next
        edit 6
            set interface "HUB2-VPN2"
            set zone "overlay2"
        next
    end
    config health-check
        edit "HUB1_HC"
            set server 10.100.100.100
            set update-static-route disable
            set members 4 3
            config sla
                edit 1
                    set latency-threshold 20
                    set jitter-threshold 20
                    set packetloss-threshold 5
                next
            end
        next
        edit "HUB2_HC"
            set server 10.100.100.200
            set update-static-route disable
            set members 6 5
            config sla
                edit 1
                    set latency-threshold 20
                    set jitter-threshold 20
                    set packetloss-threshold 5
                next
            end
        next
    end
end
config router route-map
    edit "OUT-OF-SLA"
        config rule
            edit 1
                set set-community "65001:11"
            next
        end
    next
    edit "IN-SLA-TUN1"
        config rule
            edit 1
                set set-community "65001:1"
            next
        end
    next
end
config router bgp
    config neighbor
        edit "10.50.1.1"
            set advertisement-interval 1
            set capability-graceful-restart enable
            set link-down-failover enable
            set soft-reconfiguration enable
            set description "HUB1-VPN1"
            set interface "HUB1-VPN1"
            set remote-as 65001
            set route-map-out "OUT-OF-SLA"
            set connect-timer 10
            set additional-path receive
            set route-map-out-preferable "IN-SLA-TUN1"
        next
    end
end
config system sdwan
    config neighbor
        edit "10.50.1.1"
            set health-check "HUB1_HC"
            set sla-id 1
            set member 3
        next
    end
end
config router route-map
    edit "IN-SLA-TUN2"
        config rule
            edit 1
                set set-community "65001:2"
            next
        end
    next
end
config router bgp
    config neighbor
        edit "10.50.2.1"
            set advertisement-interval 1
            set capability-graceful-restart enable
            set link-down-failover enable
            set soft-reconfiguration enable
            set description "HUB1-VPN2"
            set interface "HUB1-VPN2"
            set remote-as 65001
            set route-map-out "OUT-OF-SLA"
            set connect-timer 10
            set additional-path receive
            set route-map-out-preferable "IN-SLA-TUN2"
        next
    end
end
config system sdwan
    config neighbor
        edit "10.50.2.1"
            set health-check "HUB1_HC"
            set sla-id 1
            set member 4
        next
    end
end
config router route-map
```

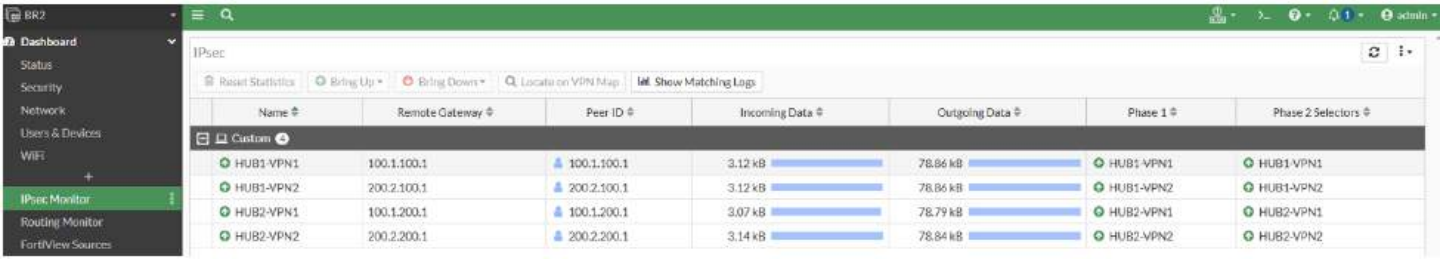
```
    edit "IN-SLA-TUN5"
        config rule
            edit 1
                set set-community "65001:5"
            next
        end
    next
end
config router bgp
    config neighbor
        edit "10.50.5.1"
            set advertisement-interval 1
            set capability-graceful-restart enable
            set link-down-failover enable
            set soft-reconfiguration enable
            set description "HUB2-VPN1"
            set interface "HUB2-VPN1"
            set remote-as 65001
            set route-map-out "OUT-OF-SLA"
            set connect-timer 10
            set additional-path receive
            set route-map-out-preferable "IN-SLA-TUN5"
        next
    end
end
config system sdwan
    config neighbor
        edit "10.50.5.1"
            set health-check "HUB2_HC"
            set sla-id 1
            set member 5
        next
    end
end
config router route-map
    edit "IN-SLA-TUN6"
        config rule
            edit 1
                set set-community "65001:6"
            next
        end
    next
end
config router bgp
    config neighbor
        edit "10.50.6.1"
            set advertisement-interval 1
            set capability-graceful-restart enable
            set link-down-failover enable
            set soft-reconfiguration enable
            set description "HUB2-VPN2"
            set interface "HUB2-VPN2"
            set remote-as 65001
            set route-map-out "OUT-OF-SLA"
            set connect-timer 10
            set additional-path receive
            set route-map-out-preferable "IN-SLA-TUN6"
        next
    end
end
config system sdwan
    config neighbor
        edit "10.50.6.1"
            set health-check "HUB2_HC"
            set sla-id 1
            set member 6
        next
    end
end
config router prefix-list
    edit "all_prefixes"
        config rule
            edit 1
                set prefix any
                unset ge
                unset le
            next
        end
    next
end
config router route-map
    edit "port3_only"
        config rule
            edit 1
                set match-interface "port3"
            next
            edit 2
                set action deny
                set match-ip-address "all_prefixes"
            next
        end
    next
end
config router bgp
    set as 65001
    set router-id 10.100.100.2
    set ibgp-multipath enable
    set additional-path enable
    set graceful-restart enable
    set additional-path-select 255
    config network
        edit 1
            set prefix 10.10.2.0 255.255.255.0
        next
    end
    set recursive-next-hop enable
    config redistribute "connected"
        set route-map "port3_only"
    end
end
```

```
config vpn ipsec phase1-interface
    edit "VPN5"
        set type dynamic
        set interface "port1"
        set ike-version 2
        set dpd on-idle
        set comments "VPN: VPN5 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set dpd-retryinterval 60
        set net-device disable
        set add-route disable
        set auto-discovery-sender enable
        set ipv4-start-ip 10.50.5.10
        set ipv4-end-ip 10.50.5.250
        set ipv4-netmask 255.255.255.0
        set psksecret ENC Z8Zpc/bwU2j1HxCFWzO/XkWz1iP/WK4qAGVHE9oazI
        set network-overlay enable
        set network-id 5
    next
end
config system interface
    edit "VPN5"
        set vdom "root"
        set ip 10.50.5.1 255.255.255.255
        set type tunnel
        set remote-ip 10.50.5.253 255.255.255.0
        set snmp-index 116
        set interface "port1"
    next
end
config vpn ipsec phase1-interface
    edit "VPN6"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set dpd on-idle
        set comments "VPN: VPN6 [Created by IPSEC Template]"
        set proposal aes256-sha256
        set peertype any
        set mode-cfg enable
        set dpd-retryinterval 60
        set net-device disable
        set add-route disable
        set auto-discovery-sender enable
        set ipv4-start-ip 10.50.6.10
        set ipv4-end-ip 10.50.6.250
        set ipv4-netmask 255.255.255.0
        set psksecret ENC Z8Zpc/bwU2j1HxCFWzO/XkWz1iP/WK4qAGVHE9oazI
        set network-overlay enable
        set network-id 6
    next
end
config system interface
    edit "VPN6"
        set vdom "root"
        set ip 10.50.6.1 255.255.255.255
        set type tunnel
        set remote-ip 10.50.6.253 255.255.255.0
        set snmp-index 117
        set interface "port2"
    next
    edit "HUB2-Lo"
        set vdom "root"
        set ip 10.100.100.200 255.255.255.255
        set allowaccess ping
        set type loopback
        set snmp-index 118
    next
end
config firewall address
    edit "all"
        set uuid f4beb98c-897c-51ef-3958-70fd1739153f
    next
end
config vpn ipsec phase2-interface
    edit "VPN5"
        set phase1name "VPN5"
        set proposal aes256-sha256
        set comments "VPN: VPN5 [Created by IPSEC Template]"
    next
    edit "VPN6"
        set phase1name "VPN6"
        set proposal aes256-sha256
        set comments "VPN: VPN6 [Created by IPSEC Template]"
    next
end
config system sdwan
    set status enable
    config zone
        edit "DIA"
        next
        edit "overlay"
        next
    end
    config members
        edit 1
            set interface "port1"
            set zone "DIA"
            set gateway 100.1.200.2
        next
        edit 2
            set interface "port2"
            set zone "DIA"
            set gateway 200.2.200.2
        next
        edit 3
            set interface "VPN5"
            set zone "overlay"
        next
        edit 4
            set interface "VPN6"
            set zone "overlay"
        next
    end
    config service
        edit 1
            set name "TUN5"
            set route-tag 5
            set src "all"
            set priority-members 3
        next
        edit 2
            set name "TUN6"
            set route-tag 6
            set src "all"
            set priority-members 4
        next
        edit 3
            set name "OUT-OF-SLA"
            set route-tag 11
            set priority-members 3 4
        next
    end
end
config router prefix-list
    edit "all_prefixes"
        config rule
            edit 1
                set prefix any
                unset ge
                unset le
            next
```

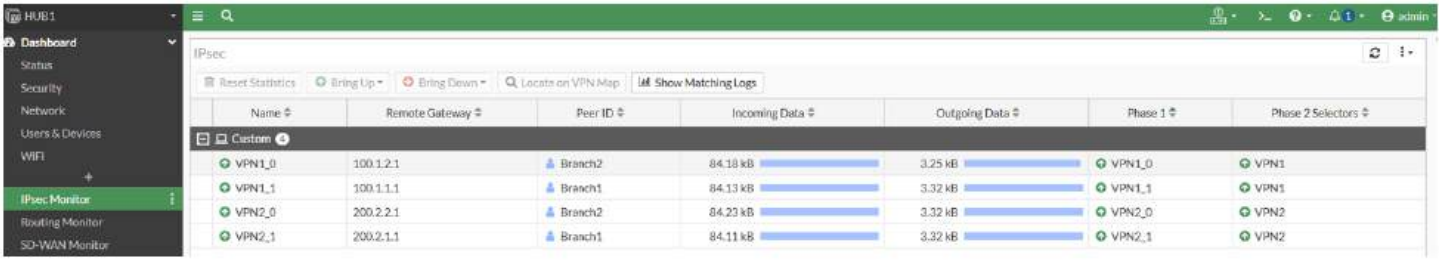
```
        end
    next
end
config router community-list
    edit "65001:1"
        config rule
            edit 1
                set action permit
                set match "65001:1"
            next
        end
    next
    edit "65001:11"
        config rule
            edit 1
                set action permit
                set match "65001:11"
            next
        end
    next
    edit "65001:2"
        config rule
            edit 1
                set action permit
                set match "65001:2"
            next
        end
    next
    edit "65001:5"
        config rule
            edit 1
                set action permit
                set match "65001:5"
            next
        end
    next
    edit "65001:6"
        config rule
            edit 1
                set action permit
                set match "65001:6"
            next
        end
    next
end
config router route-map
    edit "Route-Map-In"
        config rule
            edit 1
                set match-community "65001:1"
                set set-route-tag "1"
            next
            edit 2
                set match-community "65001:2"
                set set-route-tag "2"
            next
            edit 5
                set match-community "65001:5"
                set set-route-tag "5"
            next
            edit 6
                set match-community "65001:6"
                set set-route-tag "6"
            next
            edit 11
                set match-community "65001:11"
                set set-route-tag "11"
            next
        end
    next
    edit "port3_only"
        config rule
            edit 1
                set match-interface "port3"
            next
            edit 2
                set action deny
                set match-ip-address "all_prefixes"
            next
        end
    next
end
config router bgp
    set as 65001
    set router-id 10.100.100.200
    set ibgp-multipath enable
    set additional-path enable
    set graceful-restart enable
    set additional-path-select 255
    config neighbor-group
        edit "VPN5"
            set capability-graceful-restart enable
            set link-down-failover enable
            set next-hop-self enable
            set soft-reconfiguration enable
            set remote-as 65001
            set route-map-in "Route-Map-In"
            set additional-path send
            set route-reflector-client enable
        next
        edit "VPN6"
            set capability-graceful-restart enable
            set link-down-failover enable
            set next-hop-self enable
            set soft-reconfiguration enable
            set remote-as 65001
            set route-map-in "Route-Map-In"
            set additional-path send
            set route-reflector-client enable
        next
    end
    config neighbor-range
        edit 1
            set prefix 10.50.5.0 255.255.255.0
            set neighbor-group "VPN5"
        next
        edit 2
            set prefix 10.50.6.0 255.255.255.0
            set neighbor-group "VPN6"
        next
    end
    config network
        edit 1
            set prefix 10.10.200.0 255.255.255.0
        next
    end
    config redistribute "connected"
        set route-map "port3_only"
    end
end
```

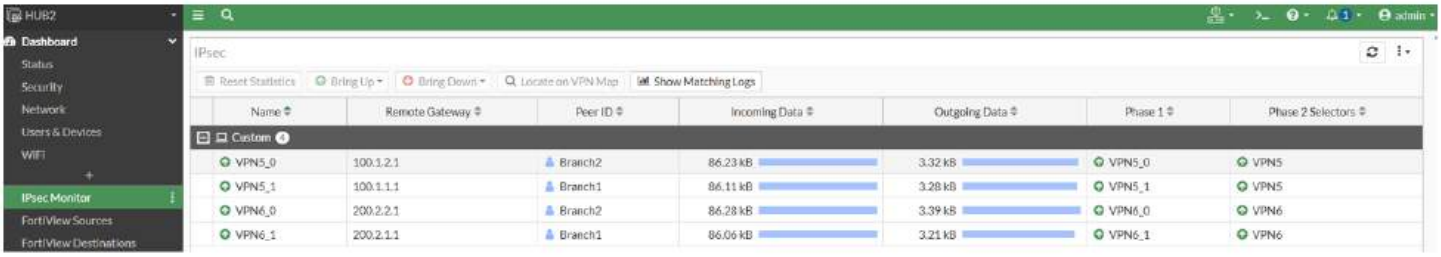
Branches IPsec Verification





HUBs IPsec Verification





Branches Routing Verification

```
BR1 # get router info bgp  summary

VRF 0 BGP router identifier 10.100.100.1, local AS number 65001
BGP table version is 2
1 BGP AS-PATH entries
1 BGP community entries

Neighbor  V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.50.1.1 4      65001      29      27        1    0    0 00:20:05        3
10.50.2.1 4      65001      30      27        1    0    0 00:20:06        3
10.50.5.1 4      65001      30      26        1    0    0 00:20:05        3
10.50.6.1 4      65001      29      26        1    0    0 00:20:05        3

Total number of neighbors 4


BR1 #  get router info routing-table bgp
Routing table for VRF=0
B       10.10.2.0/24 [200/0] via 10.50.1.10 [2] (recursive is directly connected, HUB1-VPN1), 00:19:46, [1/0]
                     [200/0] via 10.50.2.10 [2] (recursive is directly connected, HUB1-VPN2), 00:19:46, [1/0]
                     [200/0] via 10.50.5.10 [2] (recursive is directly connected, HUB2-VPN1), 00:19:46, [1/0]
                     [200/0] via 10.50.6.10 [2] (recursive is directly connected, HUB2-VPN2), 00:19:46, [1/0]
B       10.10.100.0/24 [200/0] via 10.50.1.1 (recursive via HUB1-VPN1 tunnel 100.1.100.1), 00:20:07, [1/0]
                       [200/0] via 10.50.2.1 (recursive via HUB1-VPN2 tunnel 200.2.100.1), 00:20:07, [1/0]
B       10.10.200.0/24 [200/0] via 10.50.5.1 (recursive via HUB2-VPN1 tunnel 100.1.200.1), 00:20:08, [1/0]
                       [200/0] via 10.50.6.1 (recursive via HUB2-VPN2 tunnel 200.2.200.1), 00:20:08, [1/0]
```

```
BR2 # get router info bgp  summary

VRF 0 BGP router identifier 10.100.100.2, local AS number 65001
BGP table version is 2
1 BGP AS-PATH entries
1 BGP community entries

Neighbor  V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.50.1.1 4      65001      29      26        1    0    0 00:20:09        3
10.50.2.1 4      65001      29      27        1    0    0 00:20:10        3
10.50.5.1 4      65001      29      26        1    0    0 00:20:09        3
10.50.6.1 4      65001      29      26        1    0    0 00:20:09        3

Total number of neighbors 4


BR2 #  get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 10.50.1.11 [2] (recursive is directly connected, HUB1-VPN1), 00:19:46, [1/0]
                     [200/0] via 10.50.2.11 [2] (recursive is directly connected, HUB1-VPN2), 00:19:46, [1/0]
                     [200/0] via 10.50.5.11 [2] (recursive is directly connected, HUB2-VPN1), 00:19:46, [1/0]
                     [200/0] via 10.50.6.11 [2] (recursive is directly connected, HUB2-VPN2), 00:19:46, [1/0]
B       10.10.100.0/24 [200/0] via 10.50.1.1 (recursive via HUB1-VPN1 tunnel 100.1.100.1), 00:20:12, [1/0]
                       [200/0] via 10.50.2.1 (recursive via HUB1-VPN2 tunnel 200.2.100.1), 00:20:12, [1/0]
B       10.10.200.0/24 [200/0] via 10.50.5.1 (recursive via HUB2-VPN1 tunnel 100.1.200.1), 00:20:12, [1/0]
                       [200/0] via 10.50.6.1 (recursive via HUB2-VPN2 tunnel 200.2.200.1), 00:20:12, [1/0]
```

HUBs routing Verification

```
HUB1 # get router info bgp  summary

VRF 0 BGP router identifier 10.100.100.100, local AS number 65001
BGP table version is 2
1 BGP AS-PATH entries
2 BGP community entries
Next peer check timer due in 29 seconds

Neighbor   V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.50.1.10 4      65001      54      60        2    0    0 00:08:38        1
10.50.1.11 4      65001      55      61        2    0    0 00:08:40        1
10.50.2.10 4      65001      56      60        2    0    0 00:08:38        1
10.50.2.11 4      65001      56      60        2    0    0 00:08:40        1

Total number of neighbors 4


HUB1 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 10.50.1.11 (recursive is directly connected, VPN1), 00:08:32, [1/0]
                     [200/0] via 10.50.2.11 (recursive is directly connected, VPN2), 00:08:32, [1/0]
B       10.10.2.0/24 [200/0] via 10.50.1.10 (recursive is directly connected, VPN1), 00:08:32, [1/0]
                     [200/0] via 10.50.2.10 (recursive is directly connected, VPN2), 00:08:32, [1/0]


HUB1 # get router info bgp network
VRF 0 BGP table version is 2, local router ID is 10.100.100.100
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.10.1.0/24     10.50.1.11       0             100      0        1 i <-/1>
*>i                 10.50.2.11       0             100      0        2 i <-/2>
*>i10.10.2.0/24     10.50.1.10       0             100      0        1 i <-/1>
*>i                 10.50.2.10       0             100      0        2 i <-/2>
*> 10.10.100.0/24   0.0.0.0                        100  32768        0 i <-/1>

Total number of prefixes 3
```

```
HUB2 # get router info bgp  summary

VRF 0 BGP router identifier 10.100.100.200, local AS number 65001
BGP table version is 2
1 BGP AS-PATH entries
2 BGP community entries
Next peer check timer due in 21 seconds

Neighbor   V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.50.5.10 4      65001      59      70        2    0    0 00:02:31        1
10.50.5.11 4      65001      58      72        2    0    0 00:02:29        1
10.50.6.10 4      65001      58      72        2    0    0 00:02:34        1
10.50.6.11 4      65001      57      71        2    0    0 00:02:32        1

Total number of neighbors 4


HUB2 # get router info routing-table bgp
Routing table for VRF=0
B       10.10.1.0/24 [200/0] via 10.50.5.11 (recursive is directly connected, VPN5), 00:02:35, [1/0]
                     [200/0] via 10.50.6.11 (recursive is directly connected, VPN6), 00:02:35, [1/0]
B       10.10.2.0/24 [200/0] via 10.50.5.10 (recursive is directly connected, VPN5), 00:02:37, [1/0]
                     [200/0] via 10.50.6.10 (recursive is directly connected, VPN6), 00:02:37, [1/0]


HUB2 # get router info bgp network
VRF 0 BGP table version is 2, local router ID is 10.100.100.200
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.10.1.0/24     10.50.5.11       0             100      0        5 i <-/1>
*>i                 10.50.6.11       0             100      0        6 i <-/2>
*>i10.10.2.0/24     10.50.5.10       0             100      0        5 i <-/1>
*>i                 10.50.6.10       0             100      0        6 i <-/2>
*> 10.10.200.0/24   0.0.0.0                        100  32768        0 i <-/1>

Total number of prefixes 3
```

The Overlay template should be deleted after customizations are done to the BGP, IPSec & CLI Templates because running it again will reconfigure the templates to the default provisioning setting.

1. Which template is not provisioned by the Overlay template.
   a) SDWAN
   b) BGP
   c) CLI
   d) IPsec

2. What configuration option is available within the Overlay template Wizard.
   a) Route-map
   b) Community list
   c) Health checks
   d) All the above

3. Why is it recommended to delete the overlay template after configurations are complete.
   a) If it is not deleted you can not modify the SDWAN and BGP templates created
   b) If it runs again, it can reset any customization done to the Templates
   c) All the above

4. How Many VPN tunnels are created per branch with the Single HUB overlay Template.
   a) 2
   b) 4
   c) 8

5. How Many VPN tunnels are created per HUB with the Dual HUB overlay Template.
   a) 2
   b) 4
   c) 8

6. How Many VPN tunnels are created per branch with the Dual HUB overlay Template.
   a) 2
   b) 4
   c) 8

7. Which Metadata variable is created by the overlay template
   a) branch_id
   b) Local_id
   c) Isp_gateway

8. Which device group supports creation without having any device attached and then referenced in the overlay template.
   a) Branches
   b) Hubs
   c) None, all groups must have devices attached.

Answers to End of Chapter Questions

Chapter 2

1.a 2.b 3.b 4.a 5.a 6.a 7.b.

Chapter 3

1.c 2.a 3.c 4.c 5.c 6.a 7.a&b

chapter 4

1.b 2.c 3.c 4.a 5.d

Chapter 5

1.a 2.a&b 3.b 4.a 5.a 6.a 7.a 8.a