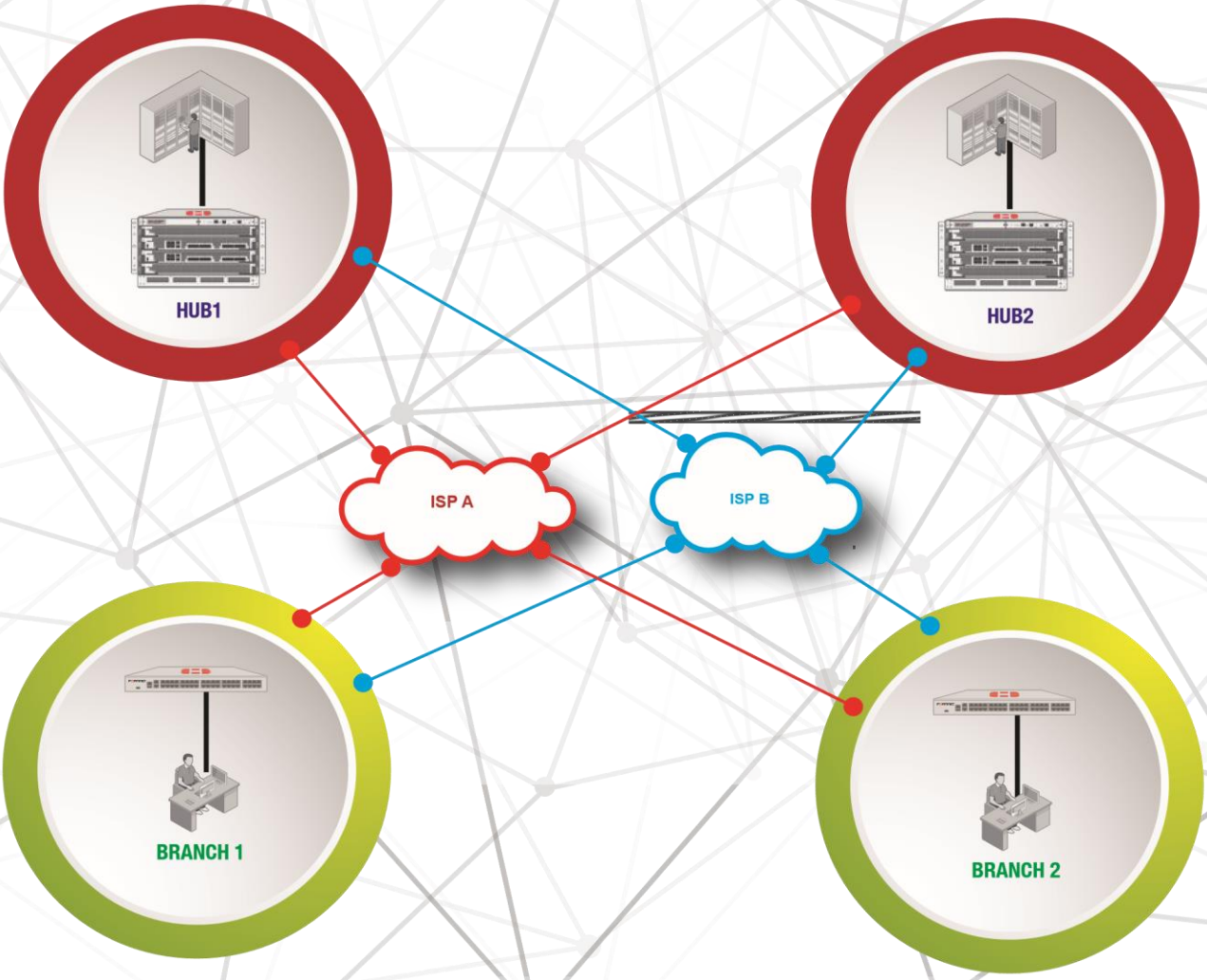
FORTINET SD-WAN DEMYSTIFIED

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.

This book is available for sale at: https://leanpub.com/fortinetsd-wandemystified

Check Linkedin for any coupon links at: https://www.linkedin.com/in/nicanory-ateya/

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.

Contents

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.

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:

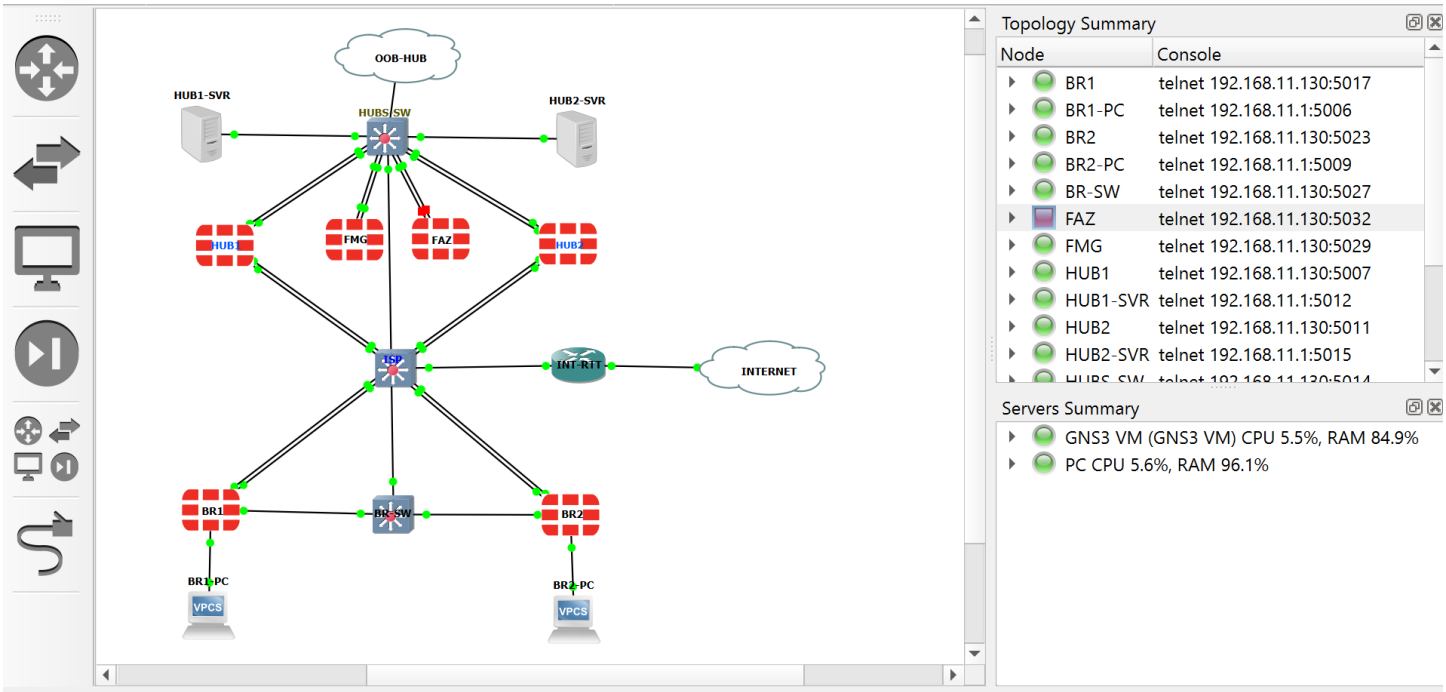
i. Whether the selected interface is meeting the defined SLA
ii. The lowest cost between the selected links.

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

HUB

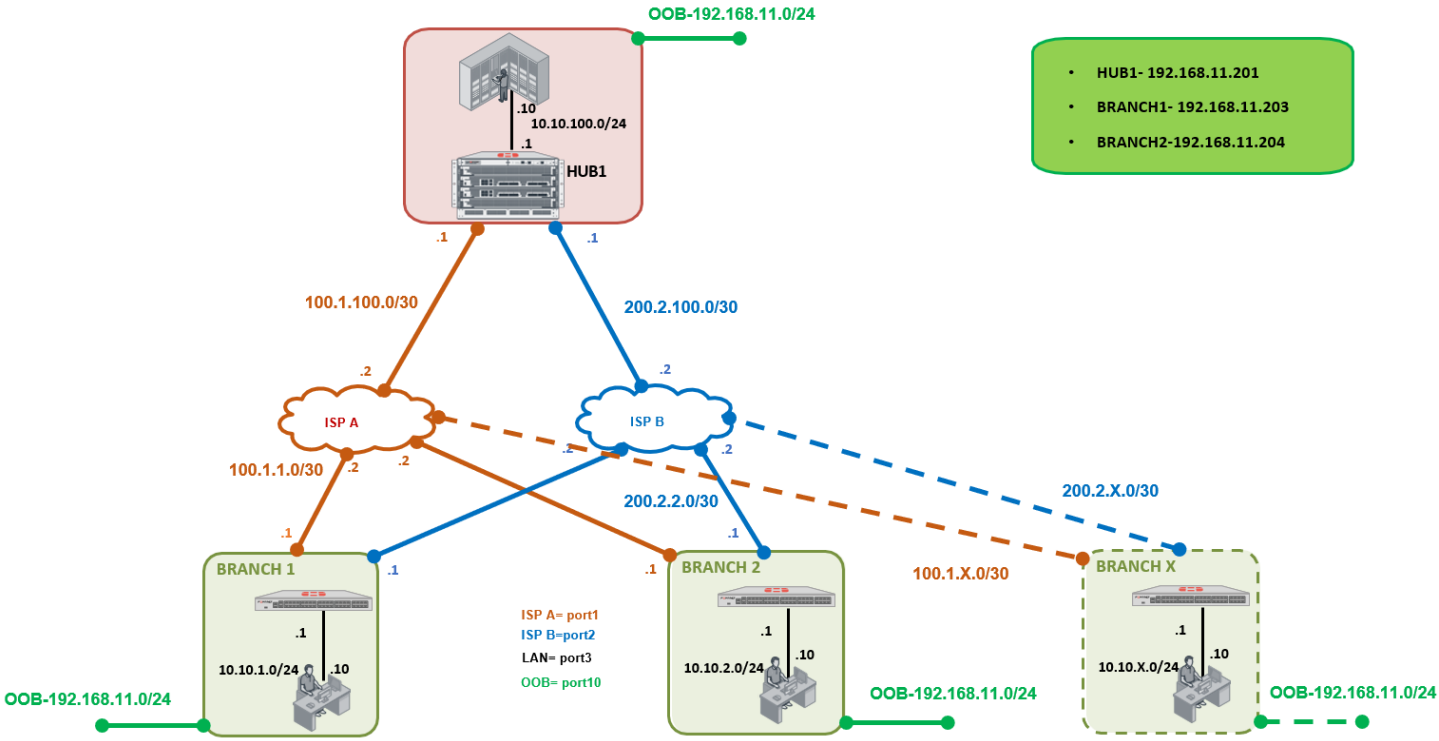
HUB VPN tunnels are named VPNx where X is the tunnel number e.g. VPN1, VPN2 etc.

Branch

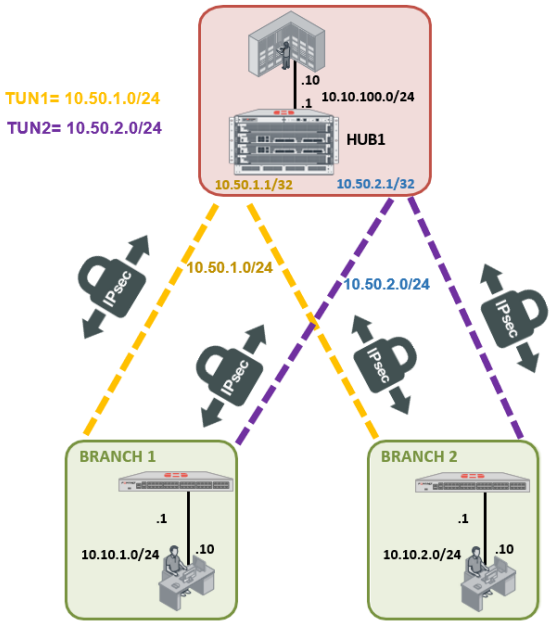
Branch tunnels are named HUBy-VPNx where y is the HUB site identifier and x the tunnel number e.g. The first tunnel to HUB1 would be HUB1-VPN1 while the first to HUB2 will be HUB2-VPN5. This will be elaborated in the LABs

Chapter 2: Deployment using GUI & CLI

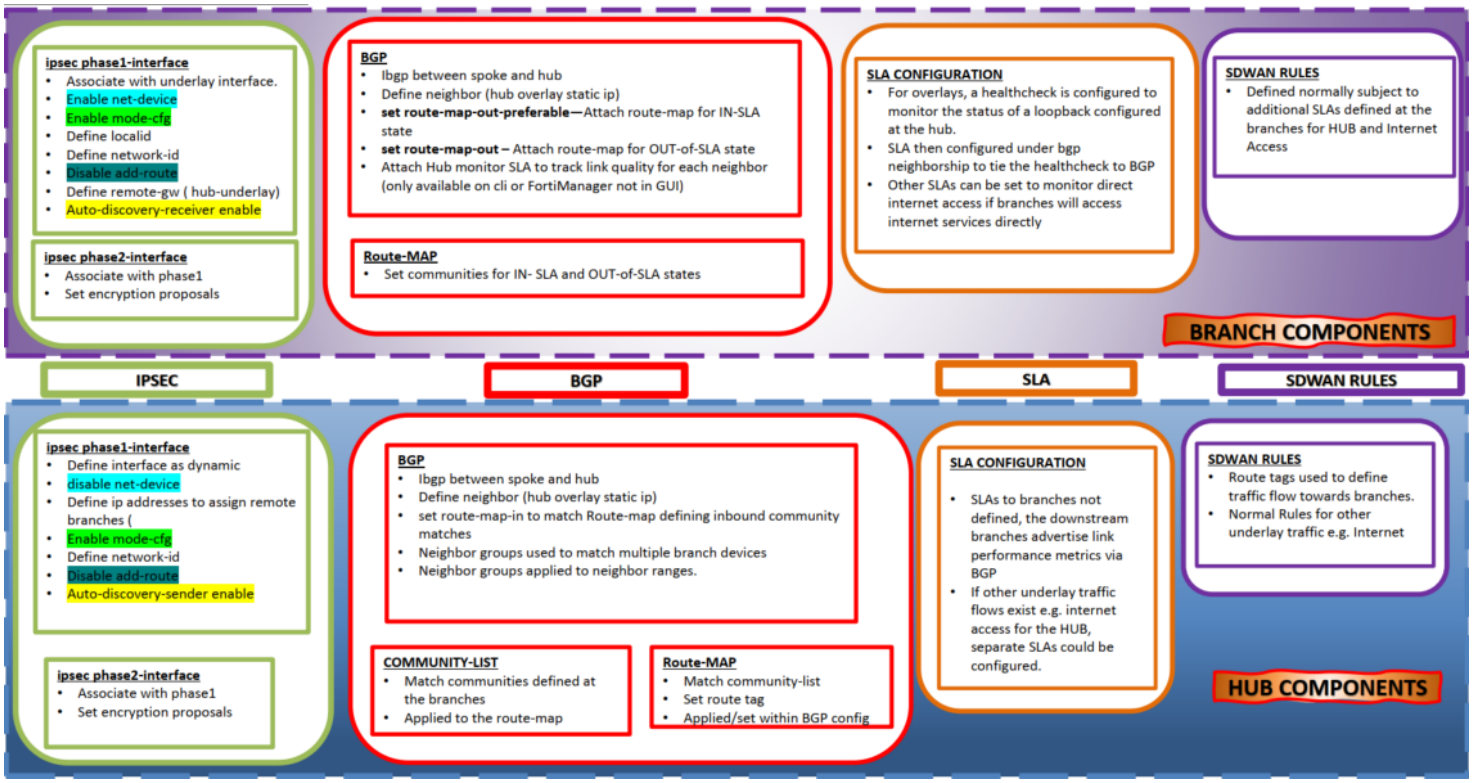
The diagram below represents the reference topology for the GUI & CLI setup made up of two branches and a single hub. Each site has two undelay/ISP links. The underlay path has a single overlay network from the branches to the HUB.



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.



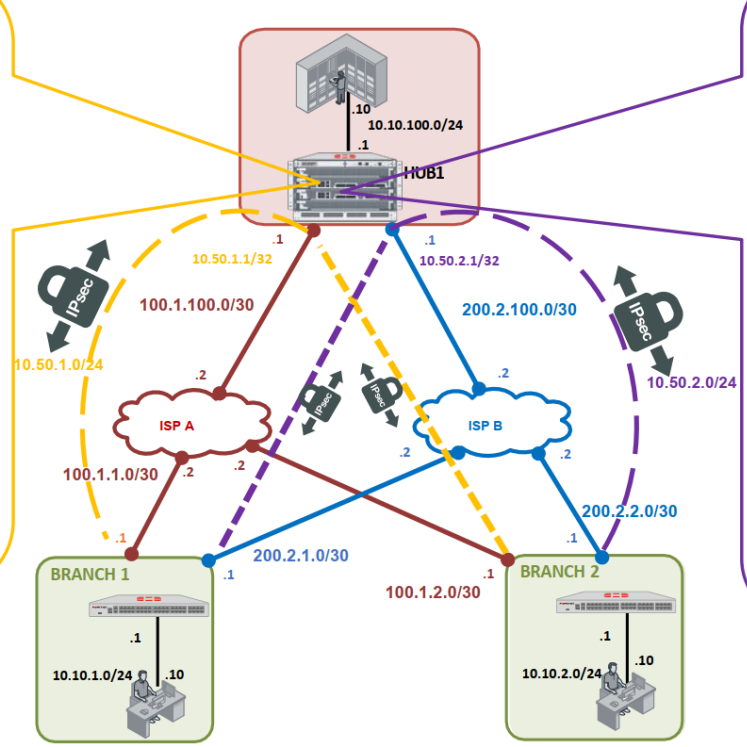
Fortinet's SD-WAN has four main components that are configured at the HUB and respective branches. These include **IPsec, BGP, SLA and SD-WAN rules**. Each has sub-components as highlighted in the diagram below. This will be referenced to configure HUB1 to connect to the two branches via CLI and GUI.



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.

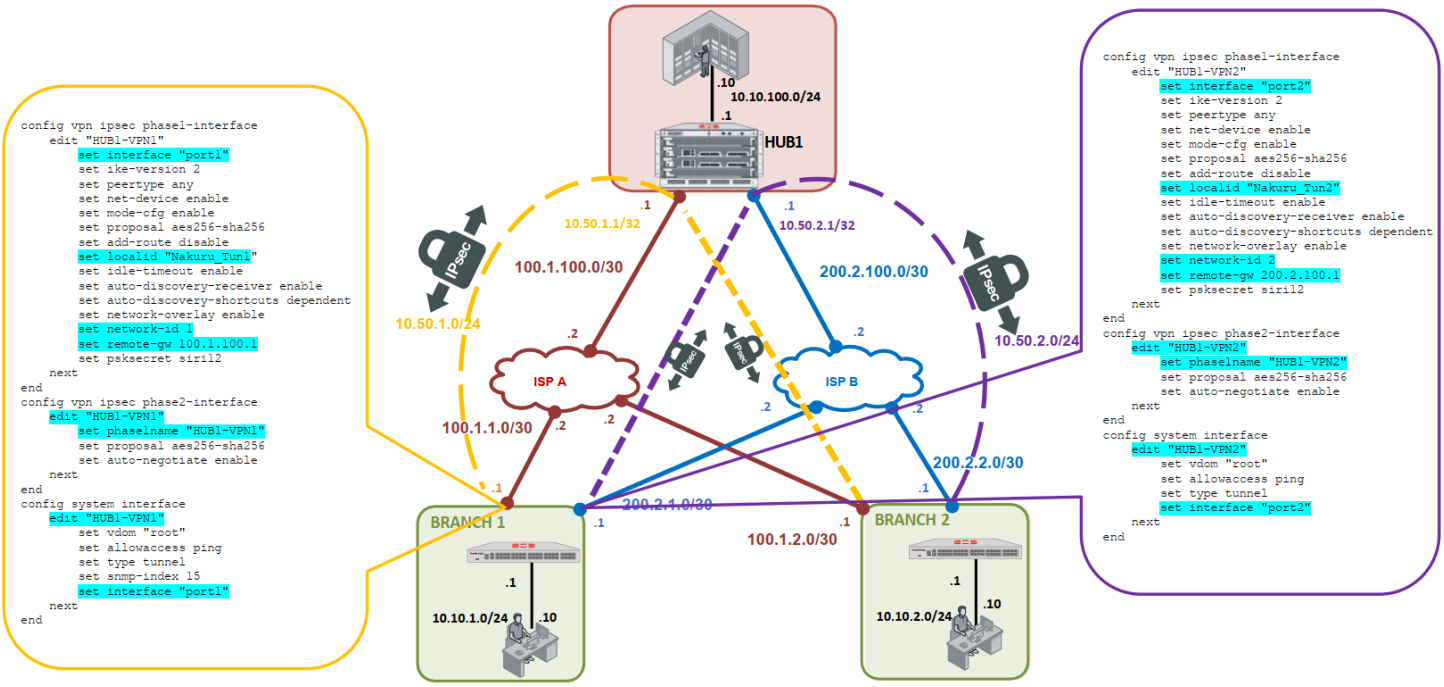
```
config vpn ipsec phase1-interface
    edit "VPN1"
        set type dynamic
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes256-sha256
        set add-route disable
        set dpd on-idle
        set auto-discovery-sender enable
        set network-overlay enable
        set localid NAIROBI-TUN1
        set network-id 1
        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 siril2
        set dpd-retryinterval 60
    next
end
config vpn ipsec phase2-interface
    edit "VPN1"
        set phaselname "VPN1"
        set proposal aes256-sha256
    next
end
config system interface
    edit "VPN1"
        set vdom "root"
        set ip 10.50.1.1 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 10.50.1.253 255.255.255.0
        set interface "port1"
    next
end
```

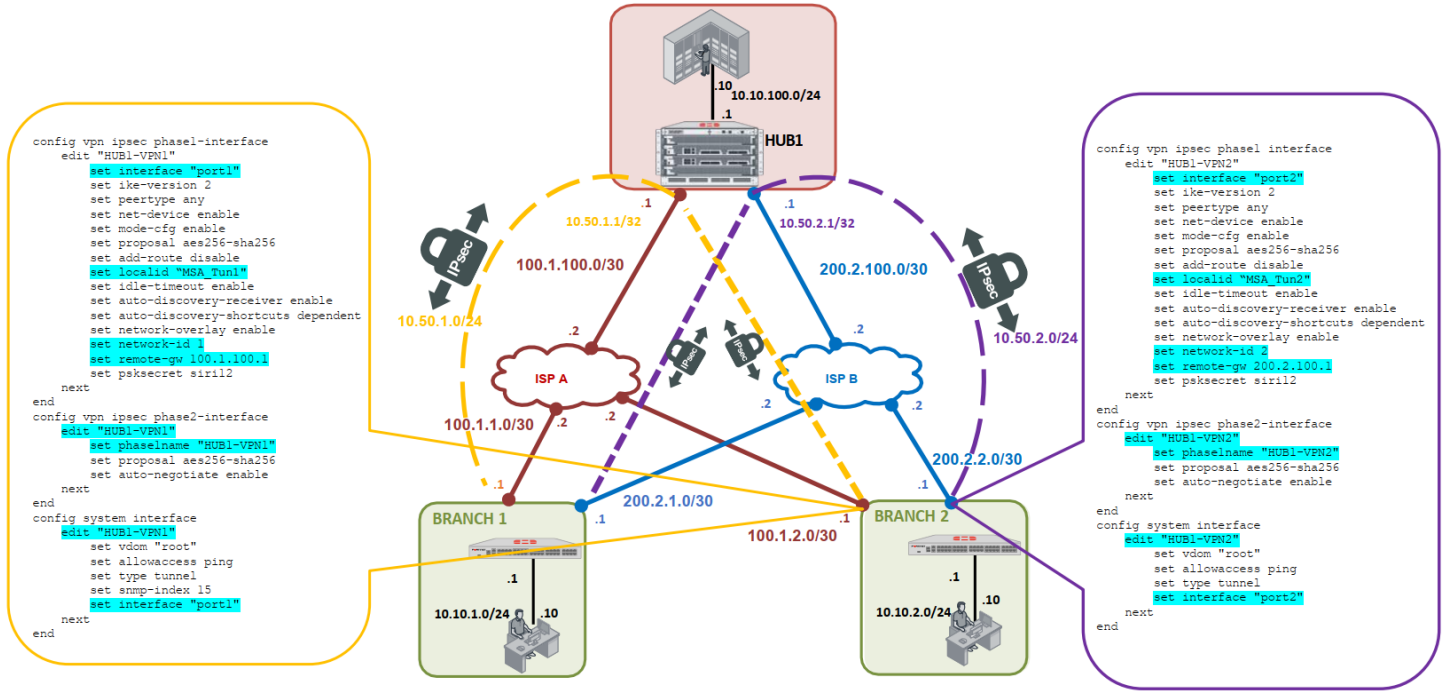



```
config vpn ipsec phase1-interface
    edit "VPN2"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes256-sha256
        set add-route disable
        set dpd on-idle
        set auto-discovery-sender enable
        set network-overlay enable
        set localid NAIROBI-TUN2
        set network-id 2
        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 siril2
        set dpd-retryinterval 60
    next
end
config vpn ipsec phase2-interface
    edit "VPN2"
        set phaselname "VPN2"
        set proposal aes256-sha256
    next
end
config system interface
    edit "VPN2"
        set vdom "root"
        set ip 10.50.2.1 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 10.50.2.253 255.255.255.0
        set interface "port2"
    next
end
```

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
```

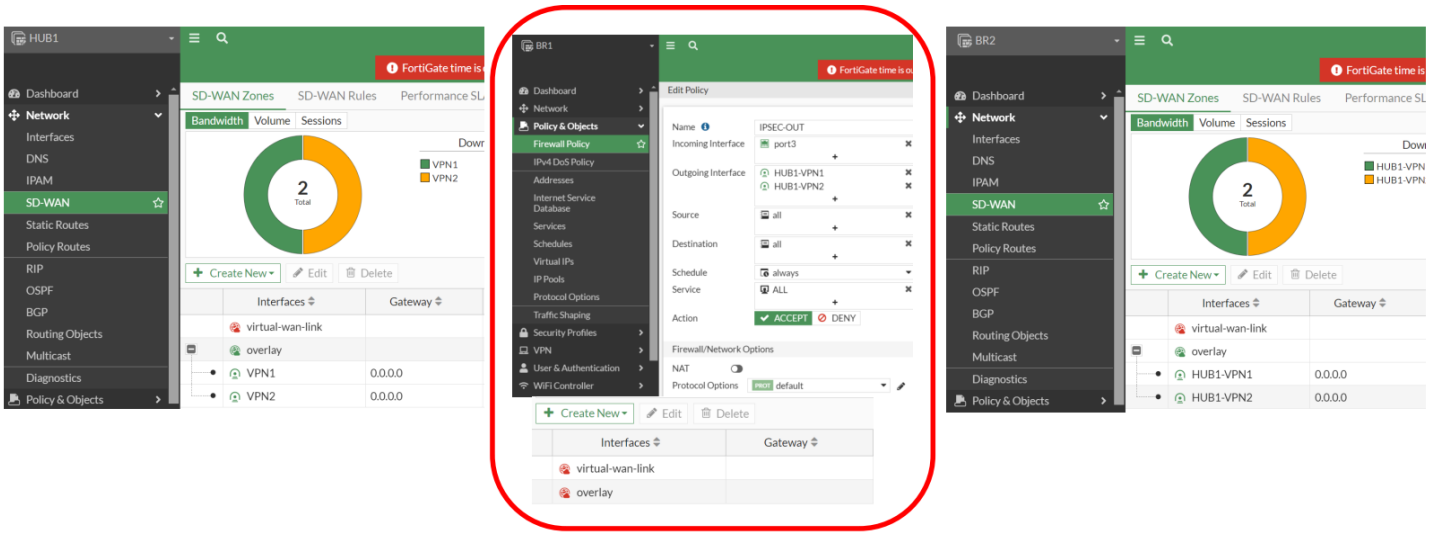
Enable the debugs above to check the VPN state. The output below is given when there is a network –ID mismatch between the HUB and branch i.e. **no proposal chosen** with the debugs done at the HUB.

```
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:

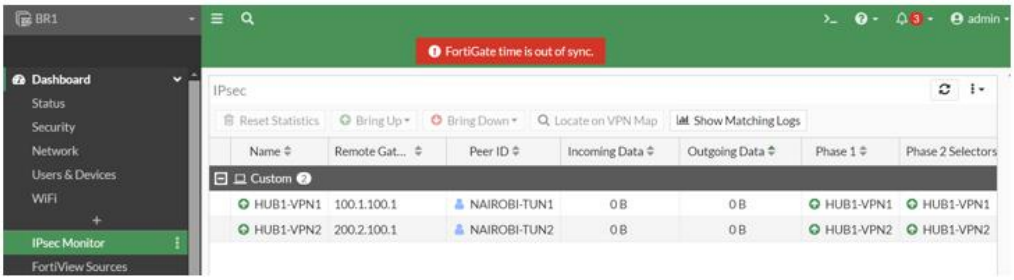
- Configure an outbound policy at the branch referencing the VPN tunnels as the destination and an inbound policy at the HUB with the VPN interfaces as the source interface.
- Bundle the VPN interfaces at the branches and HUB in respective SDWAN zones.
- A combination of the two, on side with the policy,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

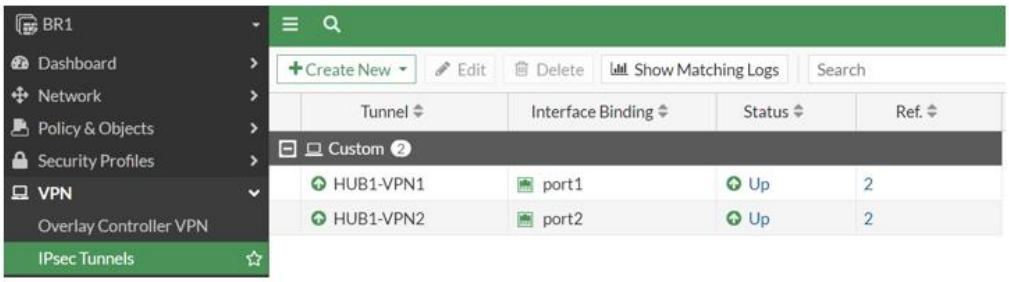


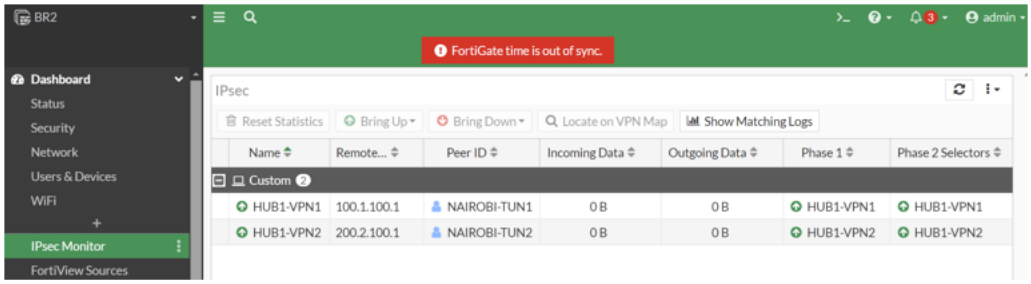
The setup above will not require policies at HUB1 and BR2 since they have SDWAN zones configured while BR1 will require outbound firewall policies to bring up the tunnels because it's HUB1-VPN1 and HUB1-VPN2 interfaces are not included in any SDWAN zone.

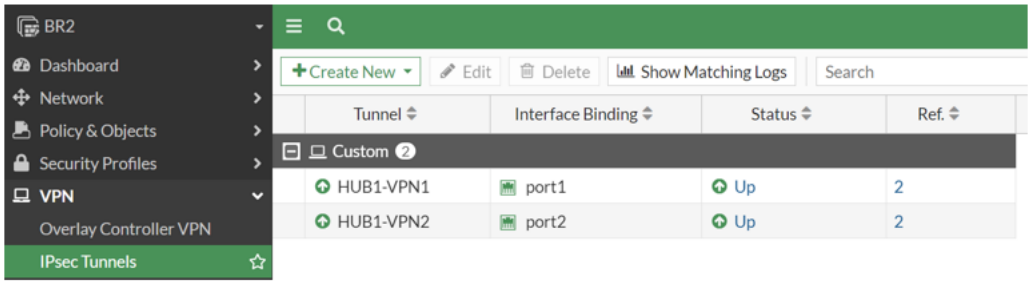
Branches IPSec Monitor

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.



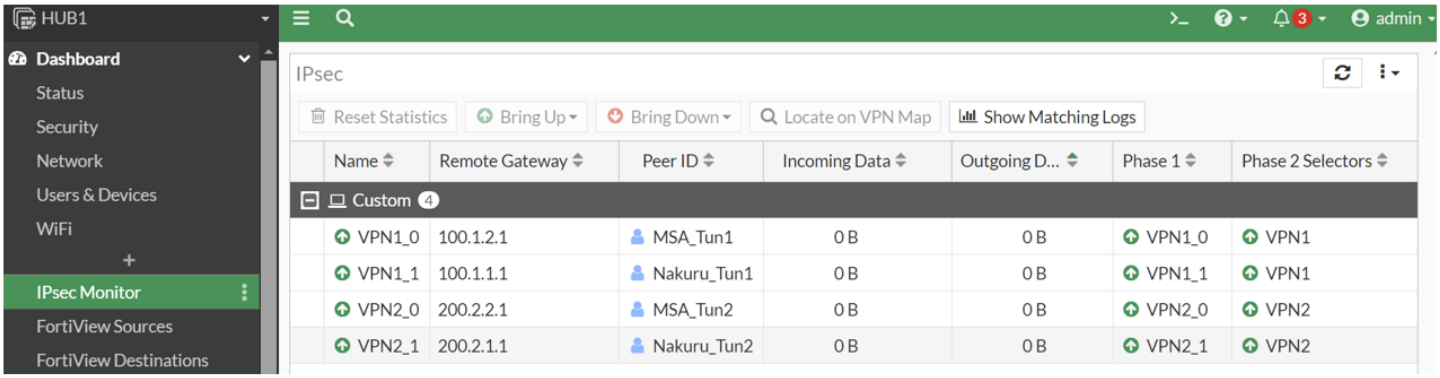


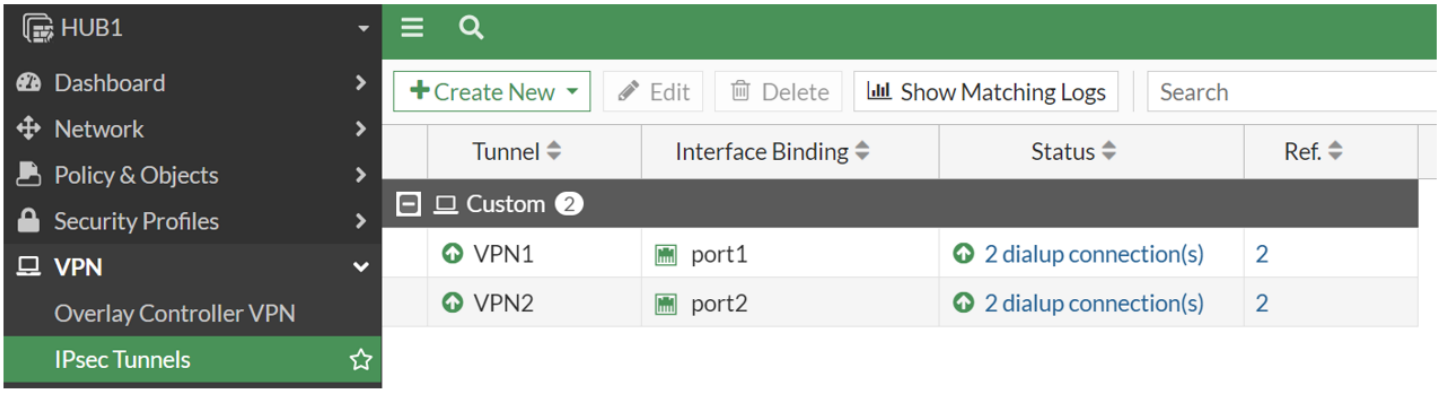




HUB IPSec Monitor

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 foes 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 phase1name "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]
```

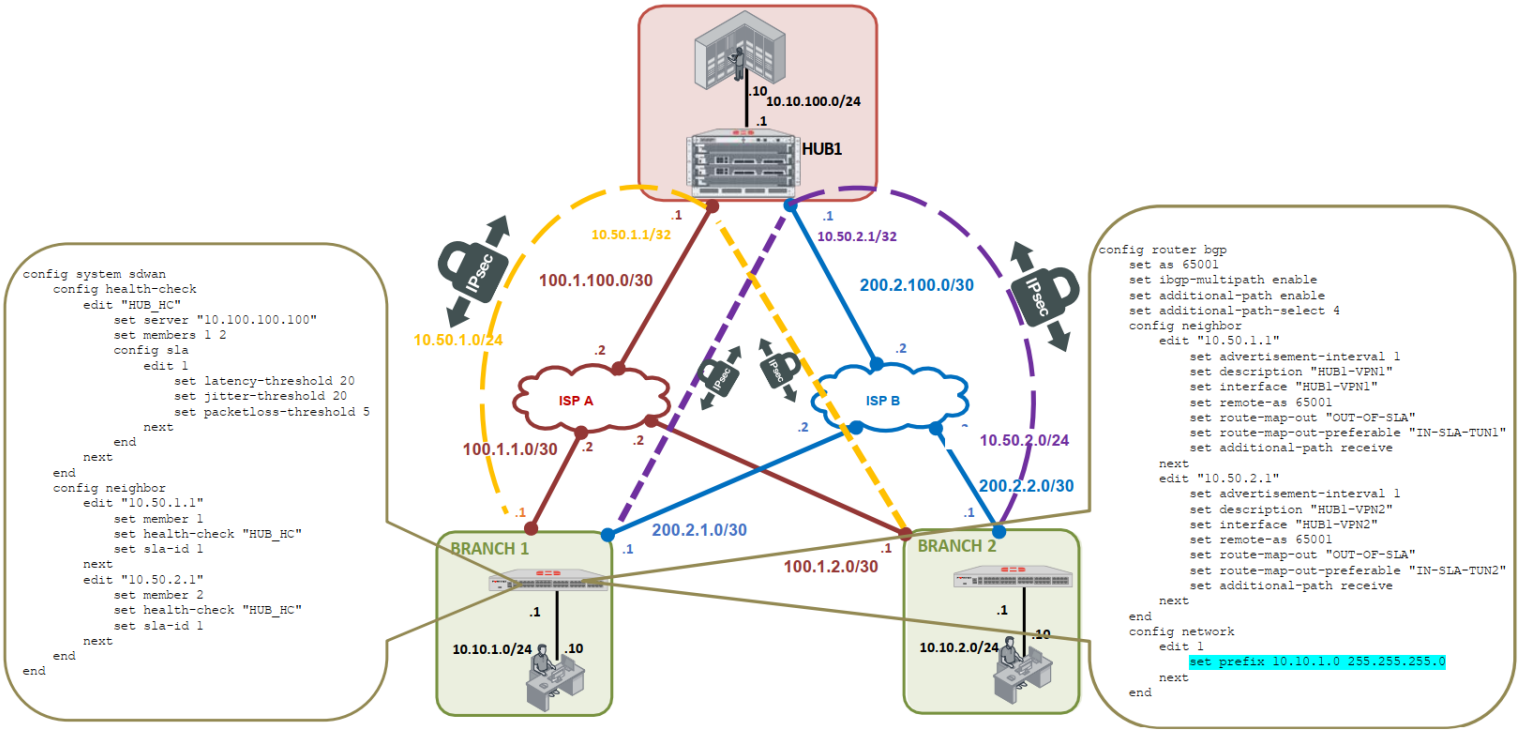
The **`neighbor-ranges`** and **`neighbor-group`** settings are configured for peering relationships to be established without defining individual peers. Branches have their tunnel interfaces configured within the range of the BGP peer and match the defined prefix under the neighbor-range defined by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

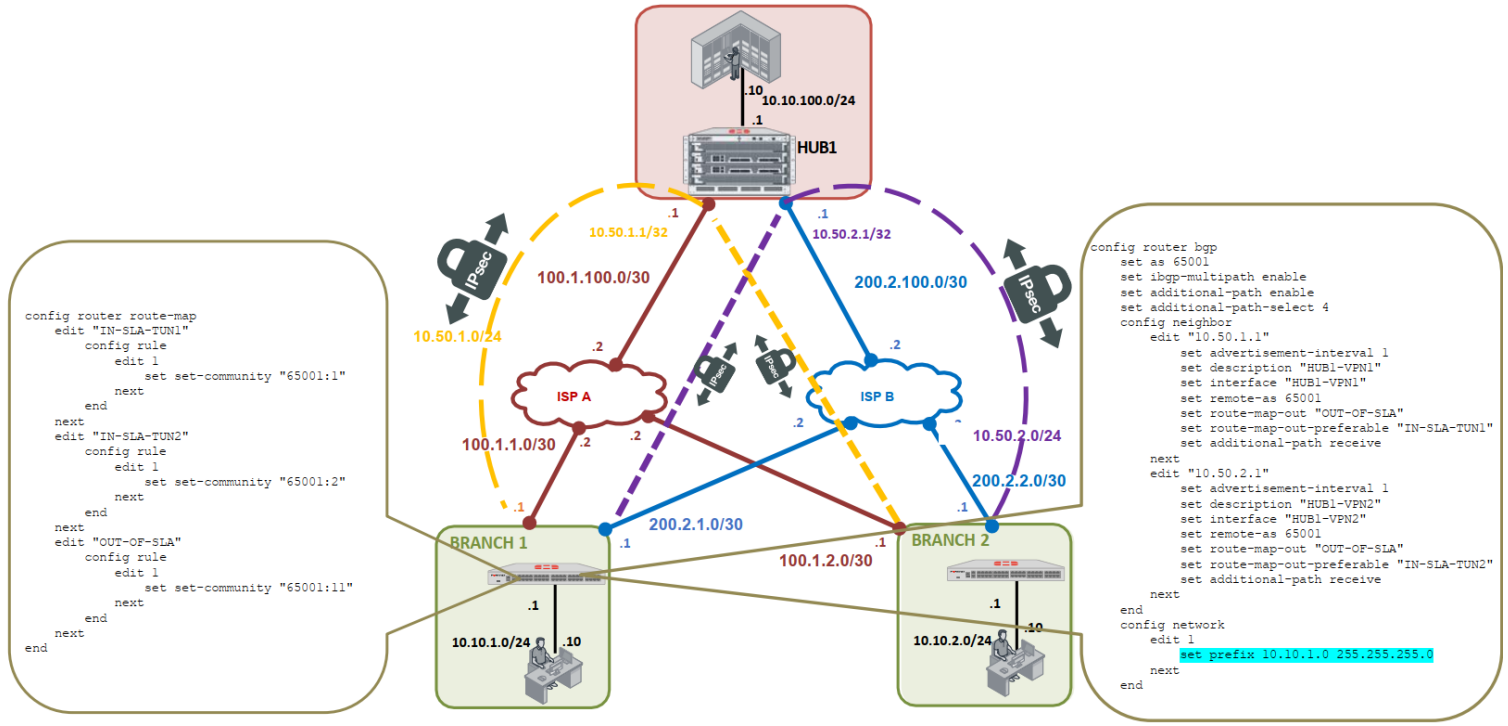
```
set prefix
```

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

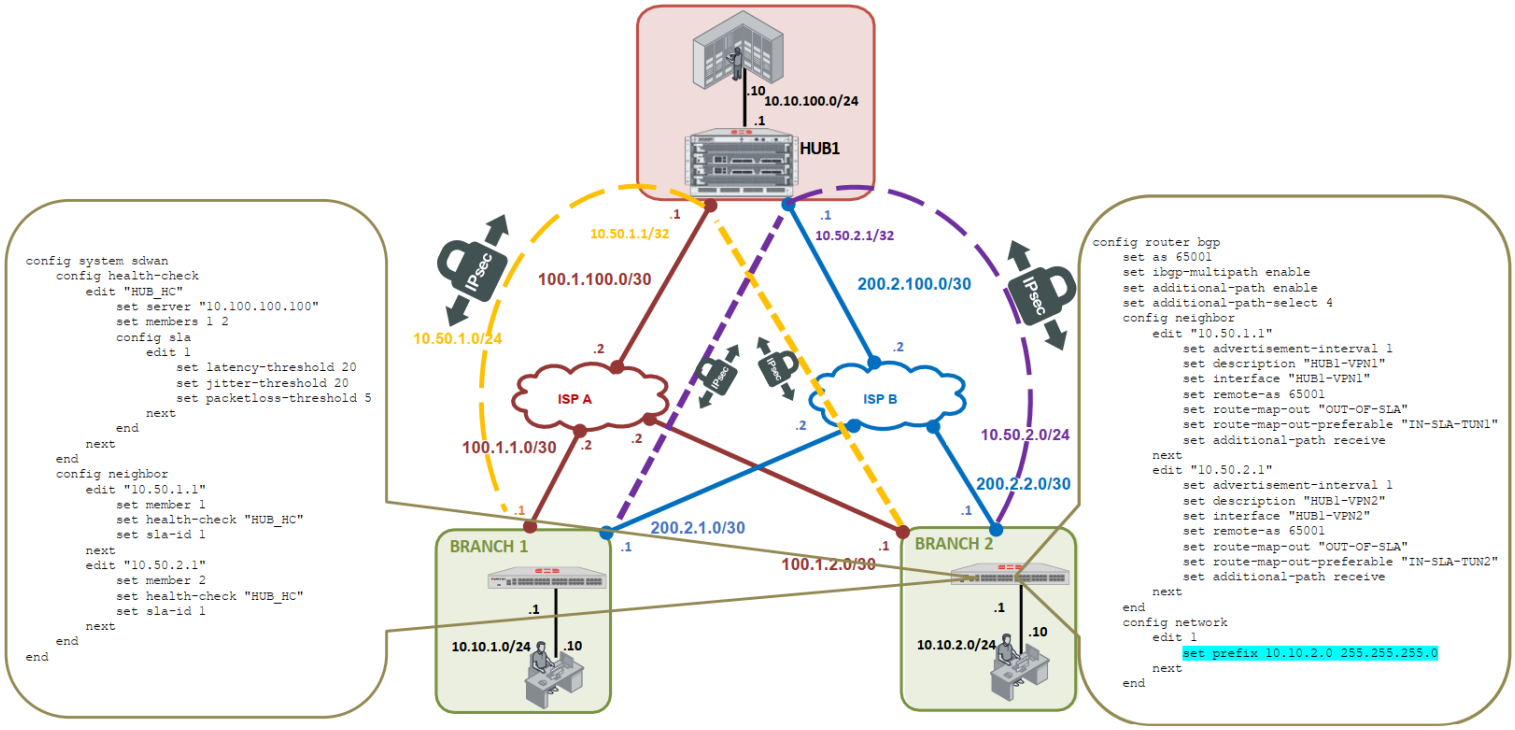
The diagrams below include the required commands to setup BGP at branch1. It includes neighbor definition, where the VPNx ip addresses are referenced as neighbor ip addresses, route-maps applied with required community settings and the SDWAN healthcheck for the HUB loopback configured.





BRANCH2 BGP Configuration

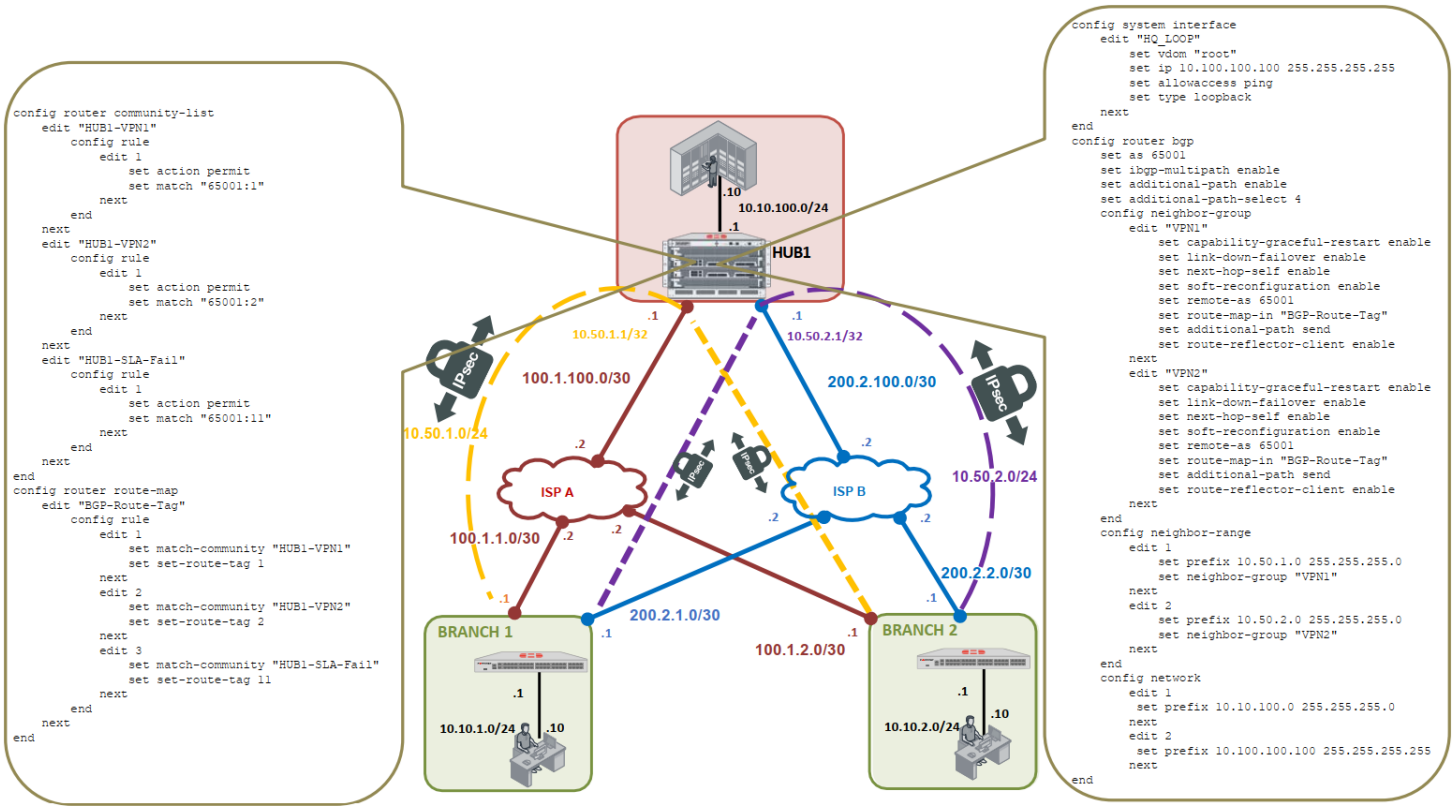
This is the same configuration as branch1 or branchX.




```
config system sdwan
    config health-check
        edit "HUB_HC"
            set server "10.100.100.100"
            set members 1 2
            config sla
                edit 1
                    set latency-threshold 20
                    set jitter-threshold 20
                    set packetloss-threshold 5
                next
            end
        next
    end
    config neighbor
        edit "10.50.1.1"
            set member 1
            set health-check "HUB_HC"
            set sla-id 1
        next
        edit "10.50.2.1"
            set member 2
            set health-check "HUB_HC"
            set sla-id 1
        next
    end
end
```

```
config router bgp
    set as 65001
    set ibgp-multipath enable
    set additional-path enable
    set additional-path-select 4
    config neighbor
        edit "10.50.1.1"
            set advertisement-interval 1
            set description "HUB1-VPN1"
            set interface "HUB1-VPN1"
            set remote-as 65001
            set route-map-out "OUT-OF-SLA"
            set route-map-out-preferable "IN-SLA-TUN1"
            set additional-path receive
        next
        edit "10.50.2.1"
            set advertisement-interval 1
            set description "HUB1-VPN2"
            set interface "HUB1-VPN2"
            set remote-as 65001
            set route-map-out "OUT-OF-SLA"
            set route-map-out-preferable "IN-SLA-TUN2"
            set additional-path receive
        next
    end
    config network
        edit 1
            set prefix 10.10.2.0 255.255.255.0
        next
    end
```

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.

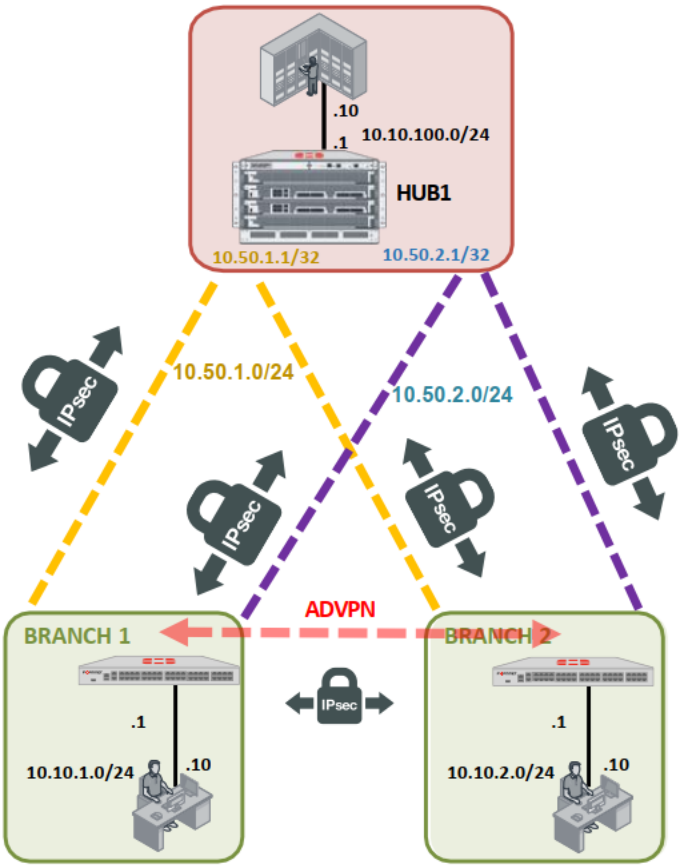
The initial communication goes through the Hub and if properly configured, the hub offers the spoke a shortcut. The spoke then negotiate a new tunnel between them on their overlay network. Once formed a shortcut tunnel would remain up even if the parent IPsec tunnel through the hub went down. This is 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>
```



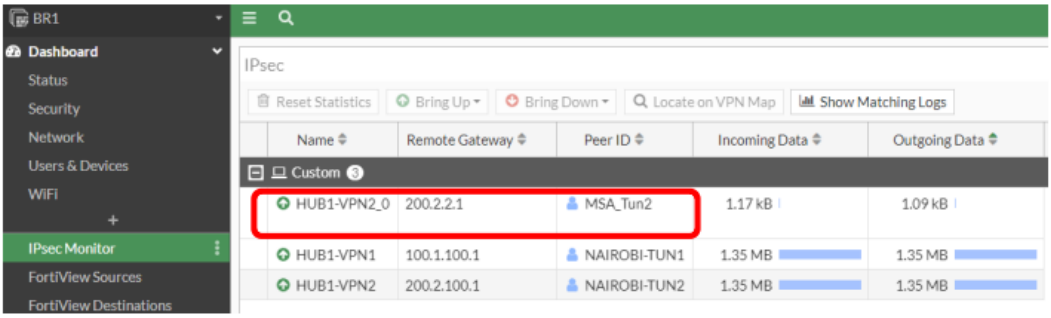
ADVPN Verification

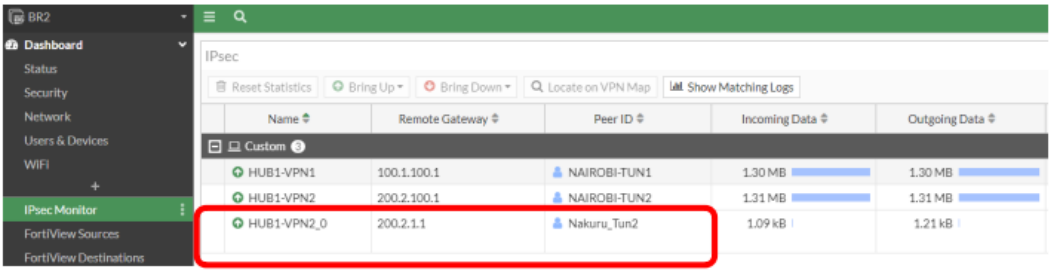
A new connected interface of the connected overlay is created and points to the ADVPN interface. A new static route is formed with the overlay ip for the remote networks with the next hop of the peer overlay IP (Shortcut).

```
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

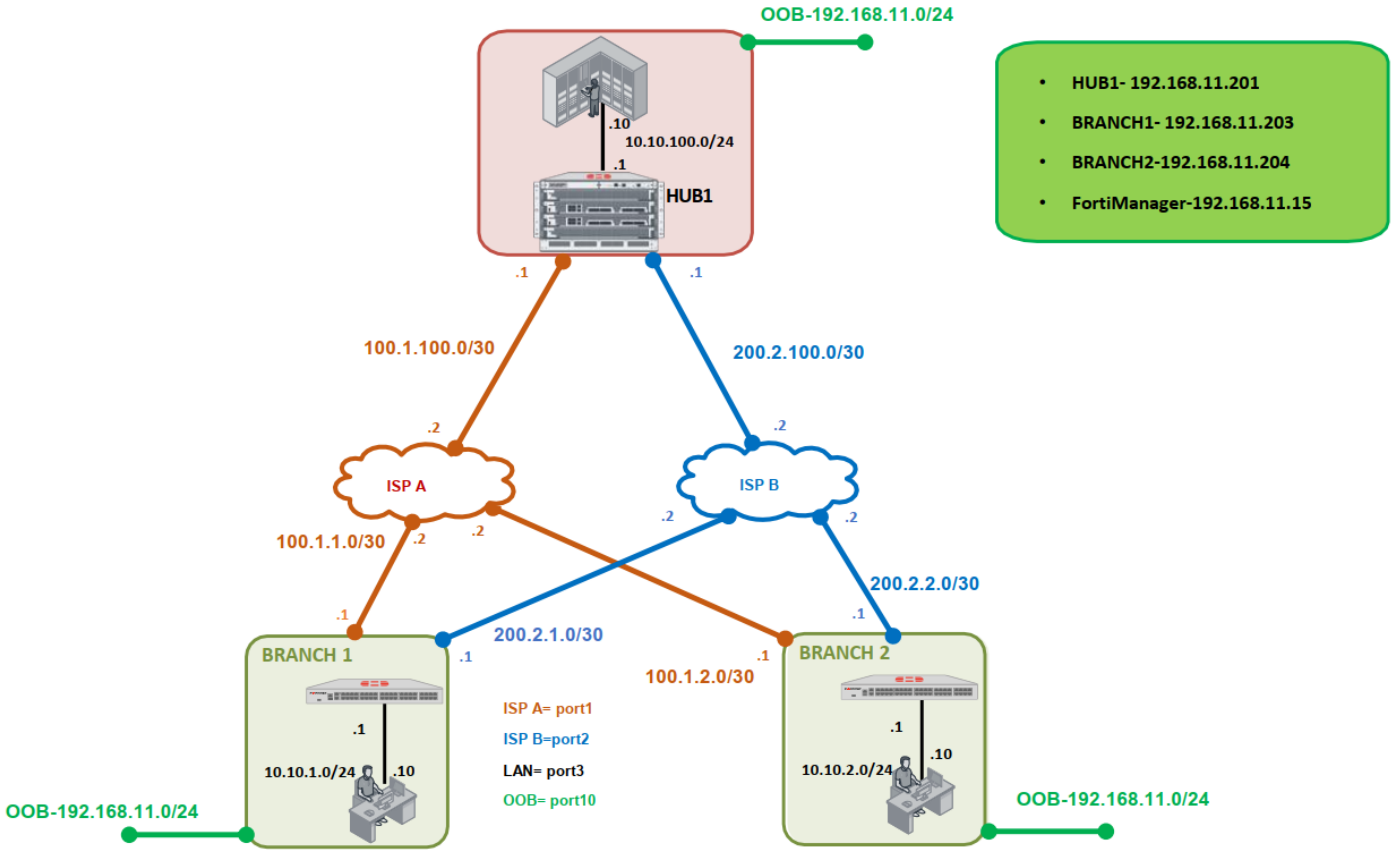
2. Which component must match between the HUB and a spoke/branch.
   a) Local ID
   b) Network ID
   c) All the above

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

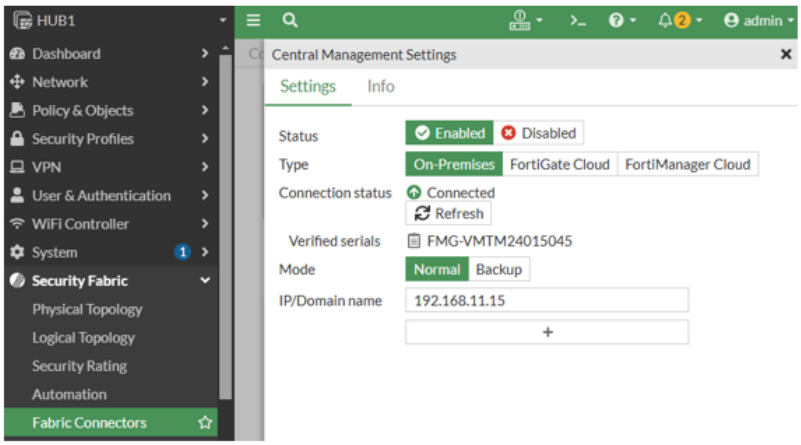
Chapter 3: FortiManager Single Hub with Dual ISP Underlays

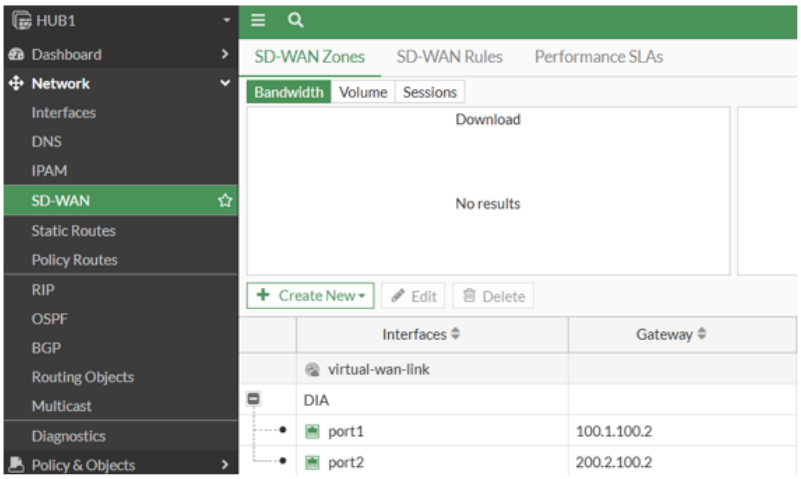
The topology used for this setup is the same as the one used in the GUI & CLI section with the addition of FortiManager VM as for centralized management. FortiManager is deployed at HUB1.

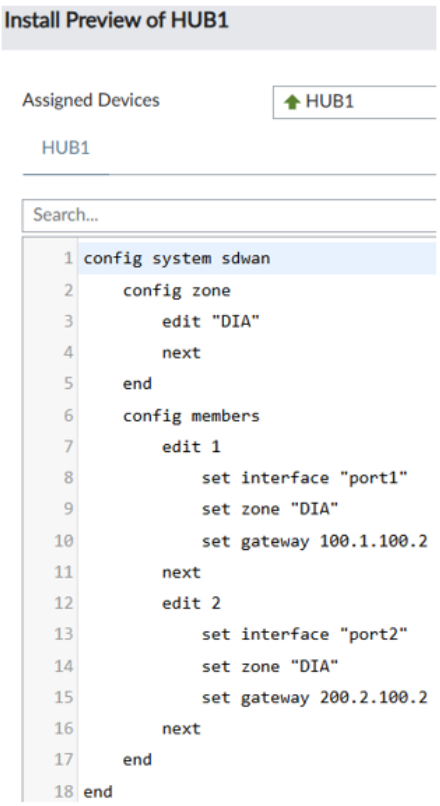


Provision the HUB for FortiManager Access.

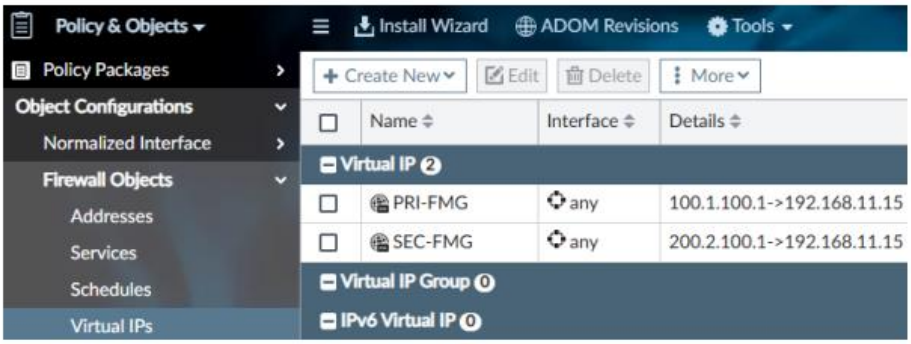
The HUB connects to FortiManager via the OOB network. Configure the underlay SDWAN settings to be used for DIA. The SDWAN configuration is done from FMG but could be done from the HUB GUI as long as the zone name is the same as what would be later used in FortiManager templates to avoid provisioning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

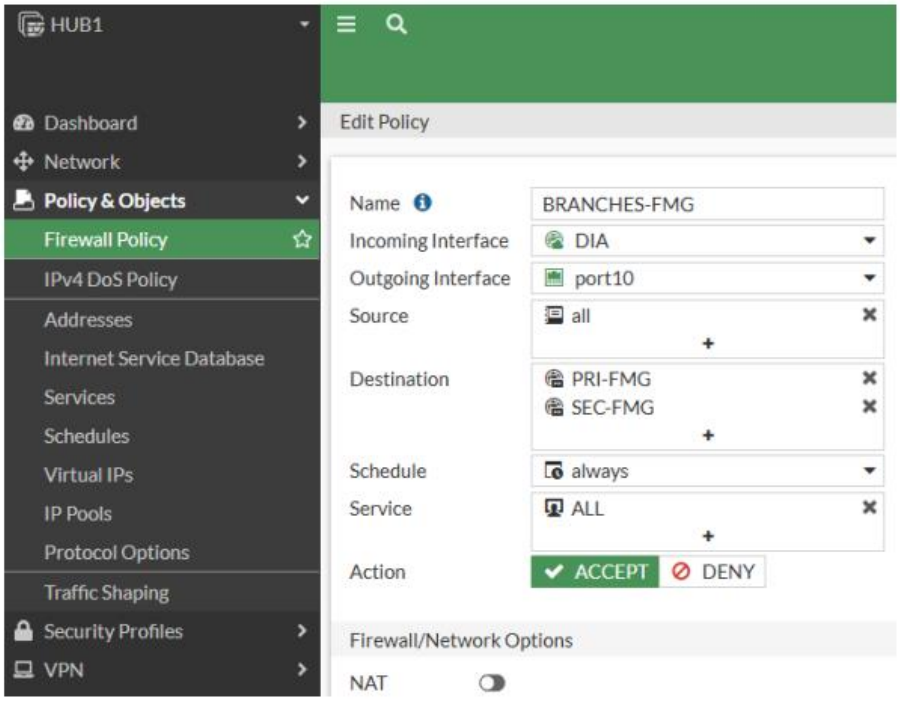




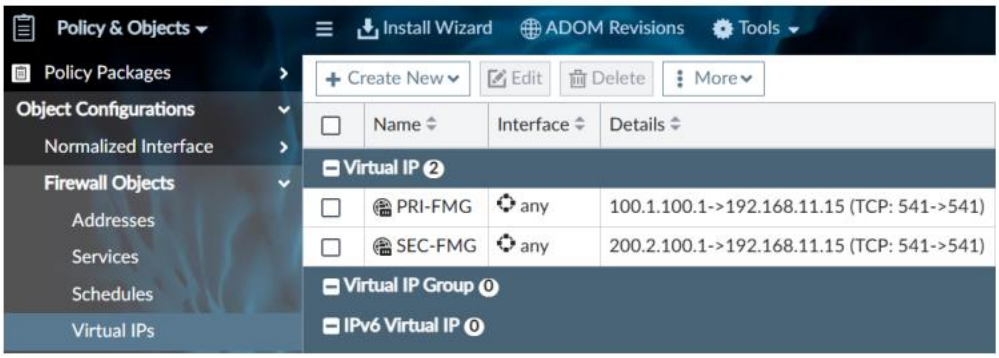


The branches will access FMG via the public IPs at the HUB. Two NAT rules and a firewall policy are required to allow the inbound connections. The inbound rule is applied on the underlay network to allow branch devices to access FortiManager





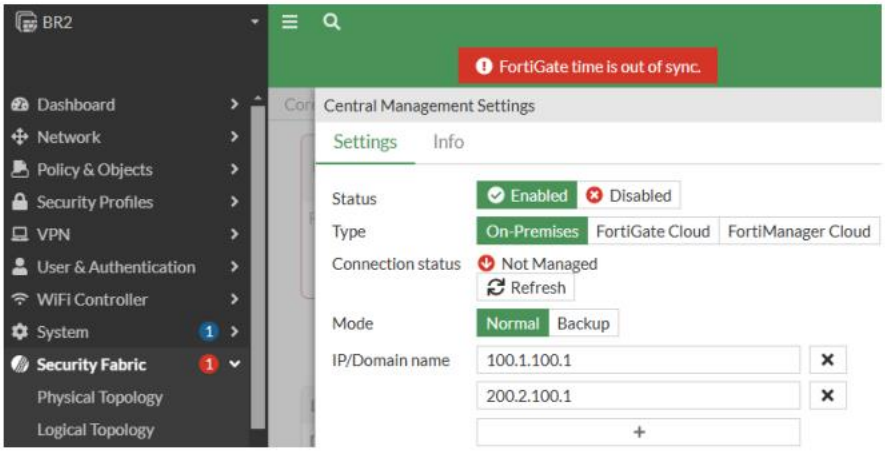
If a single public IP is available at the primary and secondary links, port address translation (PAT) must be used to avoid NAT directing all inbound traffic to FortiManager, blocking all branch devices dialing in over IPsec since all traffic would be directed to FortiManager including IPsec terminations.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



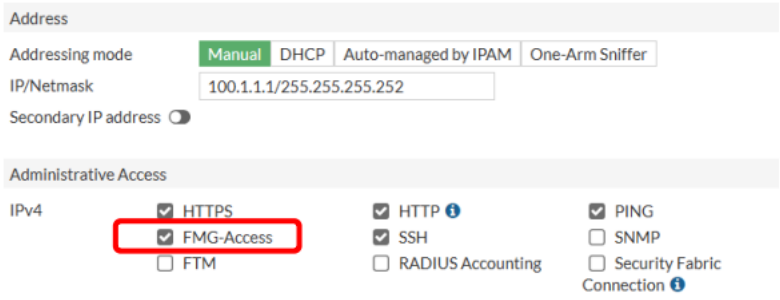
Since there are two ISP links at the HUB, the branches are configured to access both for redundancy. When multiple addresses are listed, the FortiGate attempts to establish the FGFM tunnel using the first IP listed, and if it is unreachable, it tries each subsequent IP until the tunnel is established.

```
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
```

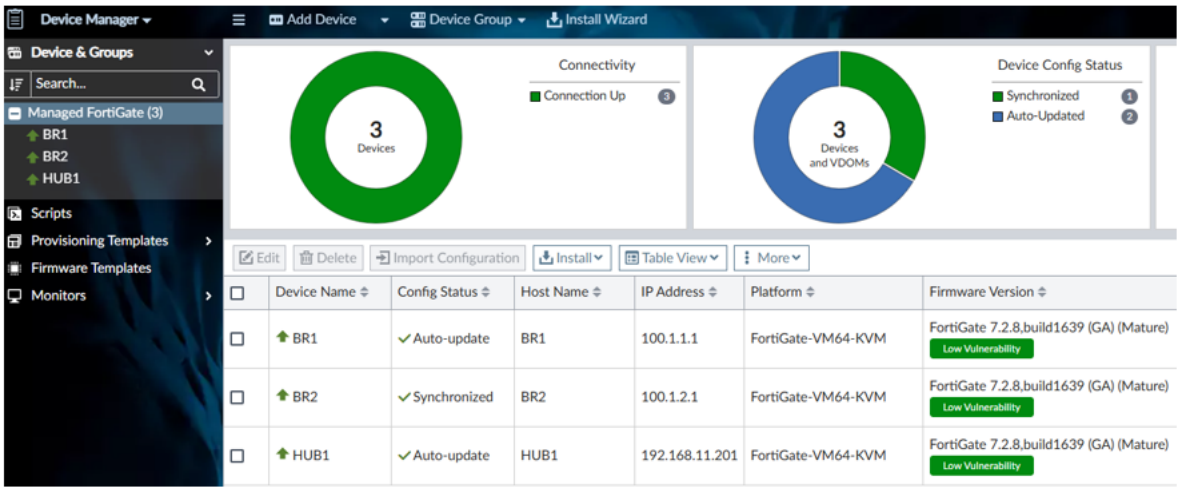


Only one address is used to establish the FGFM tunnel at a time. In FortiManager-HA, when listing multiple management addresses, the first address defines the Primary device and the second address is the Secondary device. This design supports a maximum of 10 FortiManager instances that is adequate for 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)
```

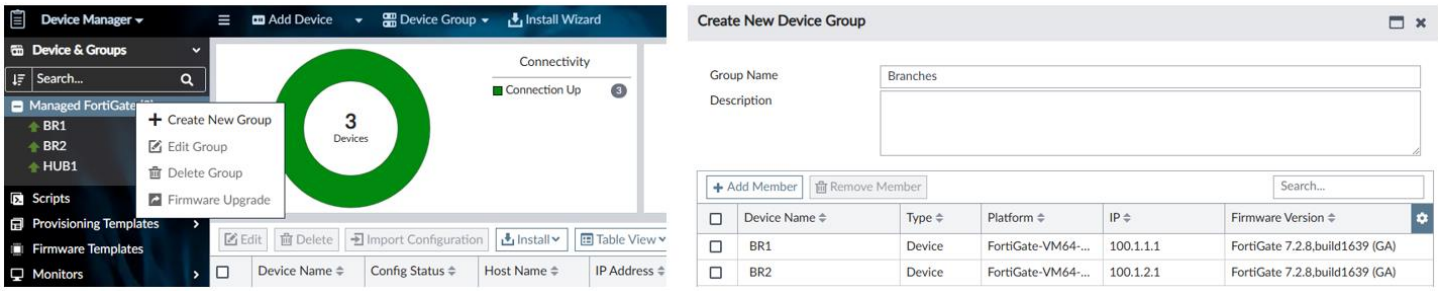
```
BR1 # diagnose fdsm central-mgmt-status
Connection status: Up
Registration status: Registered
Serial: FMG-VMTM24015045
```

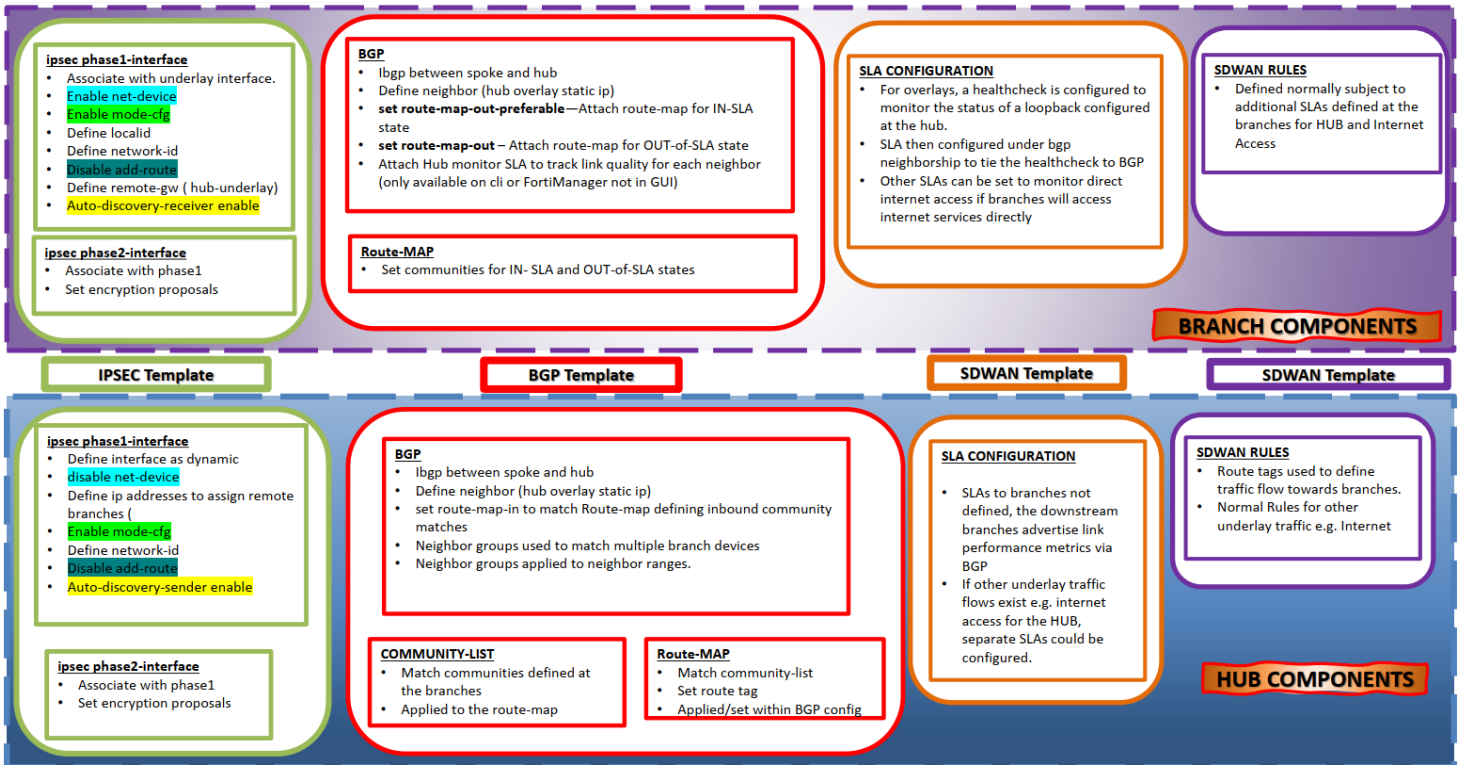
```
BR2 # diagnose fdsm central-mgmt-status
Connection status: Up
Registration status: Registered
Serial: FMG-VMTM24015045
```

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.





This content is not available in the sample book. The book can be purchased on Leanpub at:

https://leanpub.com/fortinetsd-wandemystified

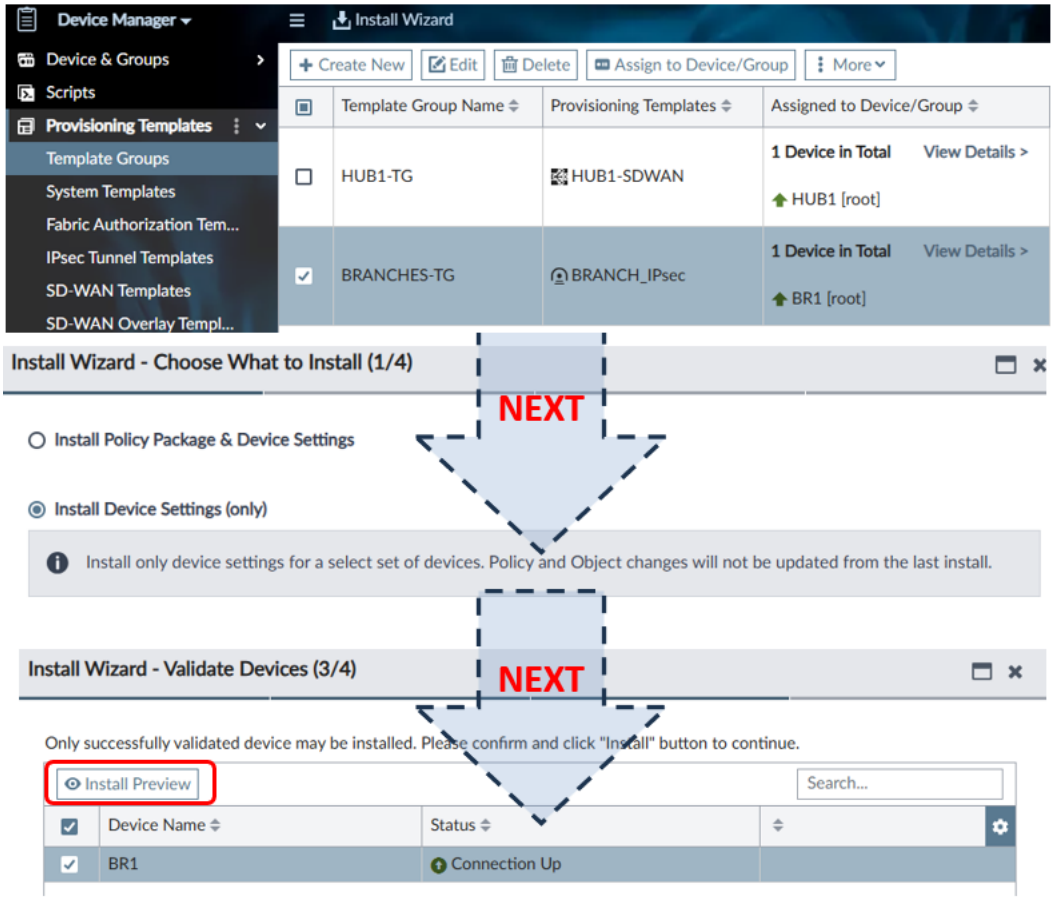
Check Linkedin for any coupon links at: https://www.linkedin.com/in/nicanory-ateya/

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.



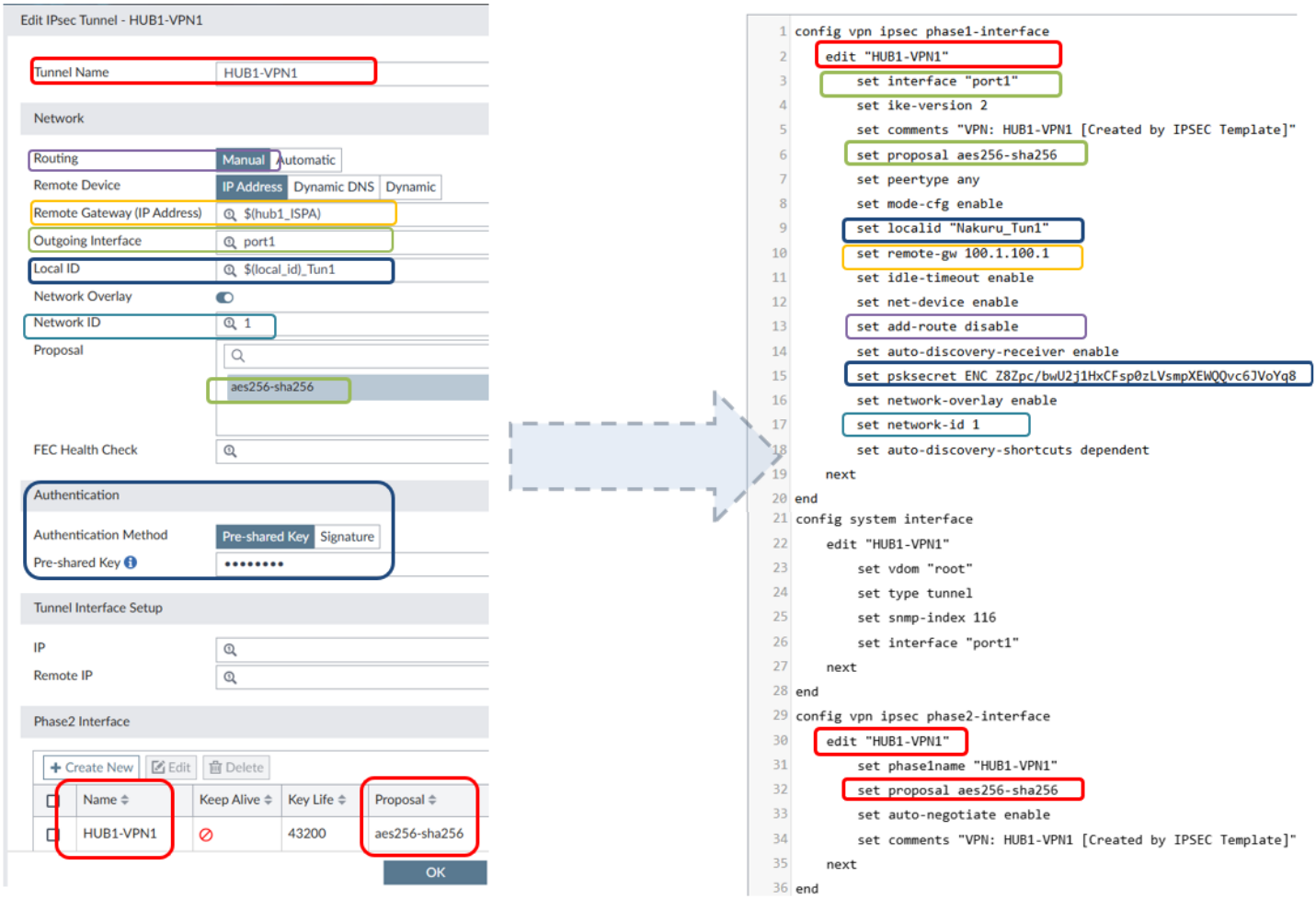
This content is not available in the sample book. The book can be purchased on Leanpub at:

https://leanpub.com/fortinetsd-wandemystified

Check Linkedin for any coupon links at: https://www.linkedin.com/in/nicanory-ateya/

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



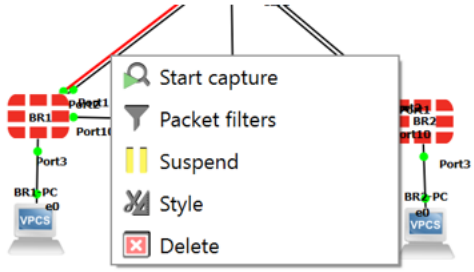
This content is not available in the sample book. The book can be purchased on Leanpub at:

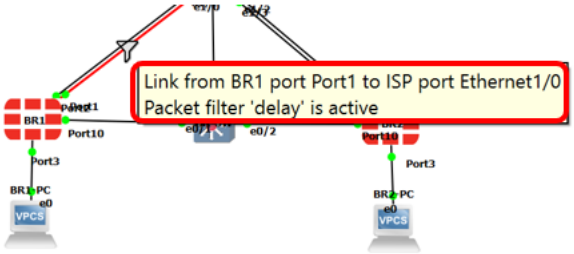
https://leanpub.com/fortinetsd-wandemystified

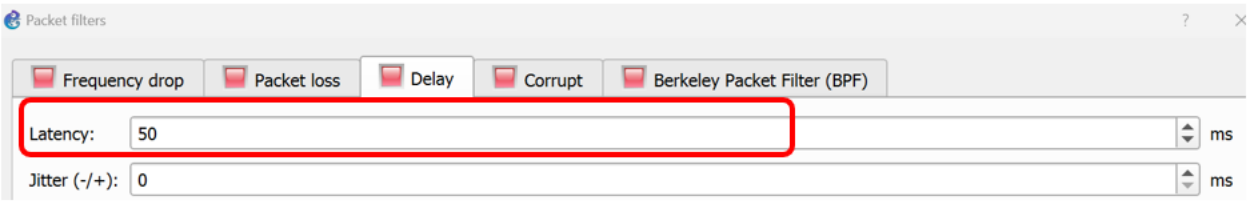
Check Linkedin for any coupon links at: https://www.linkedin.com/in/nicanory-ateya/

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.





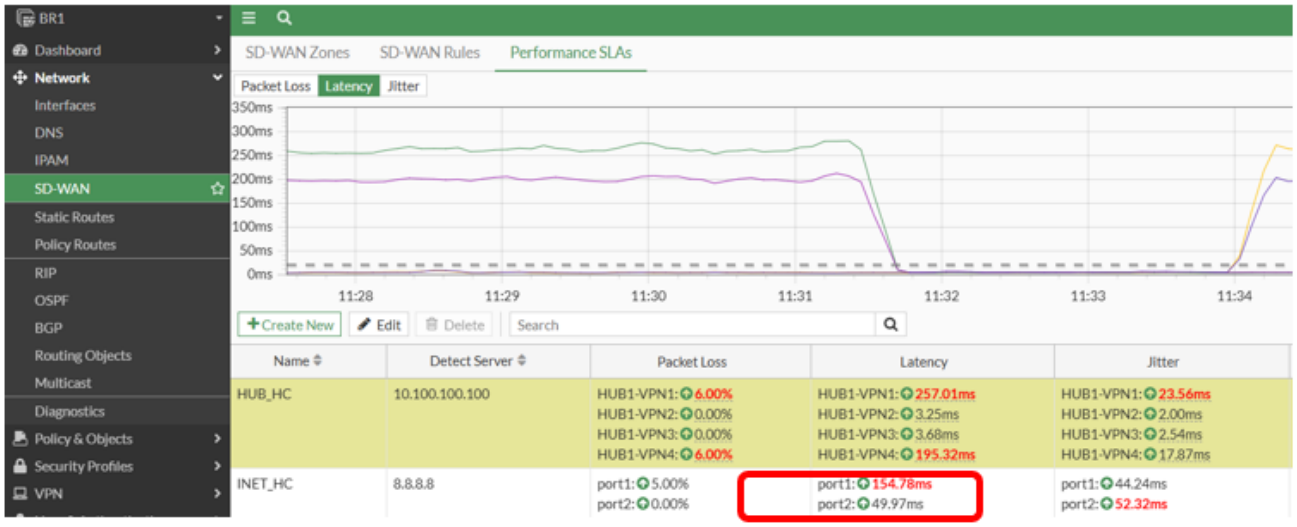


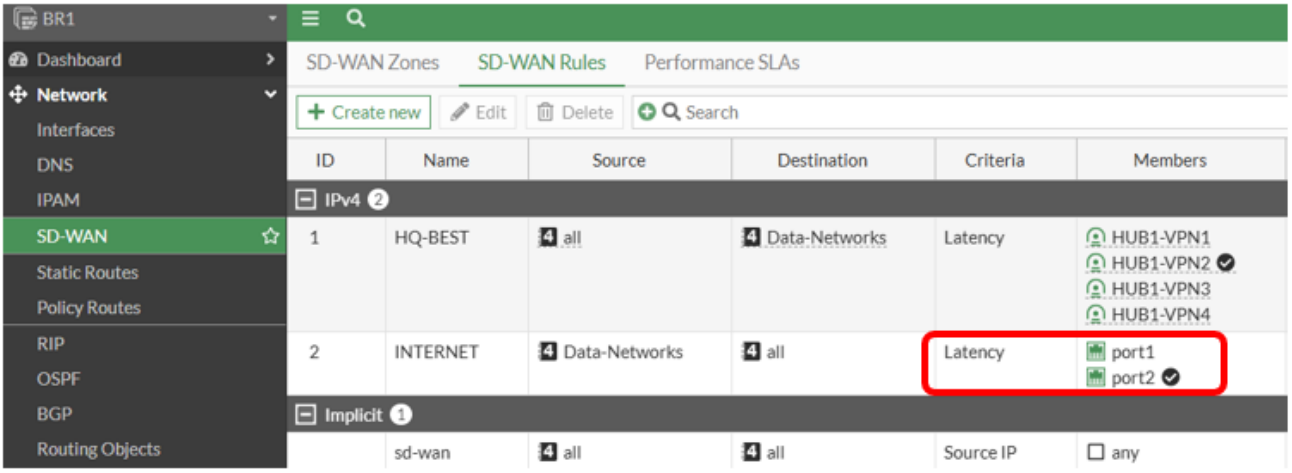
Port2 is selected as the best path since it has less latency compared to port1.

This content is not available in the sample book. The book can be purchased on Leanpub at:

https://leanpub.com/fortinetsd-wandemystified

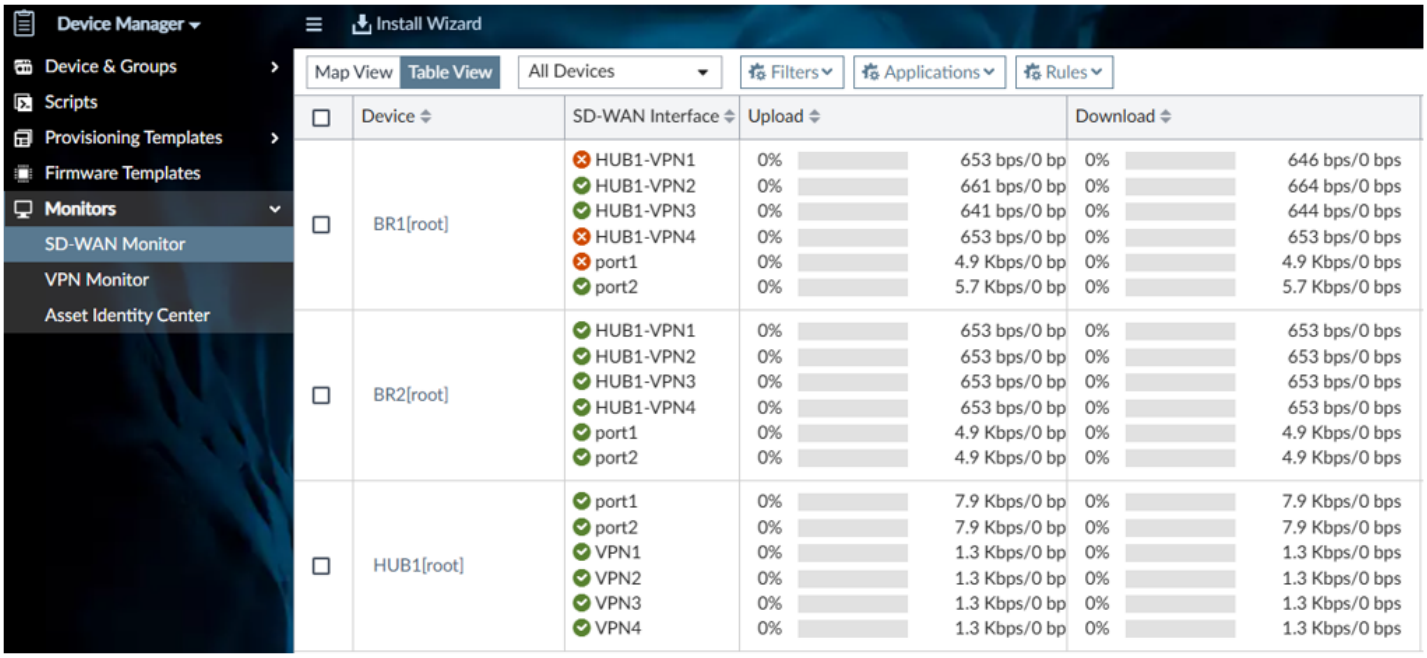
Check Linkedin for any coupon links at: https://www.linkedin.com/in/nicanory-ateya/

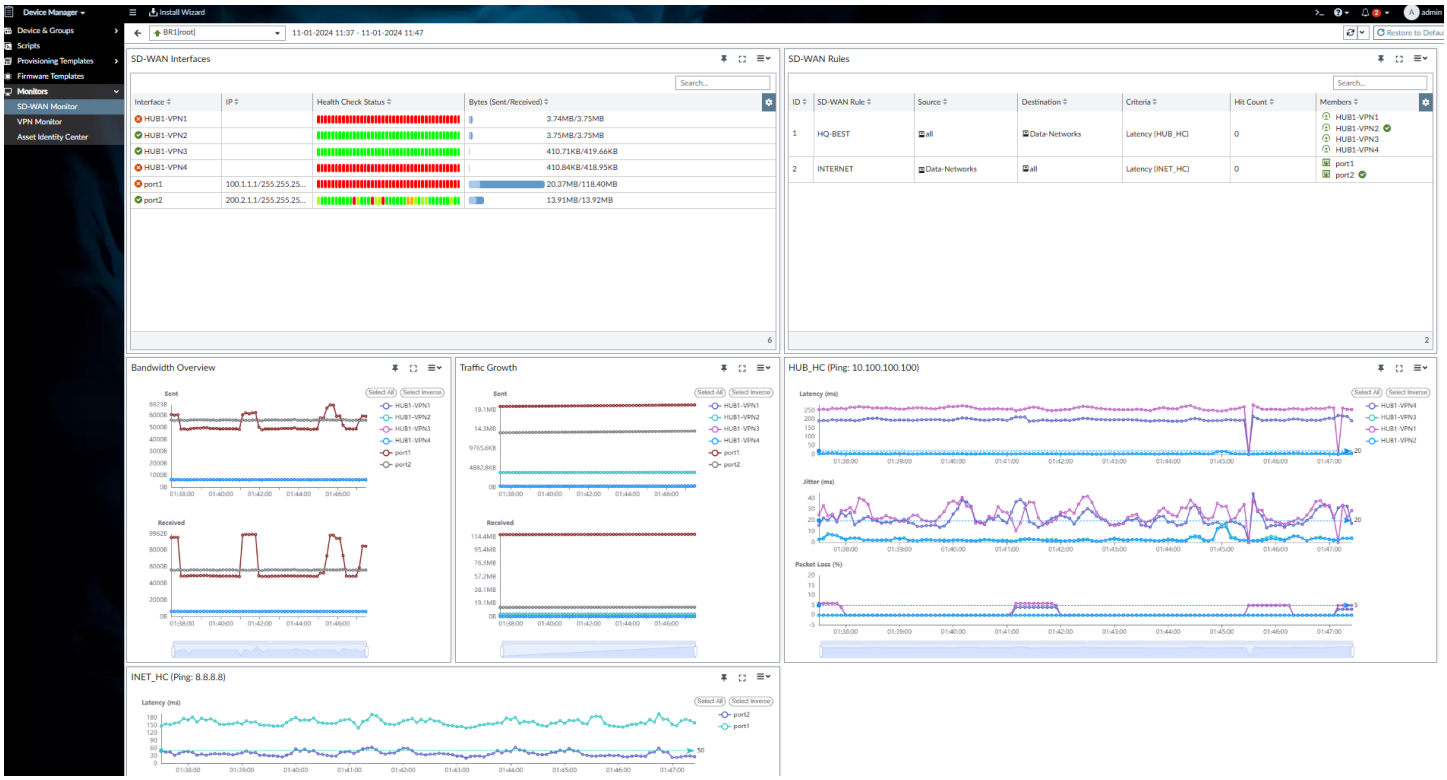

BR1
Dashboard
Network
Interfaces
DNS
IPAM
SD-WAN
Static Routes
Policy Routes
RIP
OSPF
BGP
Routing Objects
Multicast
Diagnostics
Policy & Objects
Security Profiles
VPN
SD-WAN Zones
SD-WAN Rules
Performance SLAs
Packet Loss
Latency
Jitter
350ms
300ms
250ms
200ms
150ms
100ms
50ms
0ms
11:28
11:29
11:30
11:31
11:32
11:33
11:34
Create New
Edit
Delete
Search
Name
Detect Server
Packet Loss
Latency
Jitter
HUB_HC
10.100.100.100
HUB1-VPN1: 6.00%
HUB1-VPN2: 0.00%
HUB1-VPN3: 0.00%
HUB1-VPN4: 6.00%
HUB1-VPN1: 257.01ms
HUB1-VPN2: 3.25ms
HUB1-VPN3: 3.68ms
HUB1-VPN4: 195.32ms
HUB1-VPN1: 23.56ms
HUB1-VPN2: 2.00ms
HUB1-VPN3: 2.54ms
HUB1-VPN4: 17.87ms
INET_HC
8.8.8.8
port1: 5.00%
port2: 0.00%
port1: 154.78ms
port2: 49.97ms
port1: 44.24ms
port2: 52.32ms


BR1
Dashboard
Network
Interfaces
DNS
IPAM
SD-WAN
Static Routes
Policy Routes
RIP
OSPF
BGP
Routing Objects
SD-WAN Zones
SD-WAN Rules
Performance SLAs
Create new
Edit
Delete
Search
ID
Name
Source
Destination
Criteria
Members
IPv4 2
1
HQ-BEST
all
Data-Networks
Latency
HUB1-VPN1
HUB1-VPN2
HUB1-VPN3
HUB1-VPN4
2
INTERNET
Data-Networks
all
Latency
port1
port2
Implicit 1
sd-wan
all
all
Source IP
any

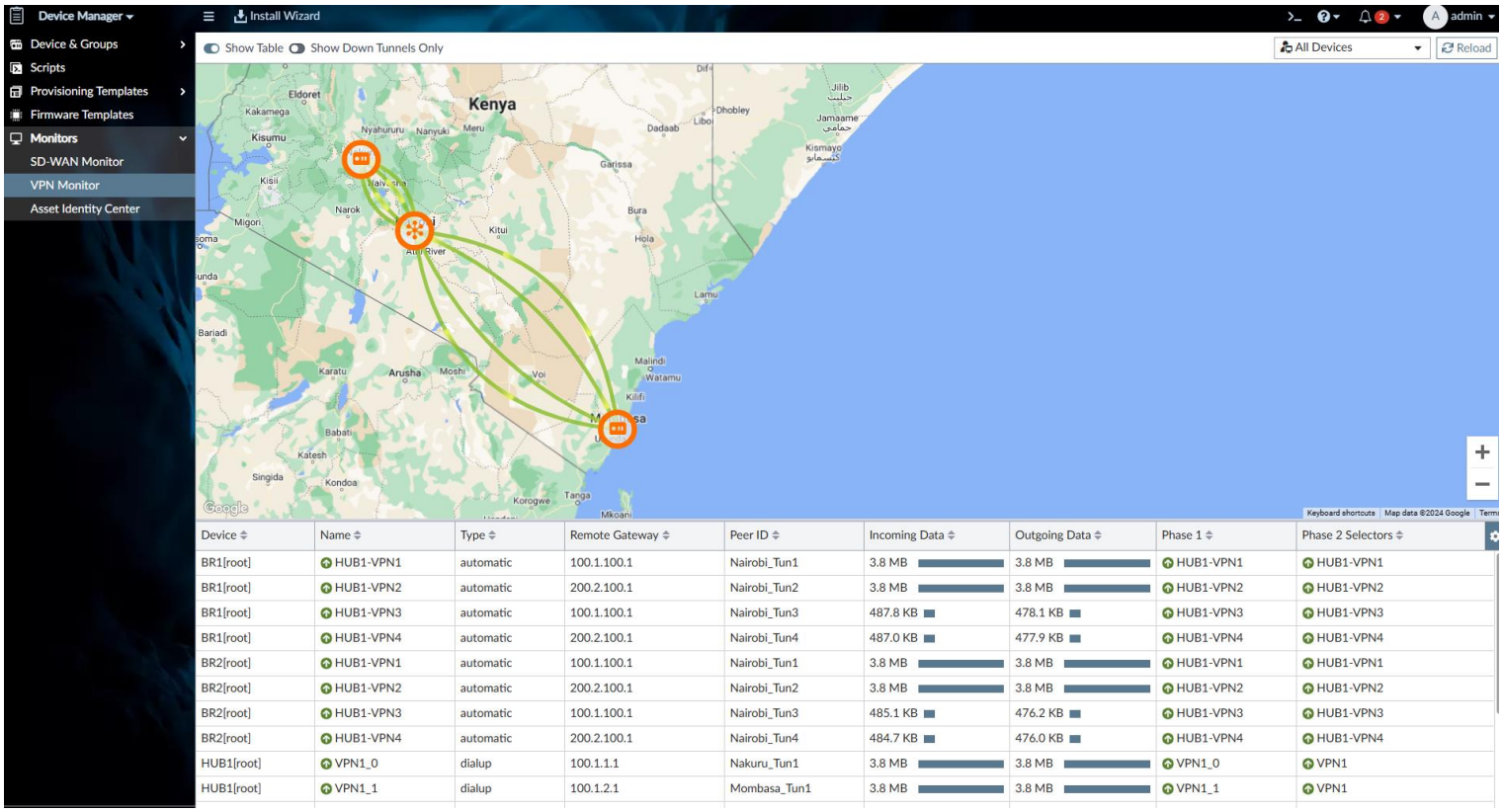
SDWAN MONITORING From FortiManager

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.

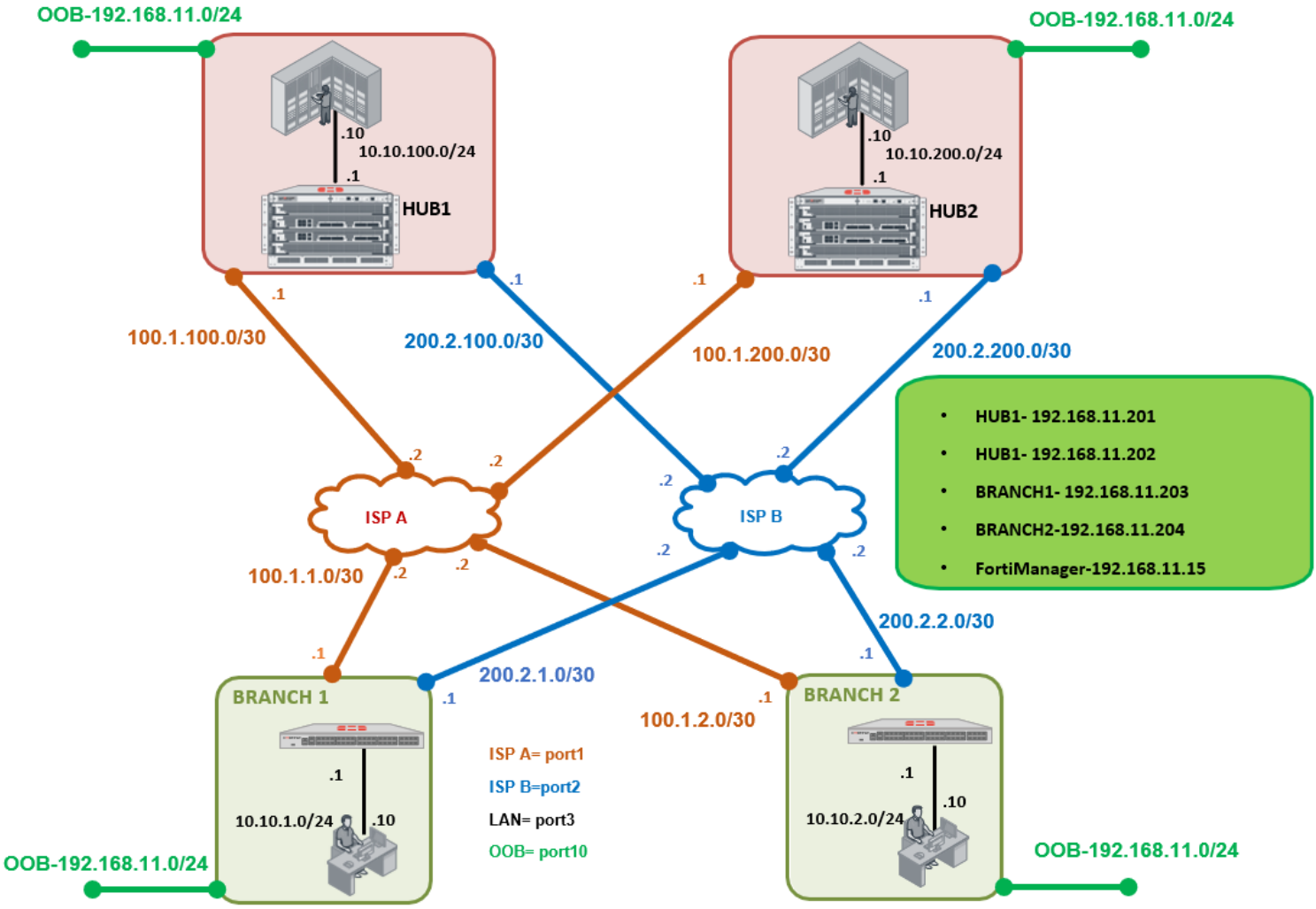


Device	Name	Type	Remote Gateway	Peer ID	Incoming Data	Outgoing Data	Phase 1	Phase 2 Selectors
BR1[root]	HUB1-VPN1	automatic	100.1.100.1	Nairobi_Tun1	3.8 MB	3.8 MB	HUB1-VPN1	HUB1-VPN1
BR1[root]	HUB1-VPN2	automatic	200.2.100.1	Nairobi_Tun2	3.8 MB	3.8 MB	HUB1-VPN2	HUB1-VPN2
BR1[root]	HUB1-VPN3	automatic	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

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:



This content is not available in the sample book. The book can be purchased on Leanpub at:

https://leanpub.com/fortinetsd-wandemystified

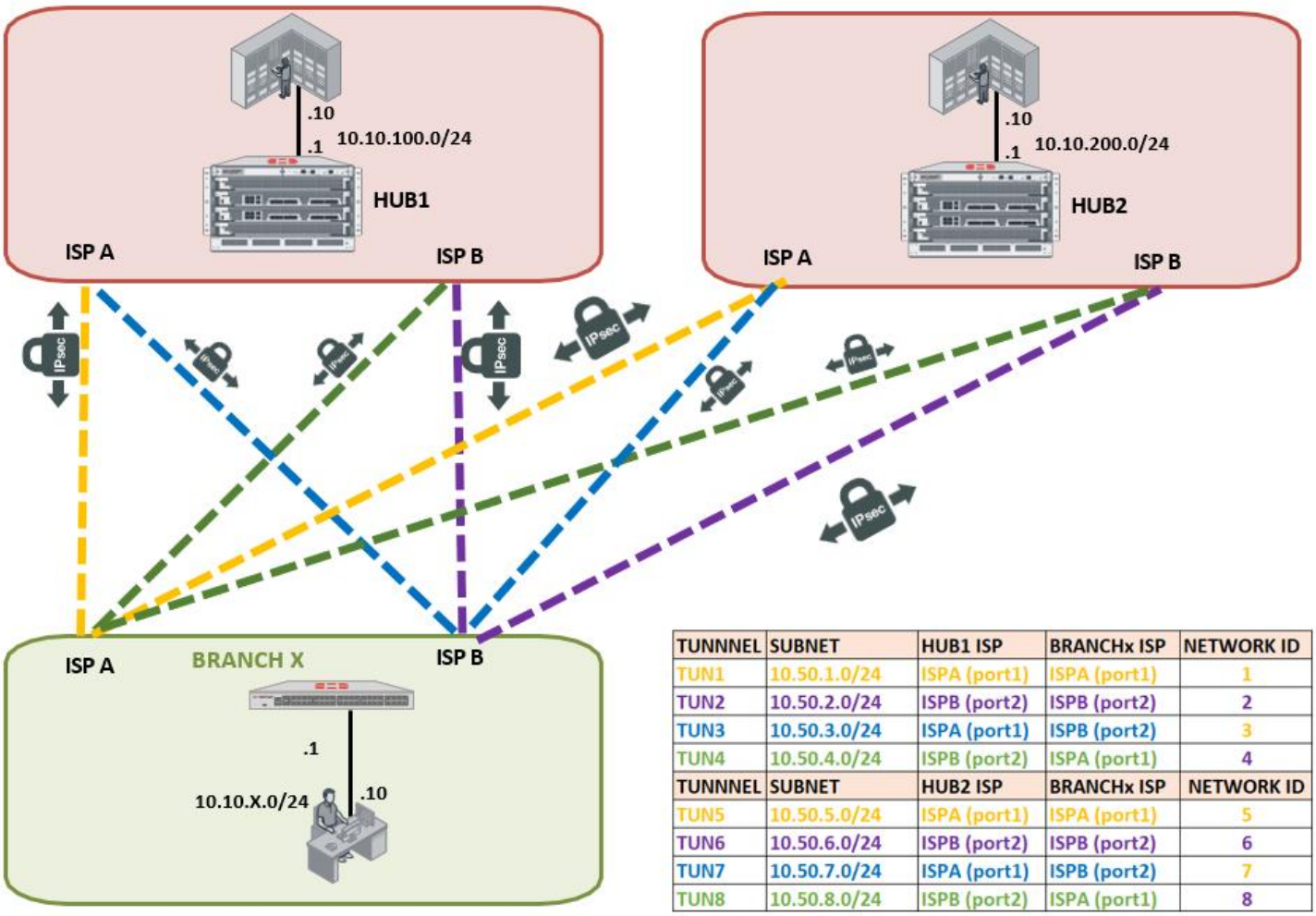
Check Linkedin for any coupon links at: https://www.linkedin.com/in/nicanory-ateya/

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 will be 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

This content is not available in the sample book. The book can be purchased on Leanpub at:

https://leanpub.com/fortinetsd-wandemystified

Check Linkedin for any coupon links at: https://www.linkedin.com/in/nicanory-ateya/

HUB2 Firewall Policies

Five key policies are required at the HUB.

This content is not available in the sample book. The book can be purchased on Leanpub at:

https://leanpub.com/fortinetsd-wandemystified

Check Linkedin for any coupon links at: https://www.linkedin.com/in/nicanory-ateya/

HUB2 LOOPBACK Health Check Verification

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

This content is not available in the sample book. The book can be purchased on Leanpub at:

https://leanpub.com/fortinetsd-wandemystified

Check Linkedin for any coupon links at: https://www.linkedin.com/in/nicanory-ateya/

BRANCHES BGP Configuration for HUB2

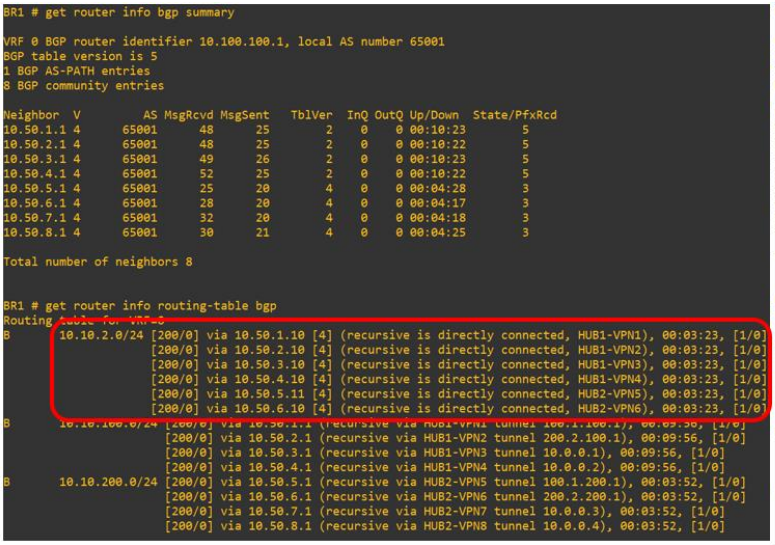
This content is not available in the sample book. The book can be purchased on Leanpub at:

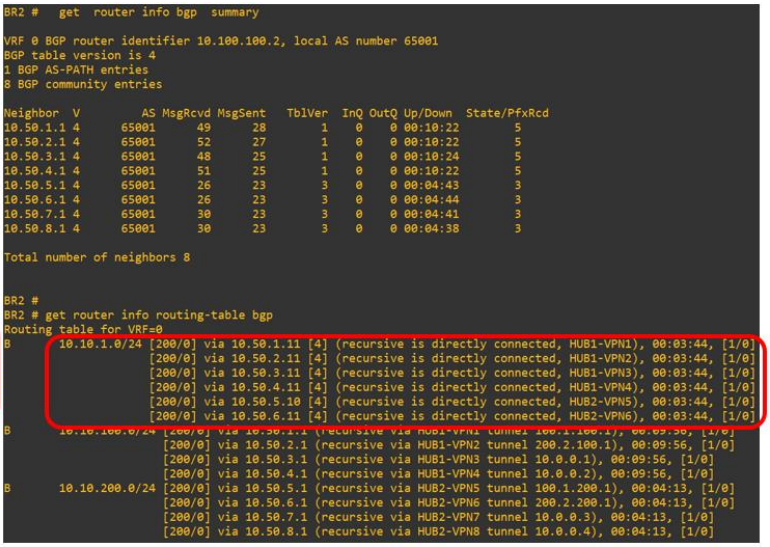
https://leanpub.com/fortinetsd-wandemystified

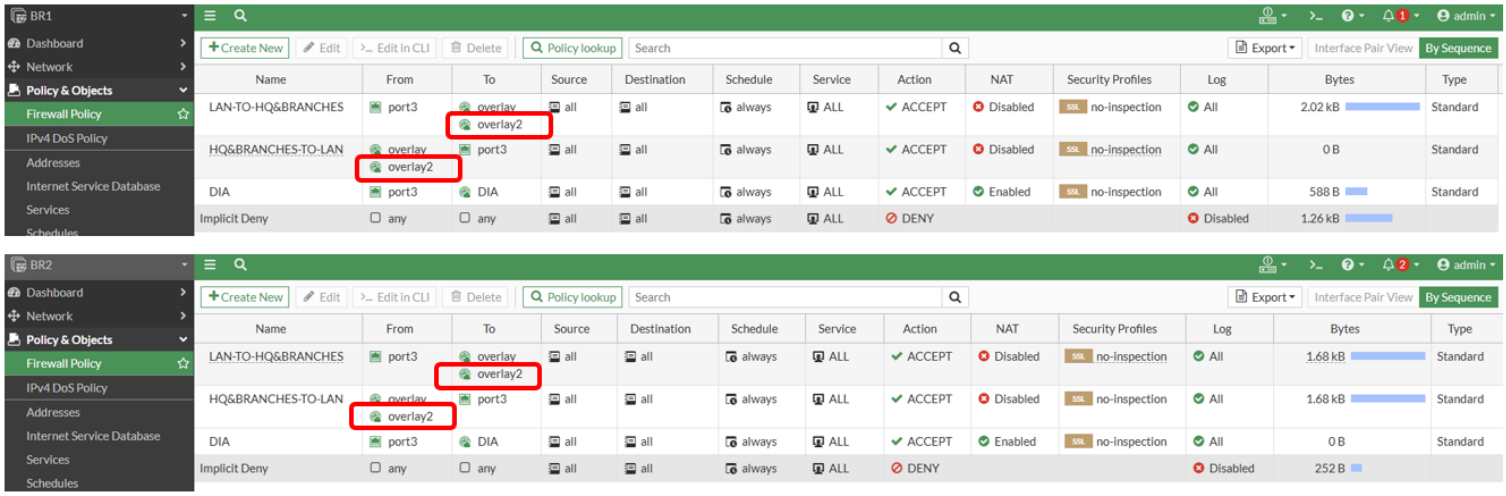
Check Linkedin for any coupon links at: https://www.linkedin.com/in/nicanory-ateya/

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

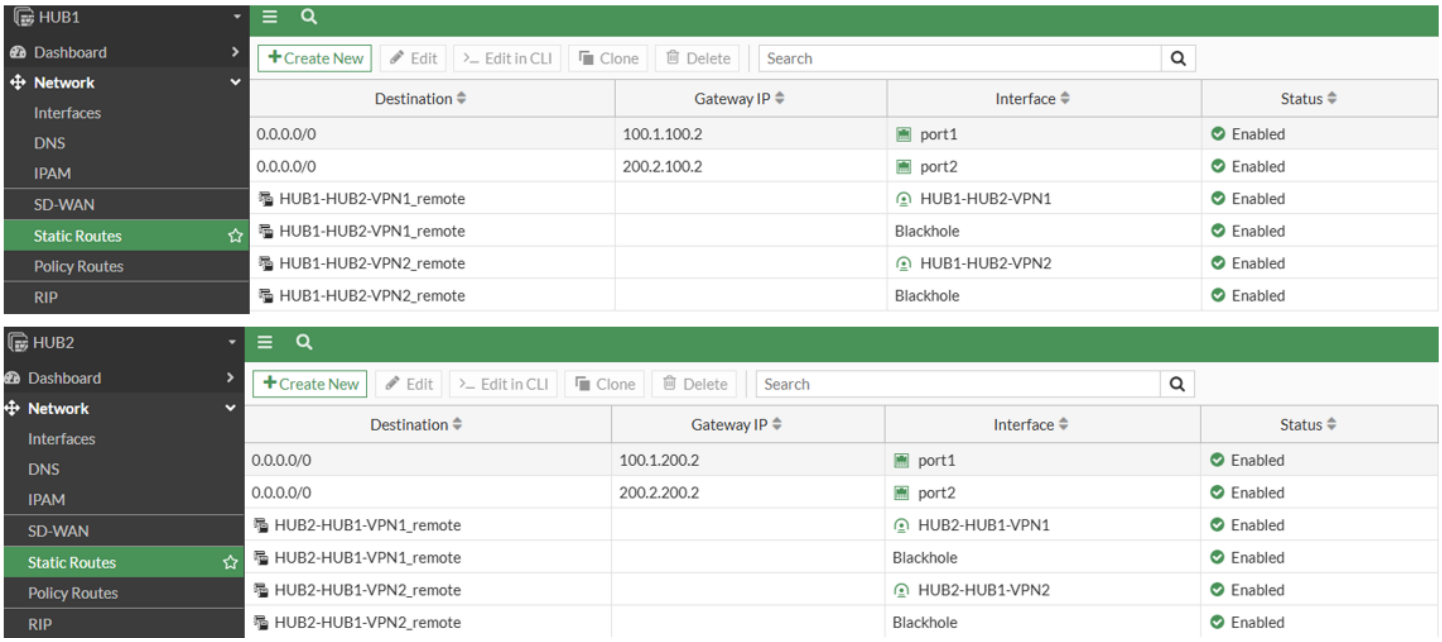
After establishing eight neighbors, the branches still have 6 paths learnt from each other. This limitation is on the number of routes HUB2 can advertise downstream to branches which is limited to 2.







ADVPN with 8 Tunnels


HUB1
Dashboard
Network
Interfaces
DNS
IPAM
SD-WAN
Static Routes
Policy Routes
RIP
Create New
Edit
Edit in CLI
Clone
Delete
Search
Destination
Gateway IP
Interface
Status
0.0.0.0/0
100.1.100.2
port1
Enabled
0.0.0.0/0
200.2.100.2
port2
Enabled
HUB1-HUB2-VPN1_remote
HUB1-HUB2-VPN1
Enabled
HUB1-HUB2-VPN1_remote
Blackhole
Enabled
HUB1-HUB2-VPN2_remote
HUB1-HUB2-VPN2
Enabled
HUB1-HUB2-VPN2_remote
Blackhole
Enabled

HUB2
Dashboard
Network
Interfaces
DNS
IPAM
SD-WAN
Static Routes
Policy Routes
RIP
Create New
Edit
Edit in CLI
Clone
Delete
Search
Destination
Gateway IP
Interface
Status
0.0.0.0/0
100.1.200.2
port1
Enabled
0.0.0.0/0
200.2.200.2
port2
Enabled
HUB2-HUB1-VPN1_remote
HUB2-HUB1-VPN1
Enabled
HUB2-HUB1-VPN1_remote
Blackhole
Enabled
HUB2-HUB1-VPN2_remote
HUB2-HUB1-VPN2
Enabled
HUB2-HUB1-VPN2_remote
Blackhole
Enabled

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

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.

This content is not available in the sample book. The book can be purchased on Leanpub at:

https://leanpub.com/fortinetsd-wandemystified

Check Linkedin for any coupon links at: https://www.linkedin.com/in/nicanory-ateya/

SDWAN OVERLAY Templates- SINGLE HUB

This content is not available in the sample book. The book can be purchased on Leanpub at:

https://leanpub.com/fortinetsd-wandemystified

Check Linkedin for any coupon links at: https://www.linkedin.com/in/nicanory-ateya/

SDWAN OVERLAY Templates- DUAL HUB

This content is not available in the sample book. The book can be purchased on Leanpub at:

https://leanpub.com/fortinetsd-wandemystified

Check Linkedin for any coupon links at: https://www.linkedin.com/in/nicanory-ateya/

HUB2 Configuration Preview

This content is not available in the sample book. The book can be purchased on Leanpub at:

https://leanpub.com/fortinetsd-wandemystified

Check Linkedin for any coupon links at: https://www.linkedin.com/in/nicanory-ateya/

Branches IPsec Verification

This content is not available in the sample book. The book can be purchased on Leanpub at:

https://leanpub.com/fortinetsd-wandemystified

Check Linkedin for any coupon links at: https://www.linkedin.com/in/nicanory-ateya/

HUBs IPsec Verification

This content is not available in the sample book. The book can be purchased on Leanpub at:

https://leanpub.com/fortinetsd-wandemystified

Check Linkedin for any coupon links at: https://www.linkedin.com/in/nicanory-ateya/

Chapter 5 Questions

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

Answers to End of Chapter Questions

This content is not available in the sample book. The book can be purchased on Leanpub at:

https://leanpub.com/fortinetsd-wandemystified

Check Linkedin for any coupon links at: https://www.linkedin.com/in/nicanory-ateya/