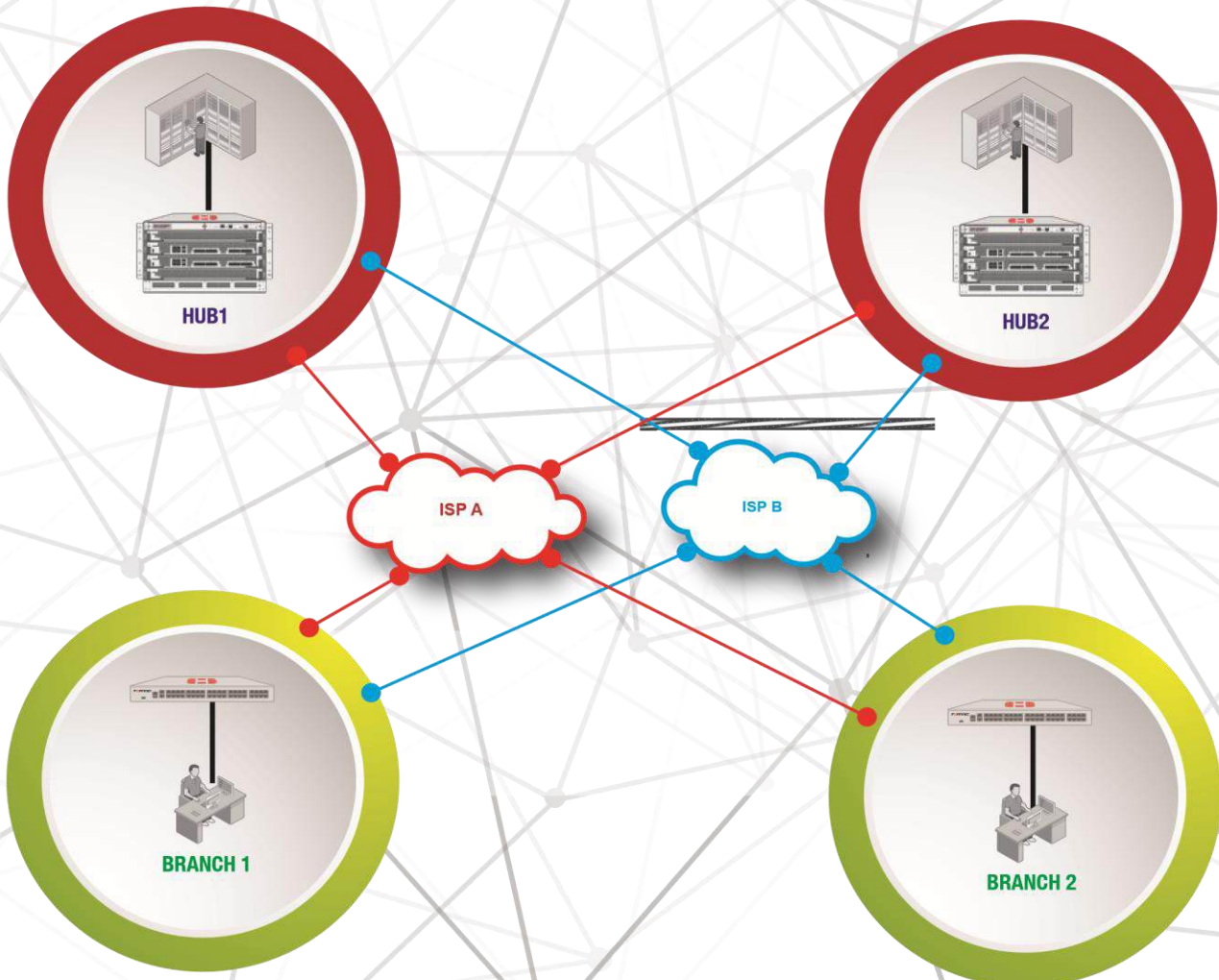


FORTINET SD-WAN DEMYSTIFIED

Ultimate Technical Deployment Guide



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.

© 2024 Nicanory Ateya

All rights reserved. No part of this book may be reproduced in any form or by any electronic or mechanical means, including information storage and retrieval systems, without written permission from the author, except for the use of brief quotations in a book review.

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	1
SDWAN Building Blocks.....	1
LAB Environment Overview.....	3
Tunnel naming convention.....	4
Chapter 2: Deployment using GUI & CLI	5
IPsec Configuration.....	7
Mismatch in PSK.....	9
Mismatch in Network ID.....	9
Requirement for firewall policies at the HUB and Branches.....	10
Branches IPsec Monitor.....	10
HUB IPsec Monitor.....	11
BRANCH IPsec Command Details.....	12
HUB IPsec Command Details.....	12
BGP Configuration.....	13
BRANCH IN-SLA-STATE.....	13
BRANCH OUT-OF-SLA-STATE.....	13
BRANCH1 BGP Configuration.....	14
BRANCH2 BGP Configuration.....	15
HUB1 BGP Configuration.....	16
BGP HUB Verification.....	17
BGP BRANCH VERIFICATION.....	20
ADVPN.....	20
ADVPN Verification.....	21
Chapter 2 questions.....	23
Chapter 3: FortiManager Single Hub with Dual ISP Underlays	24
Provision the HUB for FortiManager Access.....	25
Metadata Variables.....	29
SD-WAN FortiManager role mapping.....	30
IPsec Templates.....	30
Branch Template Group.....	32
Branch1 IPsec HUB1-VPN1 Config Preview.....	33
Branch1 IPsec HUB1-VPN2 Configuration.....	34
Branch1 HUB1-VPN1 & HUB1-VPN2 Comparison.....	35

Branch1 IPSec HUB1-VPN2 Config Preview	36
HUB IPSec Template	37
HUB1 IPSec VPN1 Config Preview	40
HUB1 IPSec VPN2	41
HUB IPSec VPN1 & VPN2 Templates Comparison	42
HUB1 IPSec VPN2 Config Preview	43
Branch SD-WAN Template	44
HUB SDWAN Template	45
Requirement for firewall policies at the HUB and Branches	46
SDWAN FortiGate GUI at HUB1 and Branch1 Verification	47
Normalized Interface.....	48
Address Objects.....	49
Firewall Policies	50
Branch Firewall Policies	50
Hub Loopback Setup.....	51
HUB LOOPBACK Interface normalization.....	51
HUB Firewall Policies	52
HUB LOOPBACK Health Check.....	52
BGP Branch Configuration	53
Branch1 BGP Neighbor1 Configuration Preview	54
Branch1 BGP Route-Map Configuration	55
Branch1 BGP Neighbor1 Configuration Preview	56
BRANCH1 SDWAN-BGP Configuration.....	57
HUB1 VPN1 BGP Configuration	57
HUB1 BGP Route-Map-In Neighbor1.....	58
HUB1 BGP Verification.....	59
Branch1 BGP Verification.....	60
Branch1 BGP Neighbor2 Configuration.....	61
SDWAN Branch1 HUB1-VPN2.....	61
Branch1 BGP Neighbor2 Configuration Preview	62
HUB1 VPN2 BGP Configuration	63
HUB1 BGP Route-Map-In Neighbor2.....	64
HUB1 VPN2 Configuration Preview.	64
HUB1 BGP Routing Verification after two Tunnels are Established.....	65
Branch1 BGP Routing Verification after two Tunnels are Established.....	66
Configuring Additional Branches.....	67

Device Settings Installation to Branch2	68
Routing Review at Branch1 & Branch 2	69
Routing Review at the HUB	70
Tunnel IP address assignment	70
HUB SDWAN Rules.....	71
ADVPN	72
Link Redundancy Single HUB dual ISP	73
HUB1 IPSEC Provisioning for VPN3.....	74
HUB1 IPSEC Provisioning for VPN4.....	75
HUB1 IPSEC Provisioning Preview for VPN3 & VPN4.....	76
Branches IPSEC Provisioning for HUB1-VPN3.....	77
Branches IPSEC Provisioning for HUB1-VPN4.....	78
BRANCH IPSEC Provisioning Preview for HUB1-VPN3 & HUB1-VPN4	79
BRANCH IPSEC Verification for HUB1-VPN3 & HUB1-VPN4	80
HUB1 VPN1 & VPN2 SDWAN overlay Zone Grouping.....	80
BRANCH HUB1-VPN3 & HUB1-VPN4 SDWAN overlay Zone Grouping.....	80
IPSEC HUB1 Status With 4 Tunnels at The Branches	81
HUB1 VPN3 & VPN4 Neighbor Group BGP Configuration.....	82
HUB1 VPN3 & VPN4 Neighbor Ranges BGP Configuration	84
HUB1 BGP Provisioning Preview for VPN3 & VPN4.....	85
BRANCH Neighbor 3 BGP Configuration.....	86
BRANCH Neighbor 4 BGP Configuration.....	86
BRANCH PROVISIONING Preview FOR NEIGHBORS 3 & 4	87
HUB1 BGP Routing verification after 4 Tunnels are Established	88
Healthcheck Configuration at Branch1 for Neighbor 3&4.....	89
Branch 1 & 2 BGP Routing Verification after 4 Tunnels are Established.....	90
Firewall policy consideration with 4 Tunnels	92
HUB SDWAN Rules with 4 tunnels.....	92
BRANCH SDWAN Rules with 4 tunnels.....	93
BRANCH SDWAN Rules with 4 tunnels Verification.....	94
LINK Performance Fluctuation.....	96
SDWAN MONITORING From FortiManager.....	98
Chapter 3 questions	100
Chapter 4: FortiManager Dual Hub with Dual ISP Underlay Deployment	101
IPSEC Redundancy Between a Branch and 2 HUBS.....	102
Register HUB2 To FortiManager	103

IPSEC HUB2 Configuration.....	104
IPSEC HUB2 VPN5-8 CONFIGURATION	104
IPSEC HUB2 Configuration Preview	106
IPSEC BRANCH Configuration for HUB2.....	108
IPSEC BRANCH HUB2-VPN5-8 CONFIGURATION	108
IPSEC BRANCH HUB2-VPN5-8 Configuration Preview	110
SDWAN HUB2 Configuration	111
SDWAN ZONES BRANCH Configuration for HUB2	112
SDWAN HUB2 Verification	113
HUB2 Normalized Interface.....	113
HUB2 LOOPBACK SETUP	114
HUB2 Firewall Policies	114
HUB2 LOOPBACK Health Check.....	115
HUB2 LOOPBACK Health Check Verification.....	115
BRANCHES BGP Configuration for HUB2	116
BRANCHES BGP Configuration Preview for HUB2	116
HUB2 BGP Configuration	117
HUB2 BGP Route-Maps Configuration	118
HUB2 BGP Configuration	118
HUB2 BGP Configuration Preview	119
HUB2 SDWAN Configuration	119
Branch 1 & 2 BGP Routing Verification after 8 Tunnels are Established.....	120
Branch 1 & 2 Firewall Policies for HUB2	122
ADVPN with 8 Tunnels.....	122
HUB1 TO HUB2 Dark Fiber/Private Circuit	123
HUB1 to HUB2 Connection via IPSEC	125
Chapter 4 Questions.....	128
Chapter 5: SD-WAN overlay Orchestration	129
SDWAN OVERLAY Templates- SINGLE HUB.....	130
Branch Configuration Preview	141
HUB Configuration Preview	143
IPsec Verification	144
Hub routing verification	145
Branch routing verification	146
SDWAN OVERLAY Templates- DUAL HUB	147
Branch 2 Configuration Preview	159

HUB2 Configuration Preview	162
Branches IPsec Verification.....	163
HUBs IPsec Verification.....	164
Branches Routing Verification	164
HUBs routing Verification	165
Chapter 5 Questions.....	166
Answers to End of Chapter Questions	167

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:

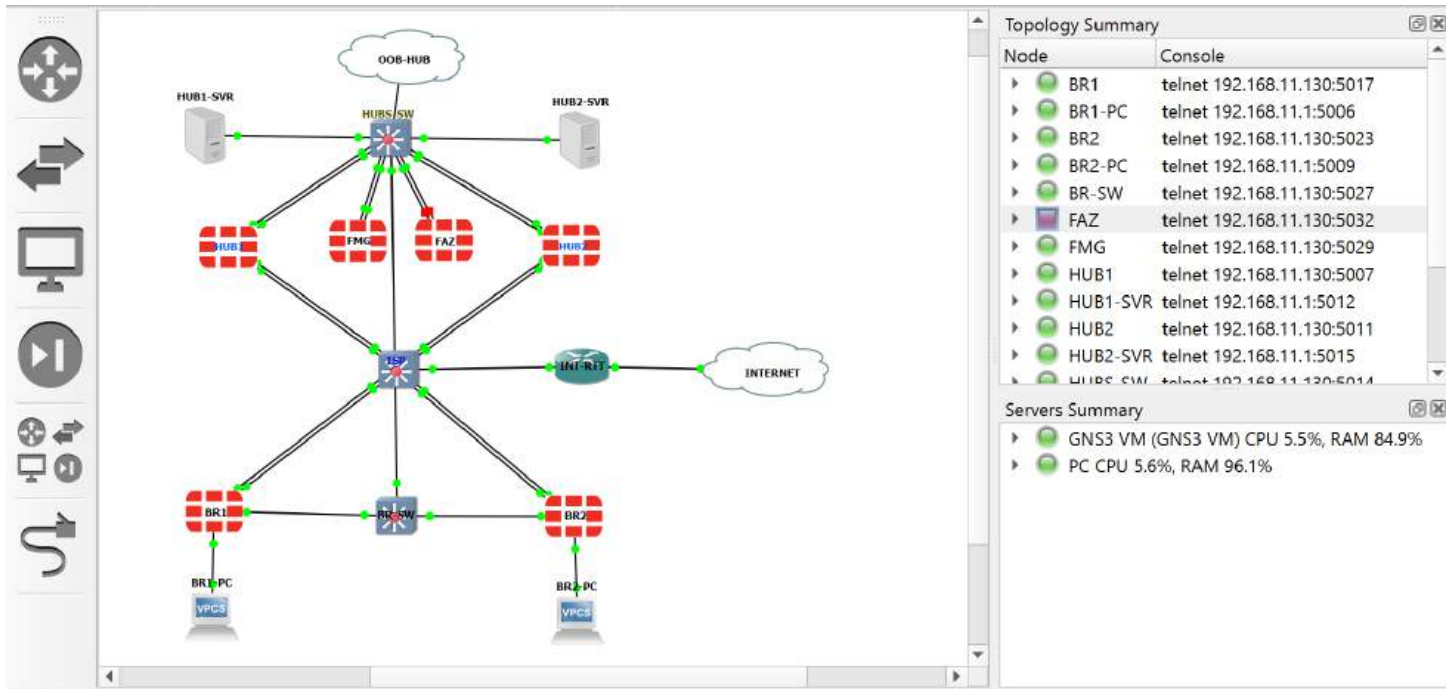
- i. Whether the selected interface is meeting the defined SLA
- ii. The lowest cost between the selected links.

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

TUNNEL	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

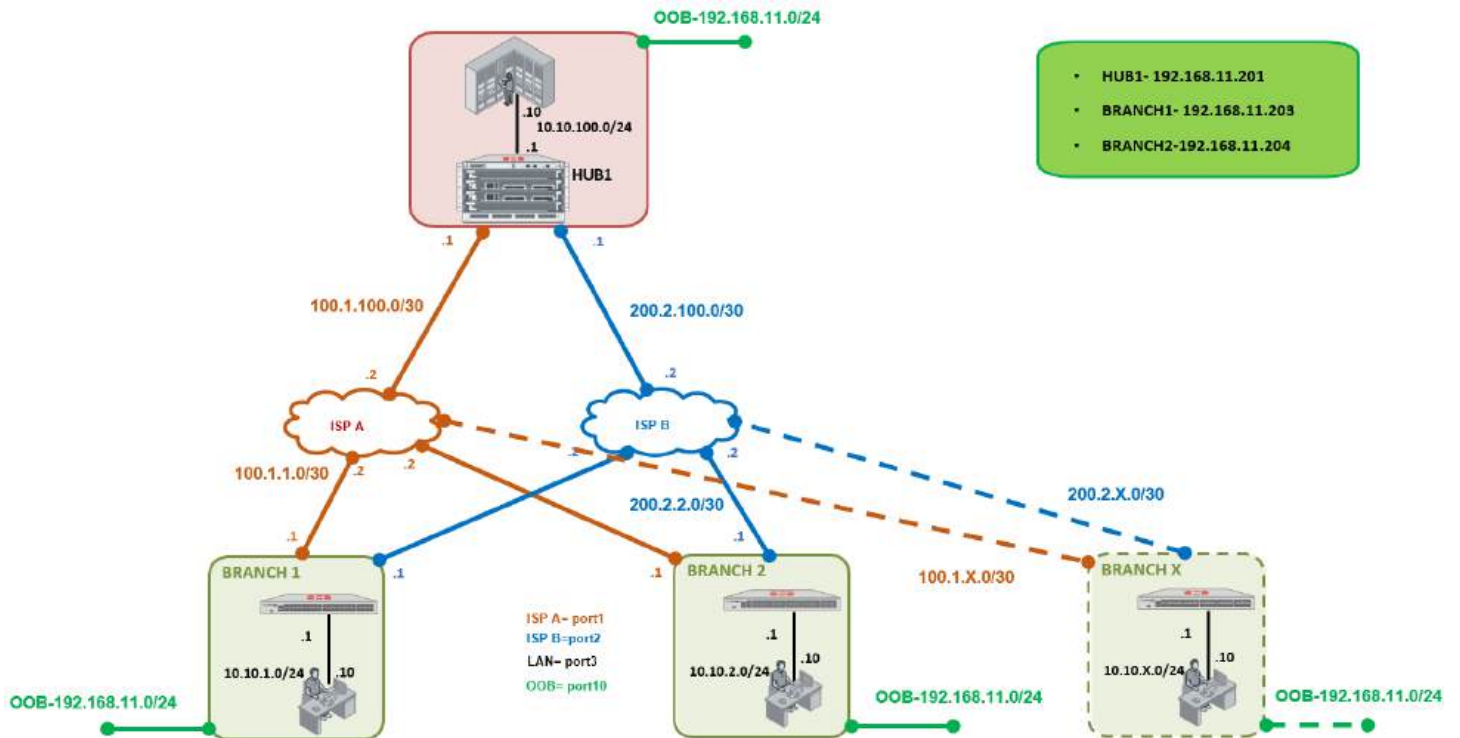
HUB VPN tunnels are named VPNx where X is the tunnel number e.g. VPN1, VPN2 etc.

Branches

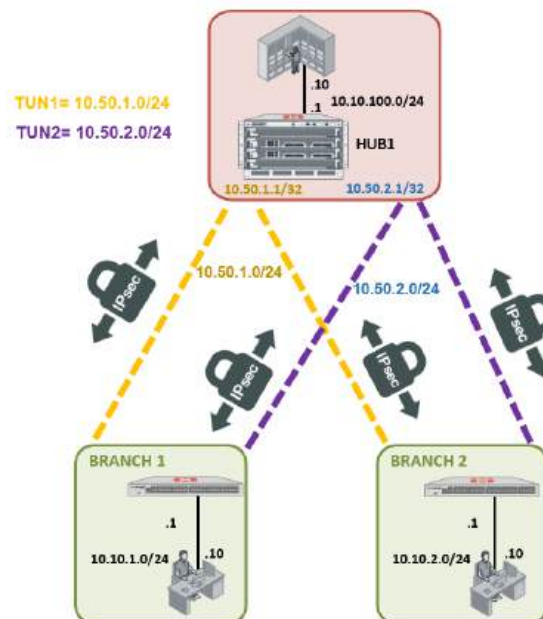
Branch tunnels are named HUBy-VPNx where y is the HUB site identifier and x the tunnel number e.g. The first tunnel to HUB1 would be HUB1-VPN1 while the first to HUB2 will be HUB2-VPN5. This will be elaborated in the LABs

Chapter 2: Deployment using GUI & CLI

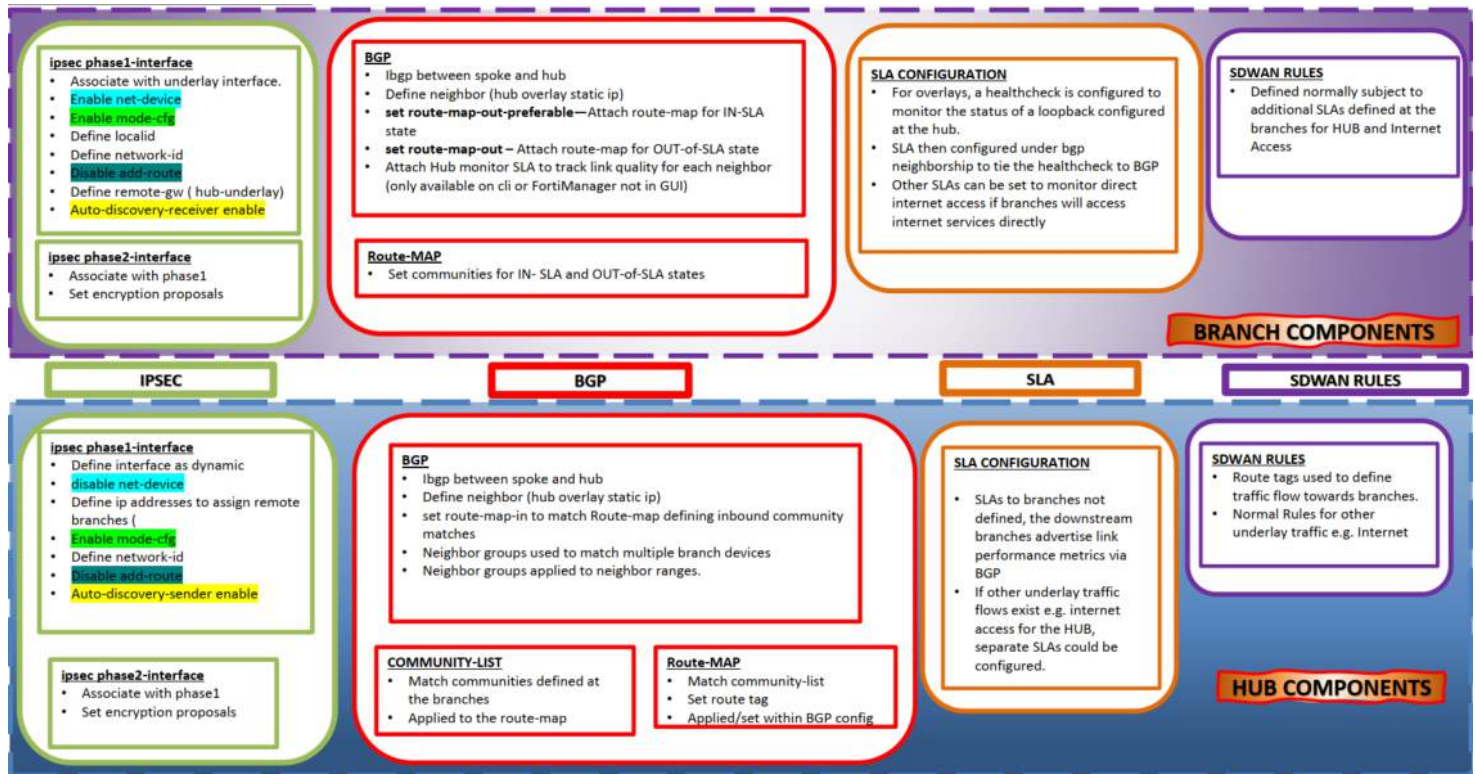
The diagram below represents the reference topology for the GUI & CLI setup made up of two branches and a single hub. Each site has two undelay/ISP links. The underlay path has a single overlay network from the branches to the HUB.



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.

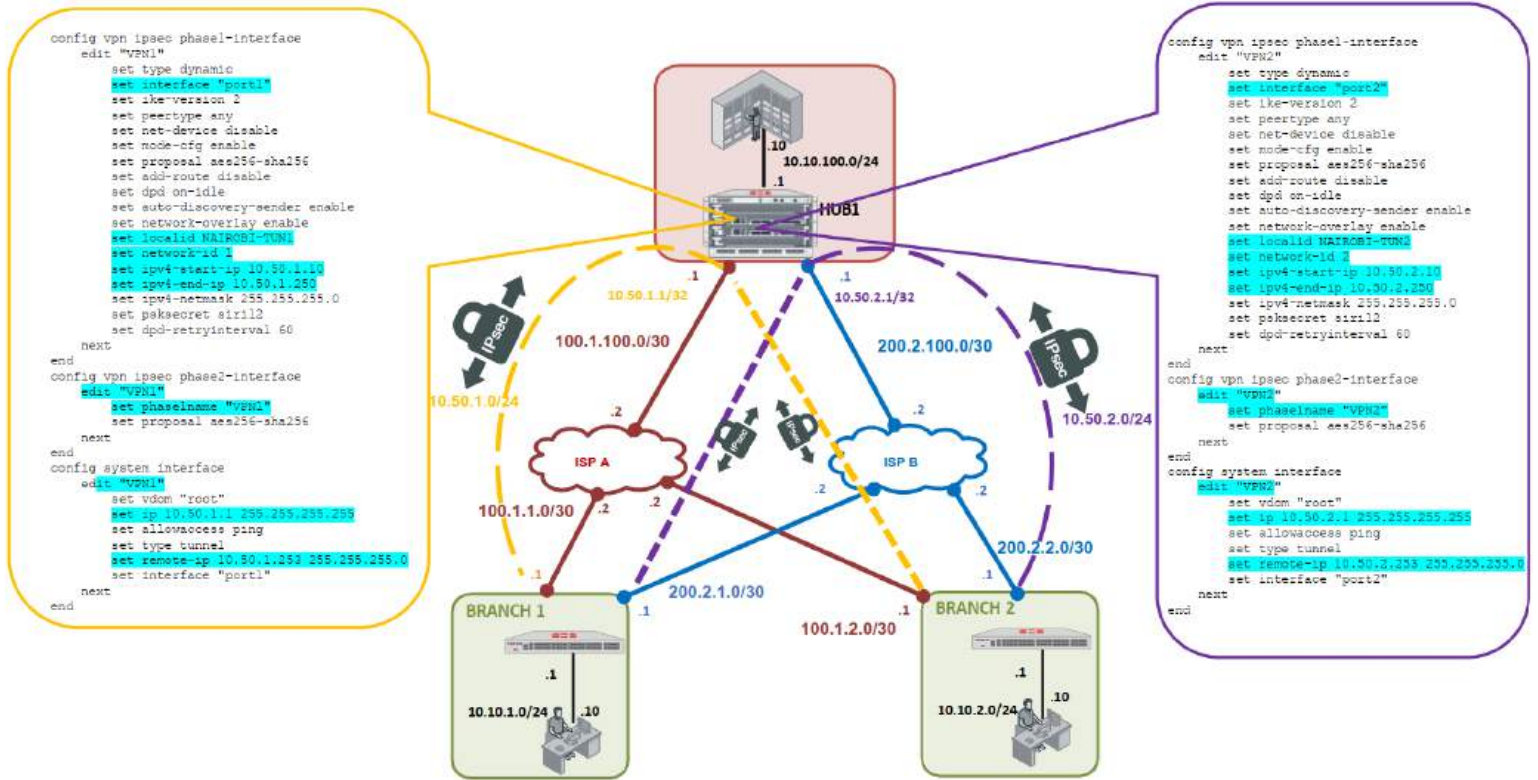


Fortinet's SD-WAN has four main components that are configured at the HUB and respective branches. These include **IPsec**, **BGP**, **SLA** and **SD-WAN rules**. Each has sub-components as highlighted in the diagram below. This will be referenced to configure HUB1 to connect to the two branches via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



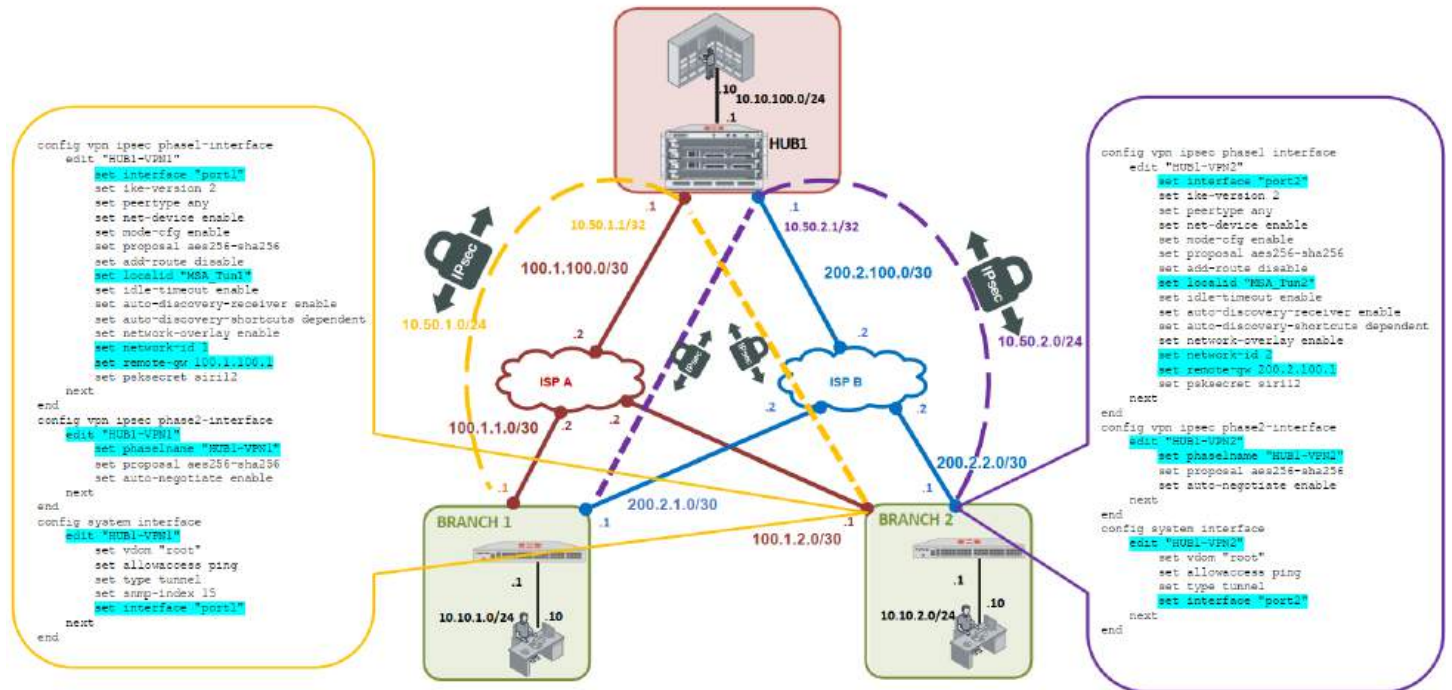
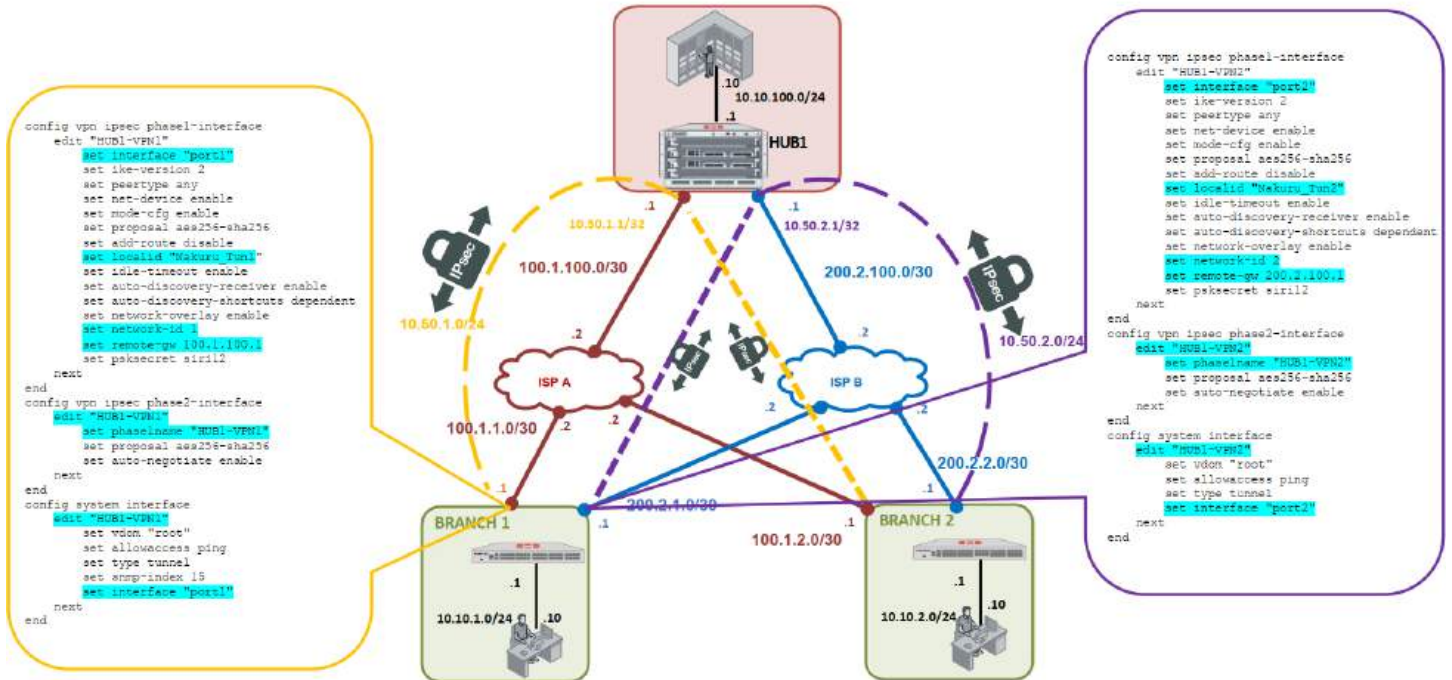
IPsec Configuration

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 HUB's 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
```

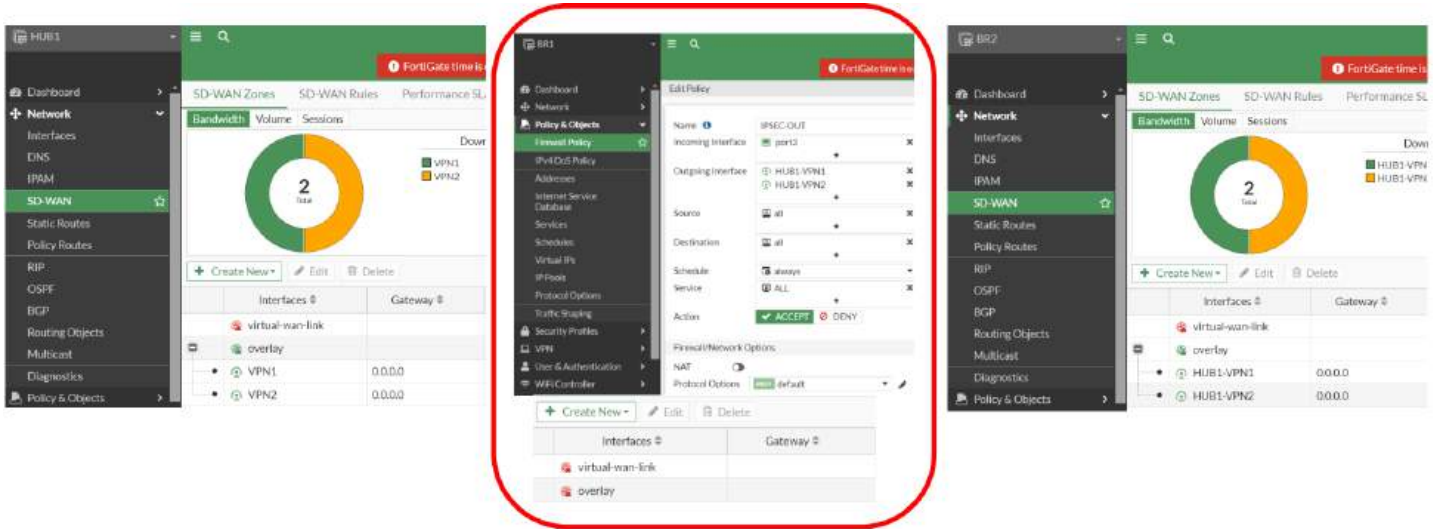
Enable the debugs above to check the VPN state. The output below is given when there is a network-ID mismatch between the HUB and branch i.e. **no proposal chosen** with the debugs done at the HUB.

```
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:

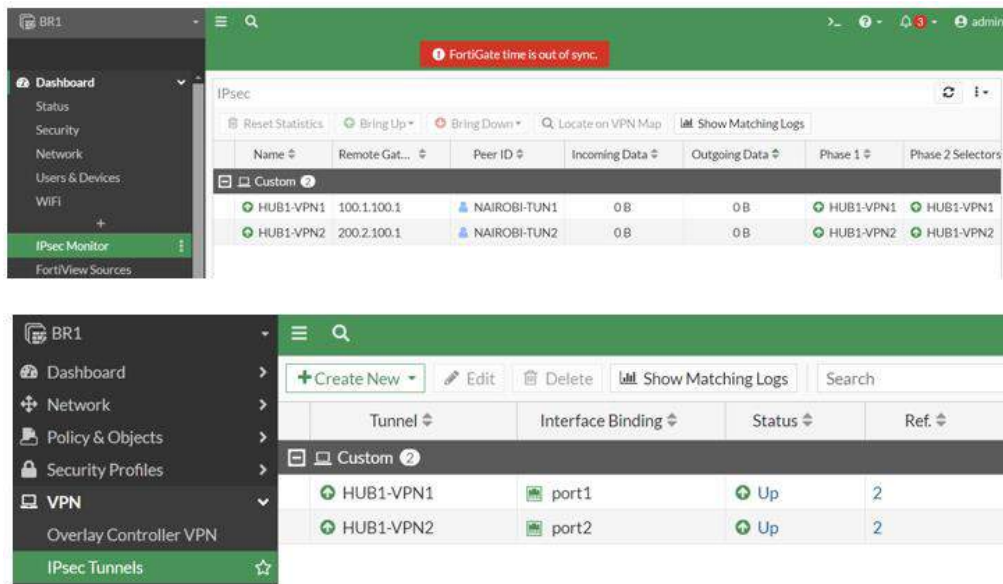
- Configure an outbound policy at the branch referencing the VPN tunnels as the destination and an inbound policy at the HUB with the VPN interfaces as the source interface.
- Bundle the VPN interfaces at the branches and HUB in respective SDWAN zones.
- A combination of the two, on side with the policy,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

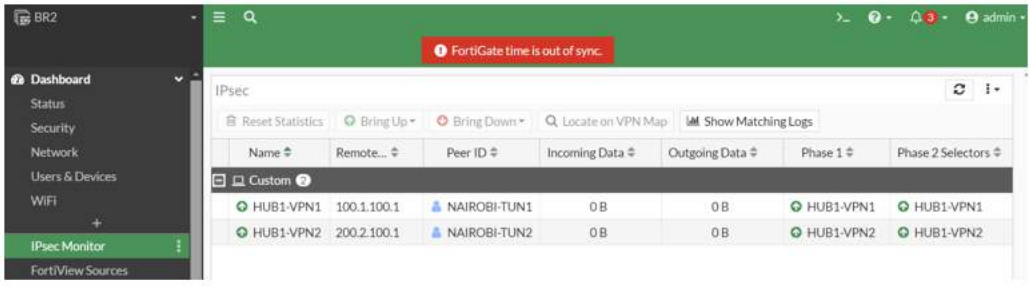


The setup above will not require policies at HUB1 and BR2 since they have SDWAN zones configured while BR1 will require outbound firewall policies to bring up the tunnels because it's HUB1-VPN1 and HUB1-VPN2 interfaces are not included in any SDWAN zone.

Branches IPsec Monitor

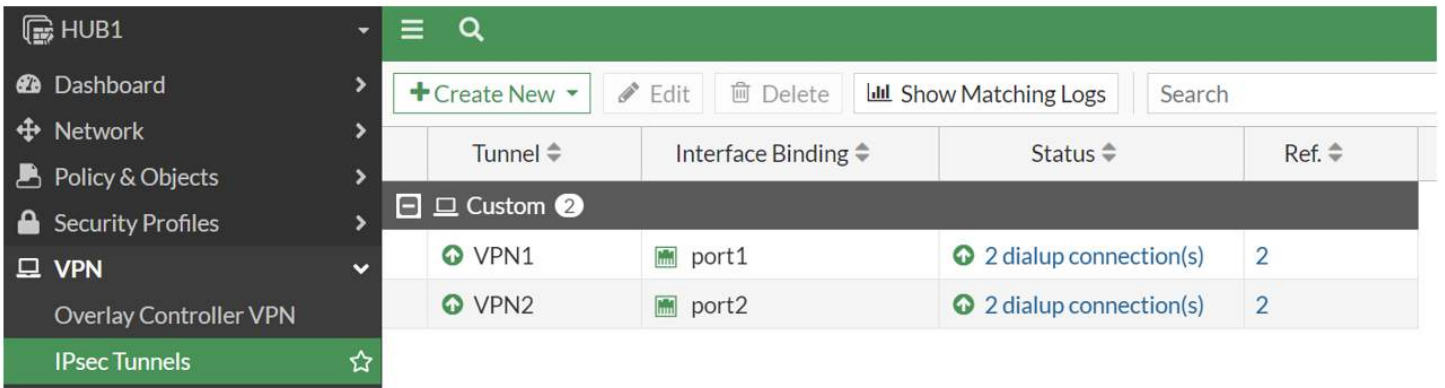
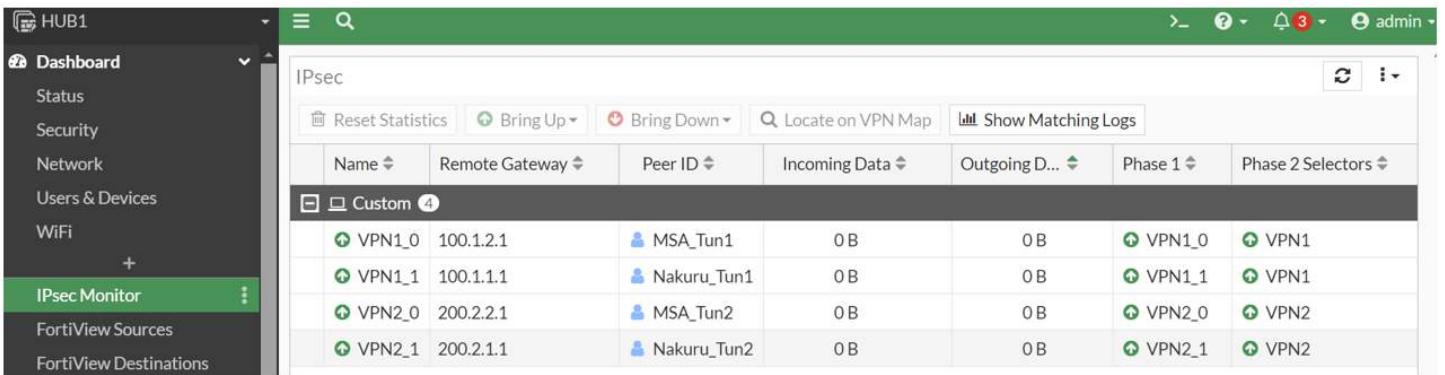
After configuring required firewall policies or assignment to SD-WAN zones at the HUB and branches, the tunnels will be active. This is seen under IPsec Monitor Dashboard or the configuration section for the VPNs at the branches. The peer ID shown at the branches is the local ID configured at the HUB.





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 siril2
set dpd-retrycount 2
set dpd-retryinterval 2
next
end

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

config system interface
edit "HUB1-VPN1"--Created Automatically after Isec 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 siril2
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]
```

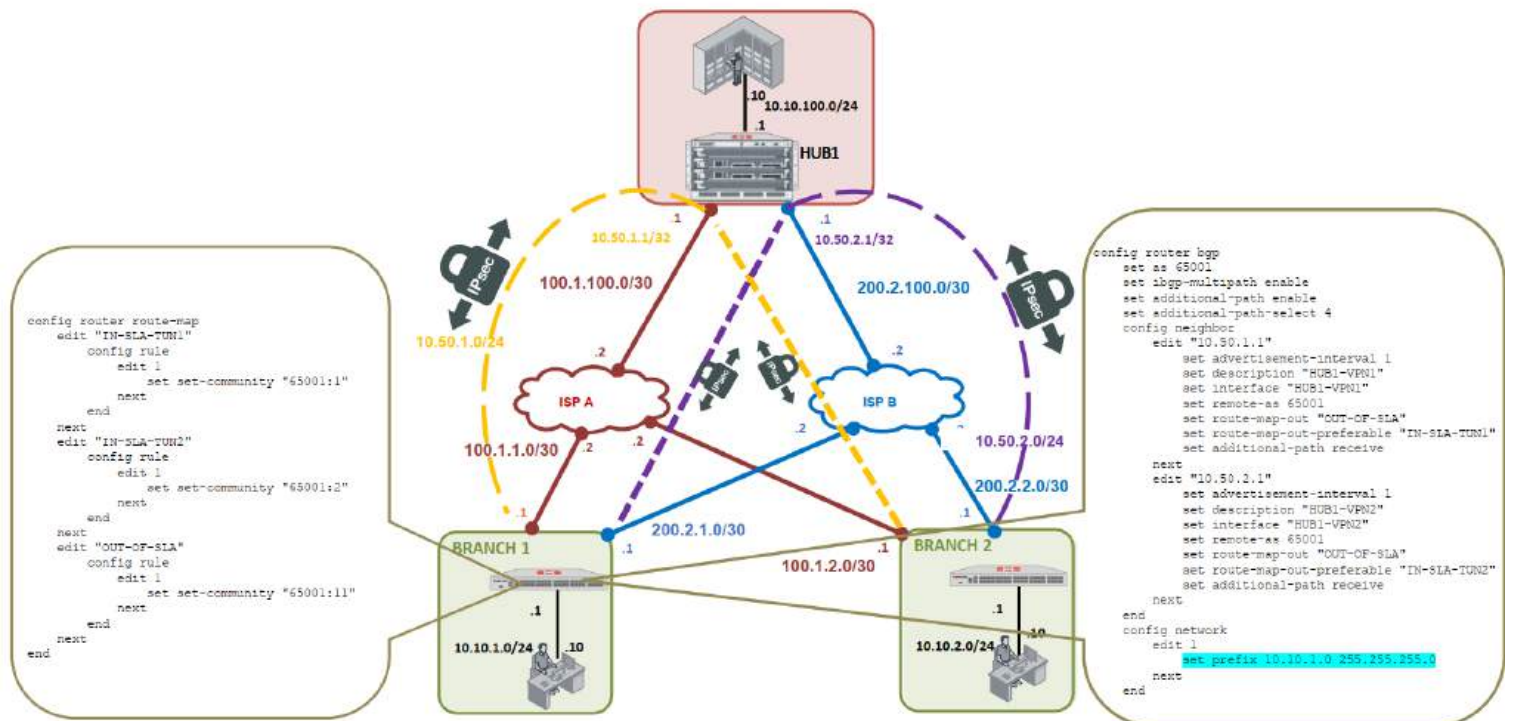
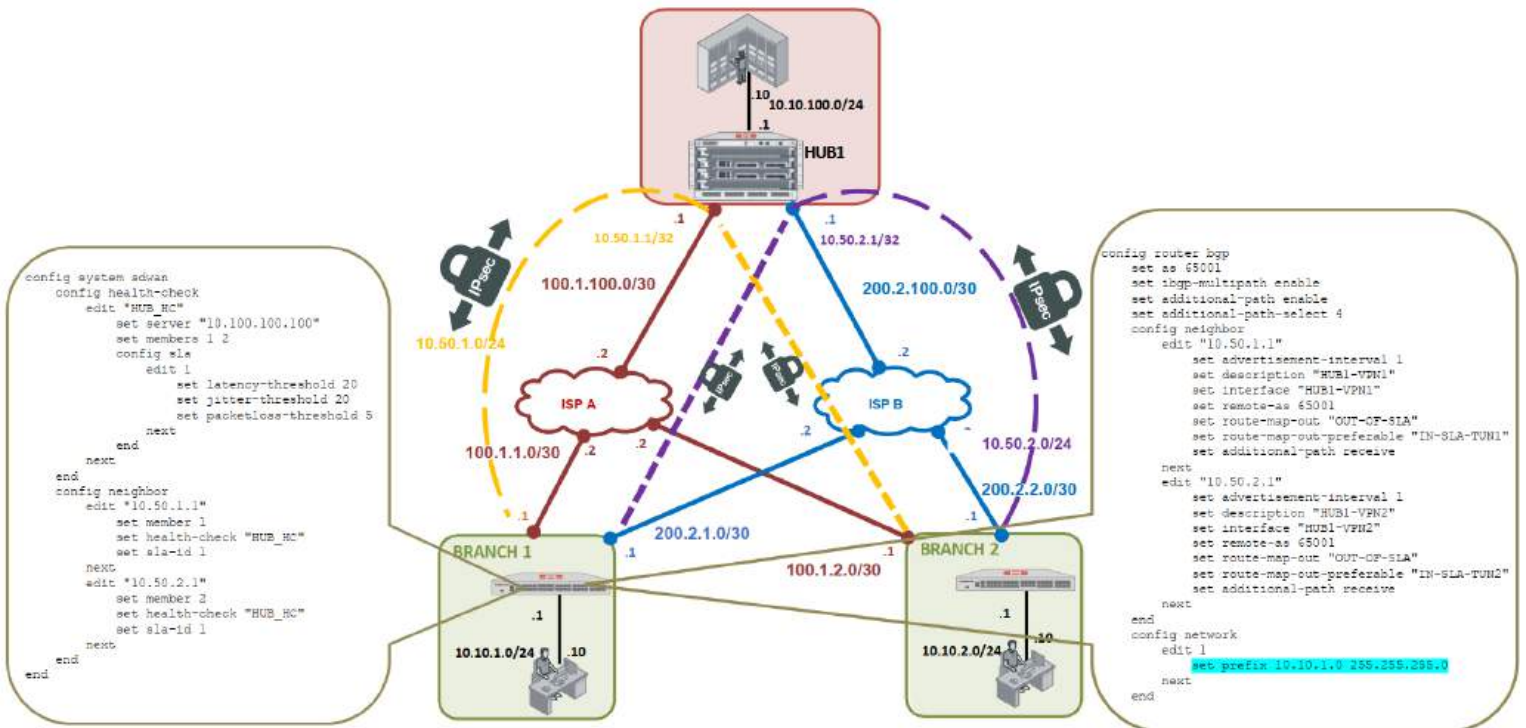
The **neighbor-ranges** and **neighbor-group** settings are configured for peering relationships to be established without defining individual peers. Branches have their tunnel interfaces configured within the range of the BGP peer and match the defined prefix under the neighbor-range defined by the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

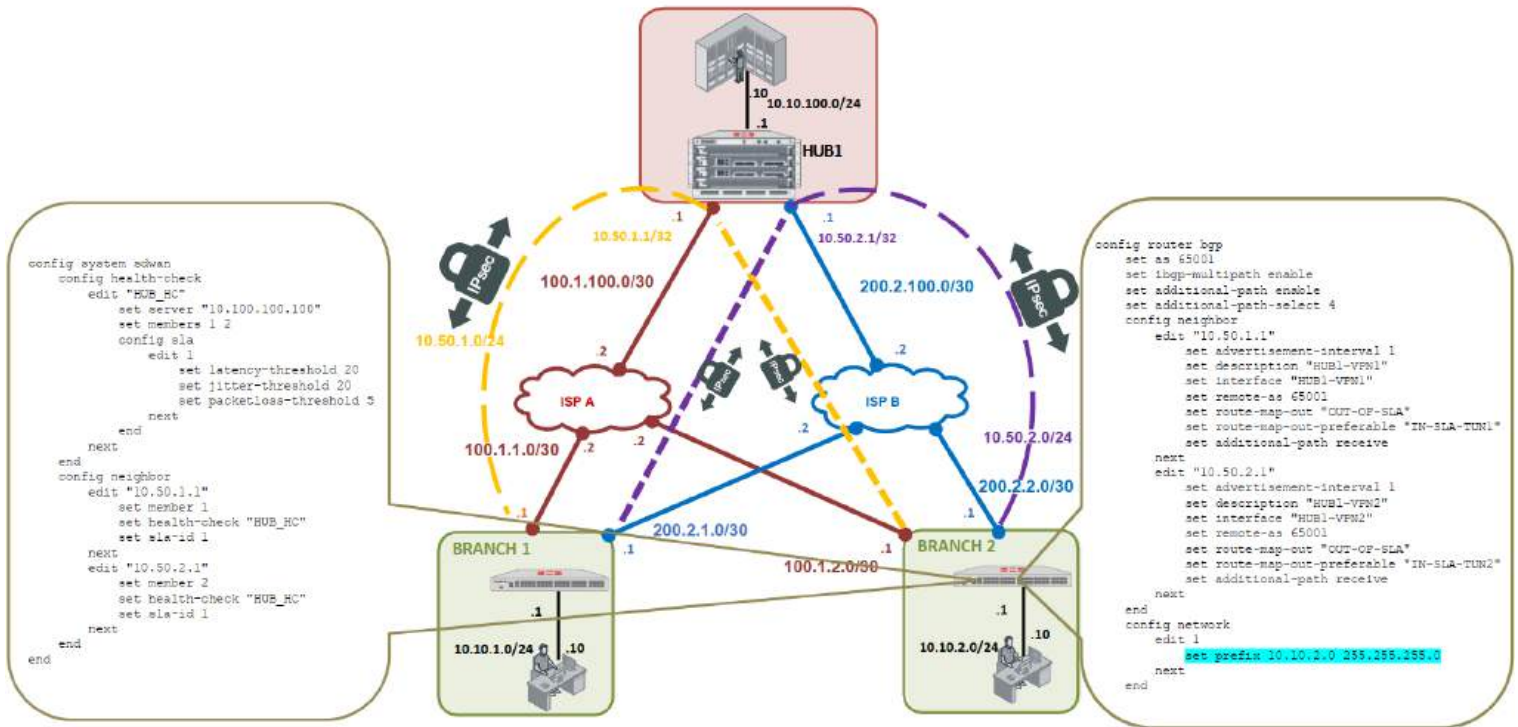
BRANCH1 BGP Configuration

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.

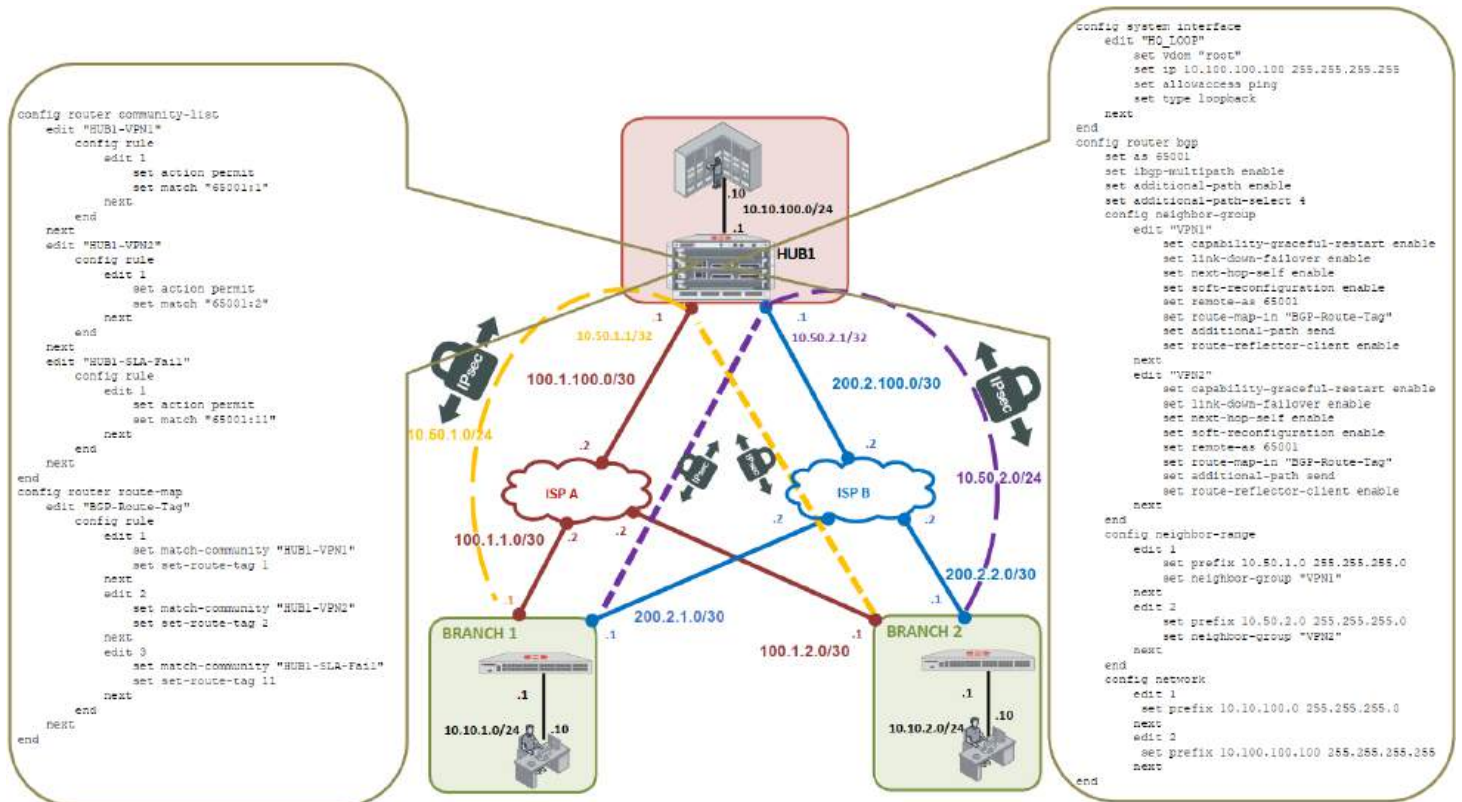


BRANCH2 BGP Configuration

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         0           100    32768    0 i <-/1>
*> 10.100.100.100/32
                0.0.0.0         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        0      2 i <-/2>
*>i10.10.1.0/24 10.50.1.11    100         0        0      1 i <-/1>
*>i10.10.2.0/24 10.50.2.11    100         0        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        0      2 i <-/2>
*>i10.10.2.0/24 10.50.2.11    100         0        0      2 i <-/2>
*>i10.10.2.0/24 10.50.1.10    100         0        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      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      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 add-on 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.

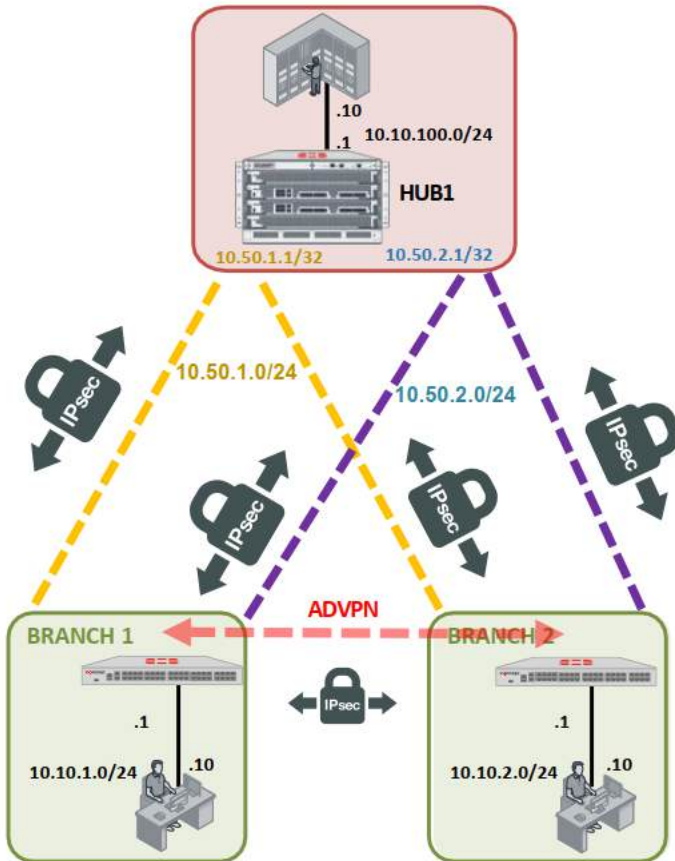
The initial communication goes through the Hub and if properly configured, the hub offers the spoke a shortcut. The spoke then negotiates a new tunnel between them on their overlay network. Once formed a shortcut tunnel would remain up even if the parent IPsec tunnel through the hub went down. This is 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
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
C 10.50.2.11/32 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 have the following syntax **<phase1name>_<index>** e.g. **HUB1_VPN2_0**

Name	Remote Gateway	Peer ID	Incoming Data	Outgoing Data
HUB1-VPN2_0	200.2.2.1	MSA_Tun2	1.17 kB	1.09 kB
HUB1-VPN1	100.1.100.1	NAIROBI-TUN1	1.35 MB	1.35 MB
HUB1-VPN2	200.2.100.1	NAIROBI-TUN2	1.35 MB	1.35 MB

Name	Remote Gateway	Peer ID	Incoming Data	Outgoing Data
HUB1-VPN2_0	200.2.1.1	Nakuru_Tun2	1.09 kB	1.21 kB
HUB1-VPN1	100.1.100.1	NAIROBI-TUN1	1.30 MB	1.30 MB
HUB1-VPN2	200.2.100.1	NAIROBI-TUN2	1.31 MB	1.31 MB

```
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), bandwidth-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), bandwidth-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), bandwidth-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), bandwidth-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

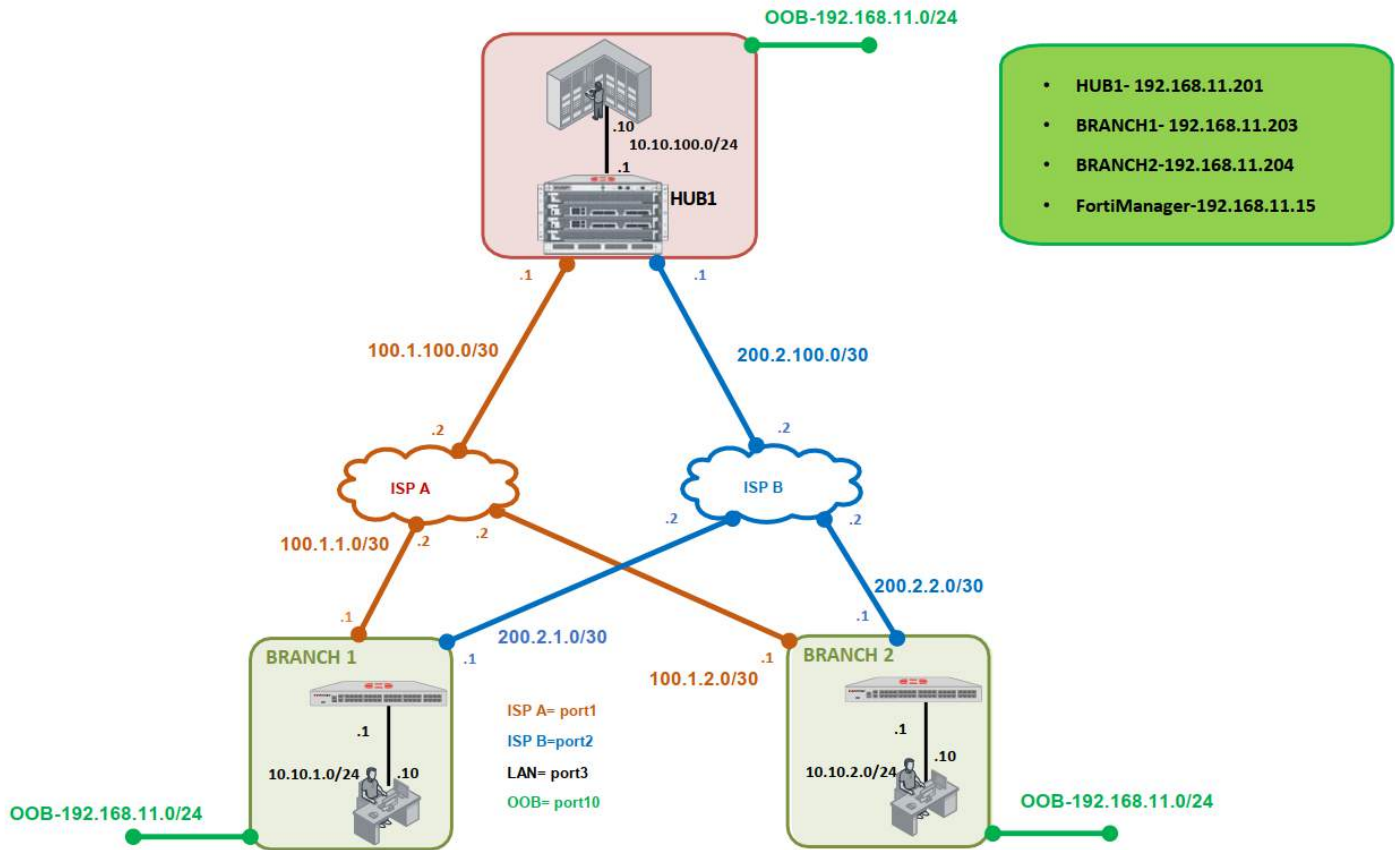
5. Which IP address do the branches reference as the bgp neighbor at the HUB.
 - a) Overlay static interface ip
 - b) Underlay static IP
 - c) Loopback at the HUB

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.

Central Management Settings

Settings Info

Status: Enabled Disabled

Type: On-Premises FortiGate Cloud FortiManager Cloud

Connection status: Connected Refresh

Verified serials: FMG-VMTM24015045

Mode: Normal Backup

IP/Domain name: 192.168.11.15

SD-WAN Zones SD-WAN Rules Performance SLAs

Bandwidth Volume Sessions

Download

No results

+ Create New Edit Delete

Interfaces	Gateway
virtual-wan-link	
DIA	
port1	100.1.100.2
port2	200.2.100.2

Install Preview of HUB1

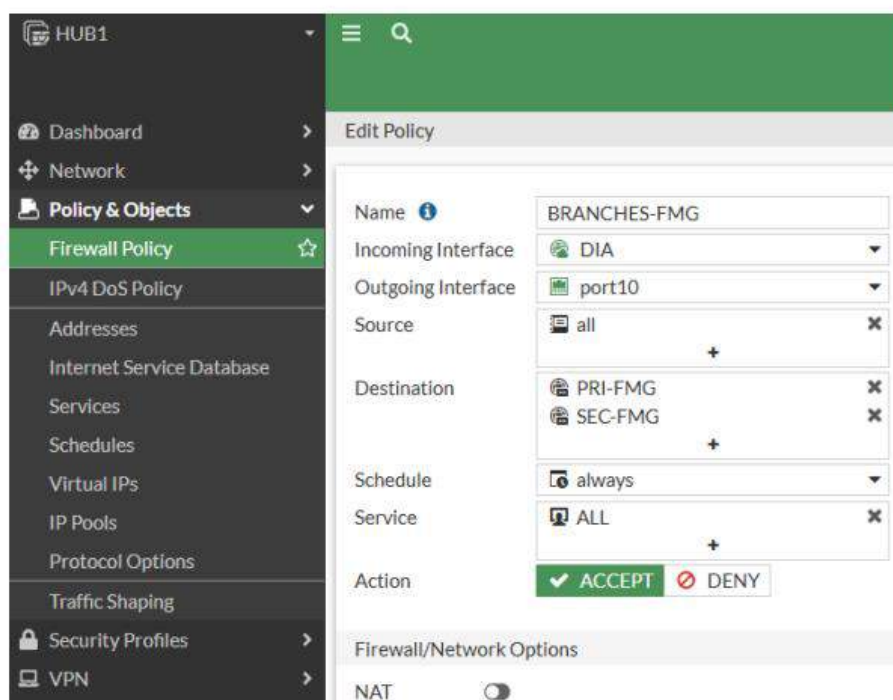
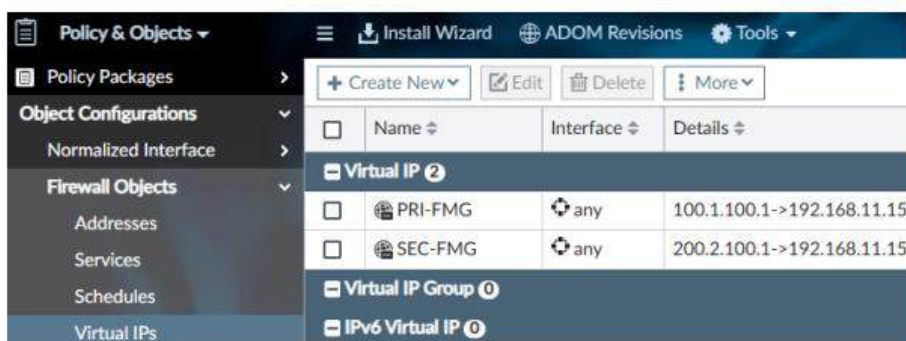
Assigned Devices: HUB1

HUB1

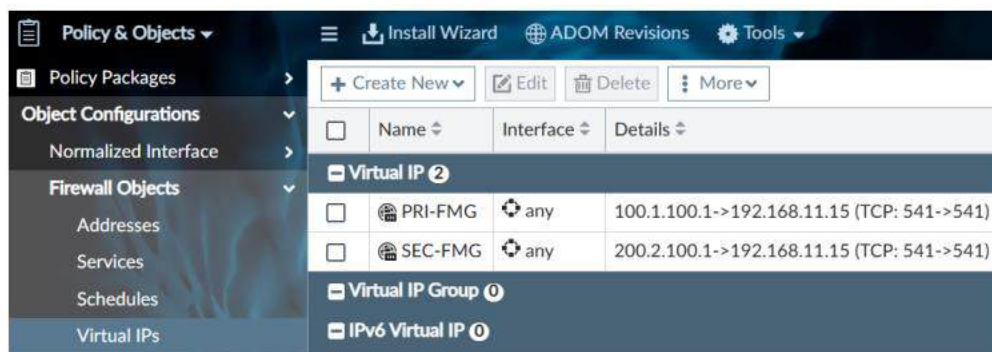
Search...

```
1 config system sdwan
2   config zone
3     edit "DIA"
4       next
5     end
6   config members
7     edit 1
8       set interface "port1"
9       set zone "DIA"
10      set gateway 100.1.100.2
11     next
12    edit 2
13      set interface "port2"
14      set zone "DIA"
15      set gateway 200.2.100.2
16    next
17  end
18 end
```

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



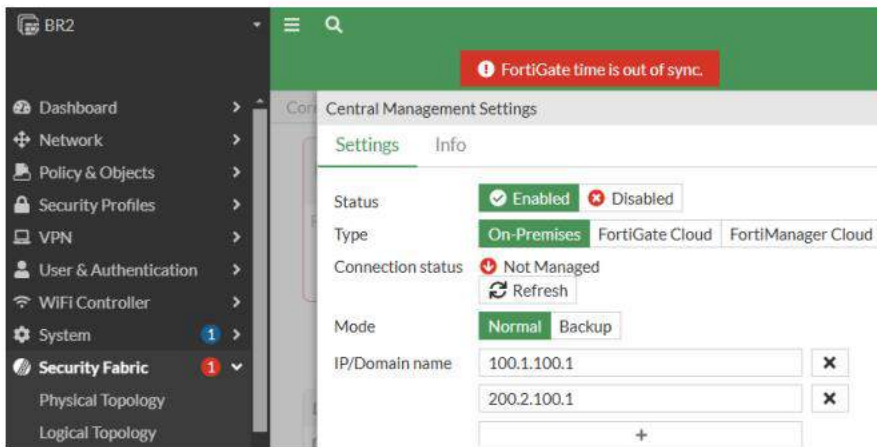
If a single public IP is available at the primary and secondary links, port address translation (PAT) must be used to avoid NAT directing all inbound traffic to FortiManager, blocking all branch devices dialing in over IPsec since all traffic would be directed to FortiManager including IPsec terminations.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



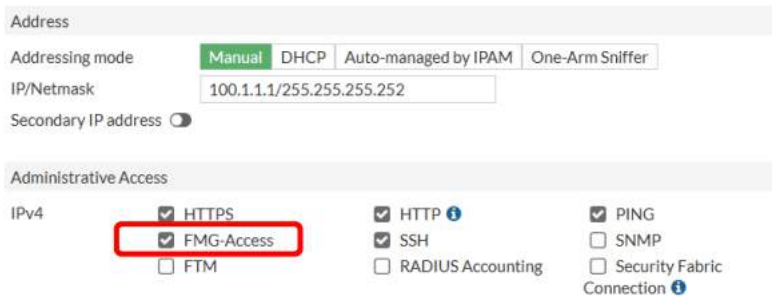
Since there are two ISP links at the HUB, the branches are configured to access both for redundancy. When multiple addresses are listed, the FortiGate attempts to establish the FGFM tunnel using the first IP listed, and if it is unreachable, it tries each subsequent IP until the tunnel is established.

```
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

The screenshot shows the FortiManager Device Manager interface. On the left, a sidebar lists 'Managed FortiGate (3)' with sub-items BR1, BR2, and HUB1. The main area features two donut charts: 'Connectivity' showing 3 devices connected, and 'Device Config Status' showing 1 synchronized and 2 auto-updated. Below the charts is a table with columns for Device Name, Config Status, Host Name, IP Address, Platform, and Firmware Version.

Device Name	Config Status	Host Name	IP Address	Platform	Firmware Version
BR1	Auto-update	BR1	100.1.1.1	FortiGate-VM64-KVM	FortiGate 7.2.8.build1639 (GA) (Mature) Low Vulnerability
BR2	Synchronized	BR2	100.1.2.1	FortiGate-VM64-KVM	FortiGate 7.2.8.build1639 (GA) (Mature) Low Vulnerability
HUB1	Auto-update	HUB1	192.168.11.201	FortiGate-VM64-KVM	FortiGate 7.2.8.build1639 (GA) (Mature) Low Vulnerability

```

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.

Time	Severity	Message	Impact
2024/10/25 10:11:59	Warning	Failed to connect FortiManager 200.2.100.1	Central Management connectivity is inactive
2024/10/25 10:05:30	Notice	Connected to FortiManager 200.2.100.1	Central Management connectivity is active
2024/10/25 10:05:30	Warning	Tunnel to FortiManager is down	Central Management connectivity is inactive
2024/10/25 10:05:30	Warning	Failed to connect FortiManager 100.1.100.1	Central Management connectivity is inactive

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

The screenshot shows the 'Create New Device Group' dialog box in FortiManager. The 'Group Name' field contains 'Branches'. Below, a table lists the devices to be added to the group: BR1 and BR2, both of type 'Device' on the 'FortiGate-VM64-KVM' platform.

Device Name	Type	Platform	IP	Firmware Version
BR1	Device	FortiGate-VM64-...	100.1.1.1	FortiGate 7.2.8.build1639 (GA)
BR2	Device	FortiGate-VM64-...	100.1.2.1	FortiGate 7.2.8.build1639 (GA)

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.

The screenshot shows the 'Policy & Objects' configuration page in the Fortinet GUI. The 'Advanced' section is expanded to show 'Metadata Variables'. A table lists several variables:

Name	Default Value
branch_id	
hub1_ISPA	100.1.100.1
hub1_ISPB	200.2.100.1
local_id	
vm_interface_number	1

The screenshot shows the 'Edit Metadata Variables - branch_id' configuration page. It includes fields for Name, Description, and Default Value. Below these is a 'Per-Device Mapping' section with a table:

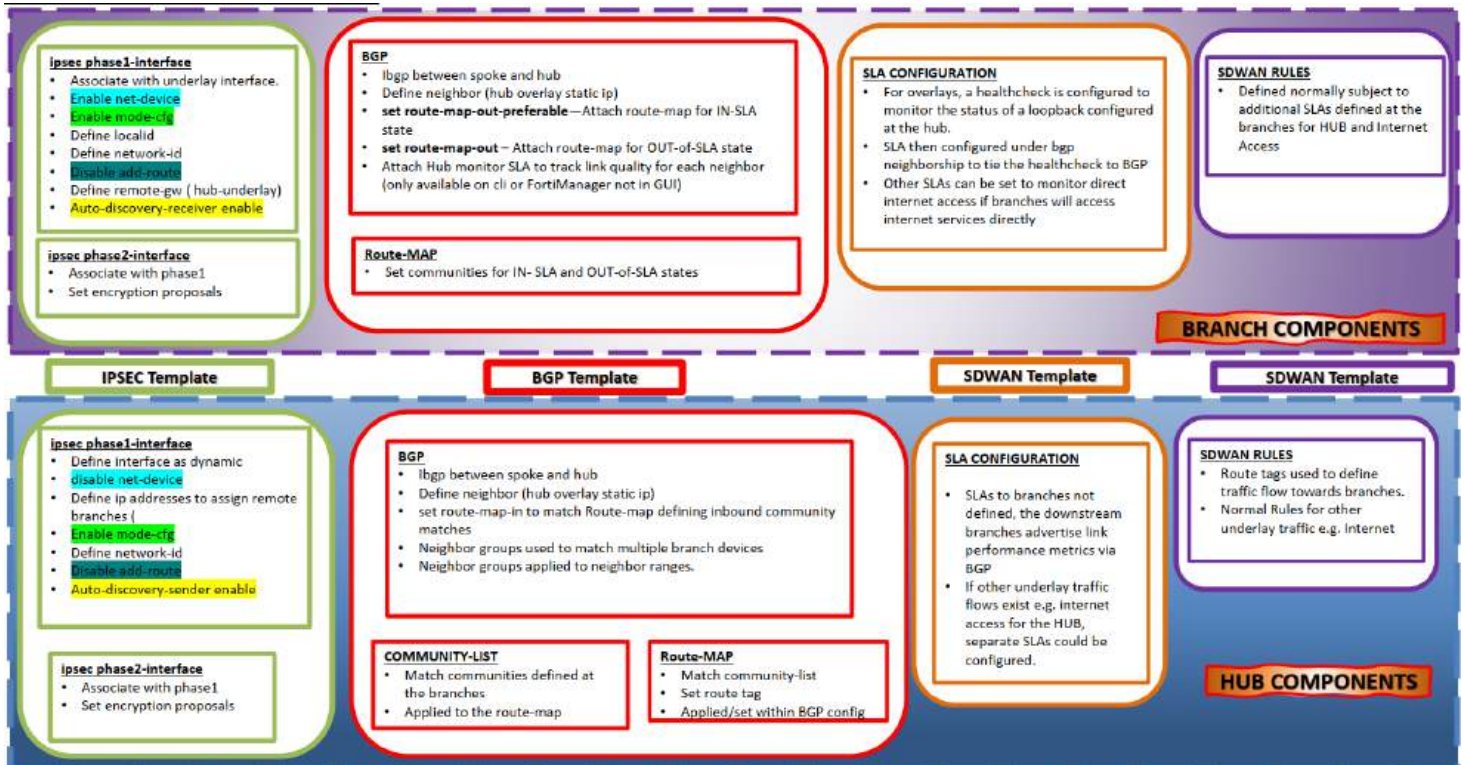
Mapped Device	Value
BR1 [root]	1
BR2 [root]	2
HUB1 [root]	100

The screenshot shows the 'Edit Metadata Variables - local_id' configuration page. It includes fields for Name, Description, and Default Value. Below these is a 'Per-Device Mapping' section with a table:

Mapped Device	Value
HUB1 [root]	Nairobi
BR2 [root]	Mombasa
BR1 [root]	Nakuru

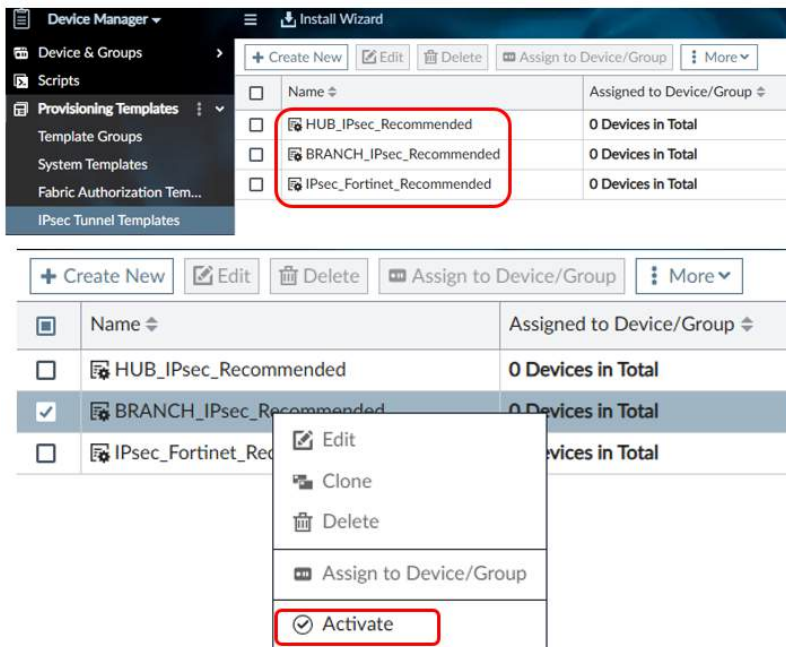
SD-WAN FortiManager role mapping.

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 screenshot shows the 'Activate BRANCH_IPsec_Recommended' dialog box with 'Template Name' set to 'BRANCH_IPsec' and 'Enable ADVPN' checked. Below it, the configuration for 'HUB1-VPN1' is shown with 'Outgoing Interface' as 'port1', 'Local ID' as '\$(local_id)_Tun1', 'Remote Gateway' as '\$(hub1_IPSA)', and 'Pre-shared Key' as a series of dots.

To the right is a table of templates:

	Name	Assigned to Device/Group
<input type="checkbox"/>	HUB_IPsec_Recommended	0 Devices in Total
<input checked="" type="checkbox"/>	BRANCH_IPsec_Recommended	0 Devices in Total
<input type="checkbox"/>	IPsec_Fortinet_Recommended	0 Devices in Total
<input type="checkbox"/>	BRANCH_IPsec	0 Devices in Total

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.

The screenshot shows the 'IPsec Template - BRANCH_IPsec' configuration window. The 'Name' is 'BRANCH_IPsec' and the 'Description' is empty. Below the configuration is a table of templates:

	Name	Type	Outgoing Interface	Local Interface
<input checked="" type="checkbox"/>	HUB1-VPN1	Static	port1	

To the right is the 'Edit IPsec Tunnel - HUB1-VPN1' configuration window. The 'Tunnel Name' is 'HUB1-VPN1'. The 'Network' section has 'Routing' set to 'Manual', 'Remote Device' set to 'IP Address', and 'Remote Gateway (IP Address)' set to '\$(hub1_IPSA)'. The 'Outgoing Interface' is 'port1' and 'Local ID' is '\$(local_id)_Tun1'. The 'Network Overlay' is checked and 'Network ID' is '1'. The 'Proposal' is 'aes256-sha256'. The 'Authentication' section has 'Authentication Method' set to 'Pre-shared Key' and 'Pre-shared Key' as a series of dots. The 'Tunnel Interface Setup' section has 'IP' and 'Remote IP' fields. The 'Phase2 Interface' section has a table:

	Name	Keep Alive	Key Life	Proposal
<input type="checkbox"/>	HUB1-VPN1	<input checked="" type="checkbox"/>	43200	aes256-sha256

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.

The screenshot shows the Fortinet SD-WAN installation wizard interface. At the top, the 'Device Manager' sidebar is visible with 'Provisioning Templates' expanded. The main window shows a table of template groups:

Template Group Name	Provisioning Templates	Assigned to Device/Group
<input type="checkbox"/> HUB1-TG	HUB1-SDWAN	1 Device in Total HUB1 [root]
<input checked="" type="checkbox"/> BRANCHES-TG	BRANCH_IPsec	1 Device in Total BR1 [root]

Below this, the 'Install Wizard - Choose What to Install (1/4)' dialog is shown with 'Install Device Settings (only)' selected. A red dashed arrow labeled 'NEXT' points to the 'Install Wizard - Validate Devices (3/4)' dialog. In this dialog, the 'Install Preview' option is highlighted with a red box. Below it, a table shows the status of the device BR1:

Device Name	Status
BR1	Connection Up

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

Edit IPsec Tunnel - HUB1-VPN1

Tunnel Name: HUB1-VPN1

Network

Routing: Manual Automatic

Remote Device: IP Address Dynamic DNS Dynamic

Remote Gateway (IP Address): \${hub1_ISPA}

Outgoing Interface: port1

Local ID: \${local_id}_Tun1

Network Overlay:

Network ID: 1

Proposal: aes256-sha256

FEC Health Check:

Authentication

Authentication Method: Pre-shared Key Signature

Pre-shared Key:

Tunnel Interface Setup

IP:

Remote IP:

Phase2 Interface

Name	Keep Alive	Key Life	Proposal
HUB1-VPN1	<input type="checkbox"/>	43200	aes256-sha256

OK



```
1 config vpn ipsec phase1-interface
2   edit "HUB1-VPN1"
3     set interface "port1"
4     set ike-version 2
5     set comments "VPN: HUB1-VPN1 [Created by IPSEC Template]"
6     set proposal aes256-sha256
7     set peertype any
8     set mode-cfg enable
9     set localid "Nakuru_Tun1"
10    set remote-gw 100.1.100.1
11    set idle-timeout enable
12    set net-device enable
13    set add-route disable
14    set auto-discovery-receiver enable
15    set psksecret ENC Z8Zpc/bwU2j1HxCFsp0zLVsmPXEWQvc6JVoYg8
16    set network-overlay enable
17    set network-id 1
18
19  next
20 end
21 config system interface
22   edit "HUB1-VPN1"
23     set vdom "root"
24     set type tunnel
25     set snmp-index 116
26     set interface "port1"
27
28  next
29 end
30 config vpn ipsec phase2-interface
31   edit "HUB1-VPN1"
32     set phaseiname "HUB1-VPN1"
33     set proposal aes256-sha256
34     set auto-negotiate enable
35     set comments "VPN: HUB1-VPN1 [Created by IPSEC Template]"
36
37  next
38 end
```

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.

IPsec Template - BRANCH_IPsec

Name: BRANCH_IPsec

Description:

+ Create New | Edit | Delete | More

<input checked="" type="checkbox"/>	Name	Type	Outgoing Interface
<input checked="" type="checkbox"/>	HUB1-VPN1	Static	port1

Clone

Edit Phase 2 Interface

Name: HUB1-VPN2

Keep Alive:

Key Life: 43200

Proposal: aes256-sha256

IPsec Template - BRANCH_IPsec

Name: BRANCH_IPsec

Description:

+ Create New | Edit | Delete | More

<input type="checkbox"/>	Name	Type	Outgoing Interface
<input type="checkbox"/>	HUB1-VPN1	Static	port1
<input checked="" type="checkbox"/>	HUB1-VPN2	Static	port2

Clone IPsec Tunnel - Clone_HUB1-VPN1

Tunnel Name: HUB1-VPN2

Network

Routing: Manual | Automatic

Remote Device: IP Address | Dynamic DNS | Dynamic

Remote Gateway (IP Address): \$(hub1_ISPB)

Outgoing Interface: port2

Local ID: \$(local_id)_Tun2

Network Overlay:

Network ID: 2

Proposal: aes256-sha256

FEC Health Check:

Authentication Method: Pre-shared Key | Signature

Pre-shared Key:

Tunnel Interface Setup

IP:

Remote IP:

Phase2 Interface

<input checked="" type="checkbox"/>	Name	Keep Alive
<input checked="" type="checkbox"/>	HUB1-VPN1	<input type="checkbox"/>

Branch1 HUB1-VPN1 & HUB1-VPN2 Comparison
 The variations between the VPNs are highlighted.

Edit IPsec Tunnel - HUB1-VPN1

Tunnel Name: HUB1-VPN1

Network

Routing: **Manual** Automatic

Remote Device: **IP Address** Dynamic DNS Dynamic

Remote Gateway (IP Address): \$(hub1_ISPA)

Outgoing Interface: port1

Local ID: \$(local_id)_Tun1

Network Overlay:

Network ID: 1

Proposal: aes256-sha256

FEC Health Check:

Authentication

Authentication Method: **Pre-shared Key** Signature

Pre-shared Key:

Tunnel Interface Setup

IP:

Remote IP:

Phase2 Interface

<input type="checkbox"/>	Name	Keep Alive	Key Life	Proposal
<input type="checkbox"/>	HUB1-VPN1	<input type="checkbox"/>	43200	aes256-sha256

OK

Edit IPsec Tunnel - HUB1-VPN2

Tunnel Name: HUB1-VPN2

Network

Routing: **Manual** Automatic

Remote Device: **IP Address** Dynamic DNS Dynamic

Remote Gateway (IP Address): \$(hub1_ISPB)

Outgoing Interface: port2

Local ID: \$(local_id)_Tun2

Network Overlay:

Network ID: 2

Proposal: aes256-sha256

FEC Health Check:

Authentication

Authentication Method: **Pre-shared Key** Signature

Pre-shared Key:

Tunnel Interface Setup

IP:

Remote IP:

Phase2 Interface

<input type="checkbox"/>	Name	Keep Alive	Key Life	Proposal
<input type="checkbox"/>	HUB1-VPN2	<input type="checkbox"/>	43200	aes256-sha256

OK

Branch1 IPsec HUB1-VPN2 Config Preview

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

Edit IPsec Tunnel - HUB1-VPN2

Tunnel Name: HUB1-VPN2

Network

Routing: Manual Automatic

Remote Device: IP Address Dynamic DNS Dynamic

Remote Gateway (IP Address): \${hub1_ISPB}

Outgoing Interface: port2

Local ID: \${local_id}_Tun2

Network Overlay:

Network ID: 2

Proposal: aes256-sha256

FEC Health Check:

Authentication

Authentication Method: Pre-shared Key Signature

Pre-shared Key:

Tunnel Interface Setup

IP:

Remote IP:

Phase2 Interface

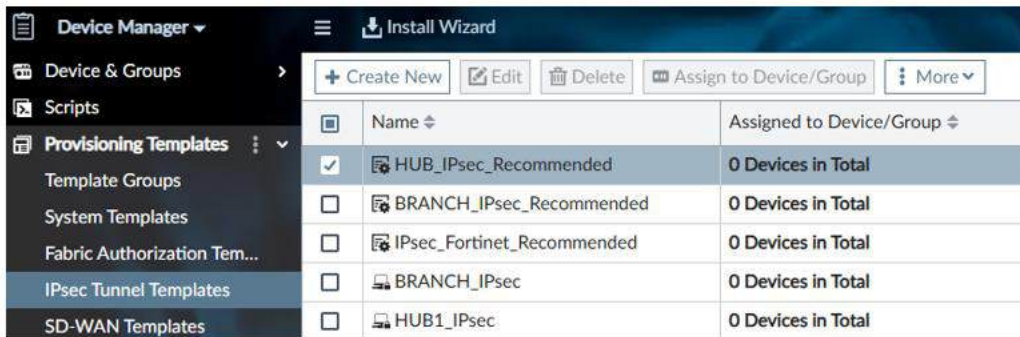
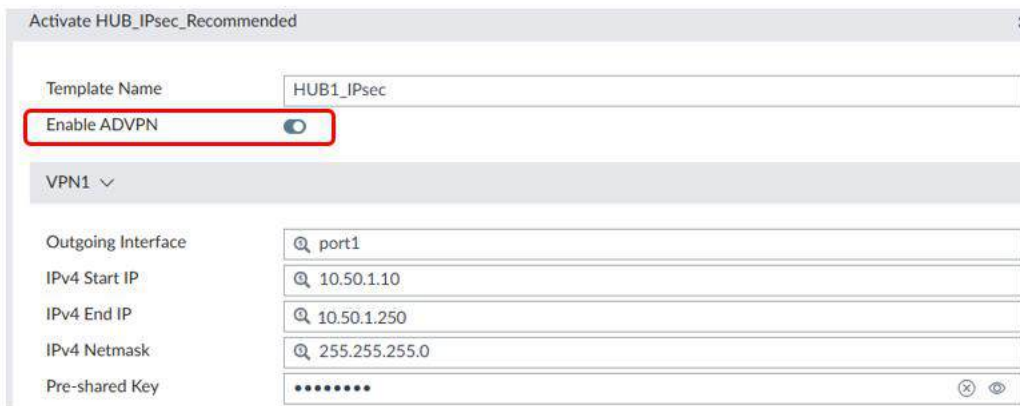
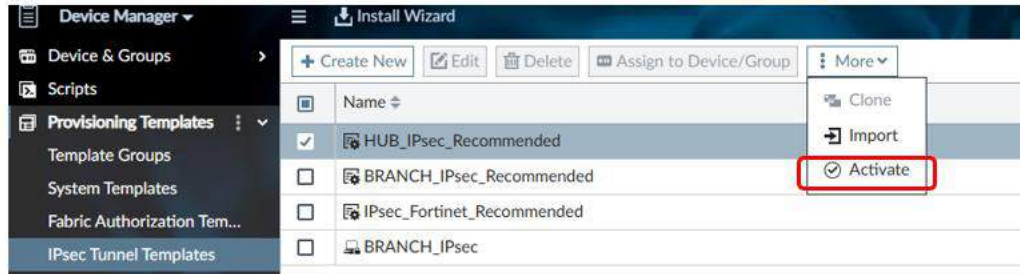
Name	Keep Alive	Key Life	Proposal
HUB1-VPN2	<input type="checkbox"/>	43200	aes256-sha256

OK

```
1 config vpn ipsec phase1-interface
2   edit "HUB1-VPN2"
3     set interface "port2"
4     set ike-version 2
5     set comments "VPN: HUB1-VPN2 [Created by IPSEC Template]"
6     set proposal aes256-sha256
7     set peertype any
8     set mode-cfg enable
9     set localid "Nakuru_Tun2"
10    set remote-gw 200.2.100.1
11    set idle-timeout enable
12    set net-device enable
13    set add-route disable
14    set auto-discovery-receiver enable
15    set psksecret ENC Z8Zpc/bwU2j1HxCFsp0zLVsmpXEWQvc6JVoYq8
16    set network-overlay enable
17    set network-id 2
18    set auto-discovery-shortcuts dependent
19  next
20 end
21 config system interface
22   edit "HUB1-VPN2"
23     set vdom "root"
24     set type tunnel
25     set snmp-index 117
26     set interface "port2"
27   next
28 end
29 config vpn ipsec phase2-interface
30   edit "HUB1-VPN2"
31     set phaseName "HUB1-VPN2"
32     set proposal aes256-sha256
33     set auto-negotiate enable
34     set comments "VPN: HUB1-VPN2 [Created by IPSEC Template]"
35   next
36 end
```

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.

The screenshot displays the Fortinet SD-WAN configuration interface, divided into two main sections: the IPsec Template configuration and the Tunnel configuration.

IPsec Template - HUB1_IPsec

- Name:** HUB1_IPsec
- Description:** (empty)
- Table:**

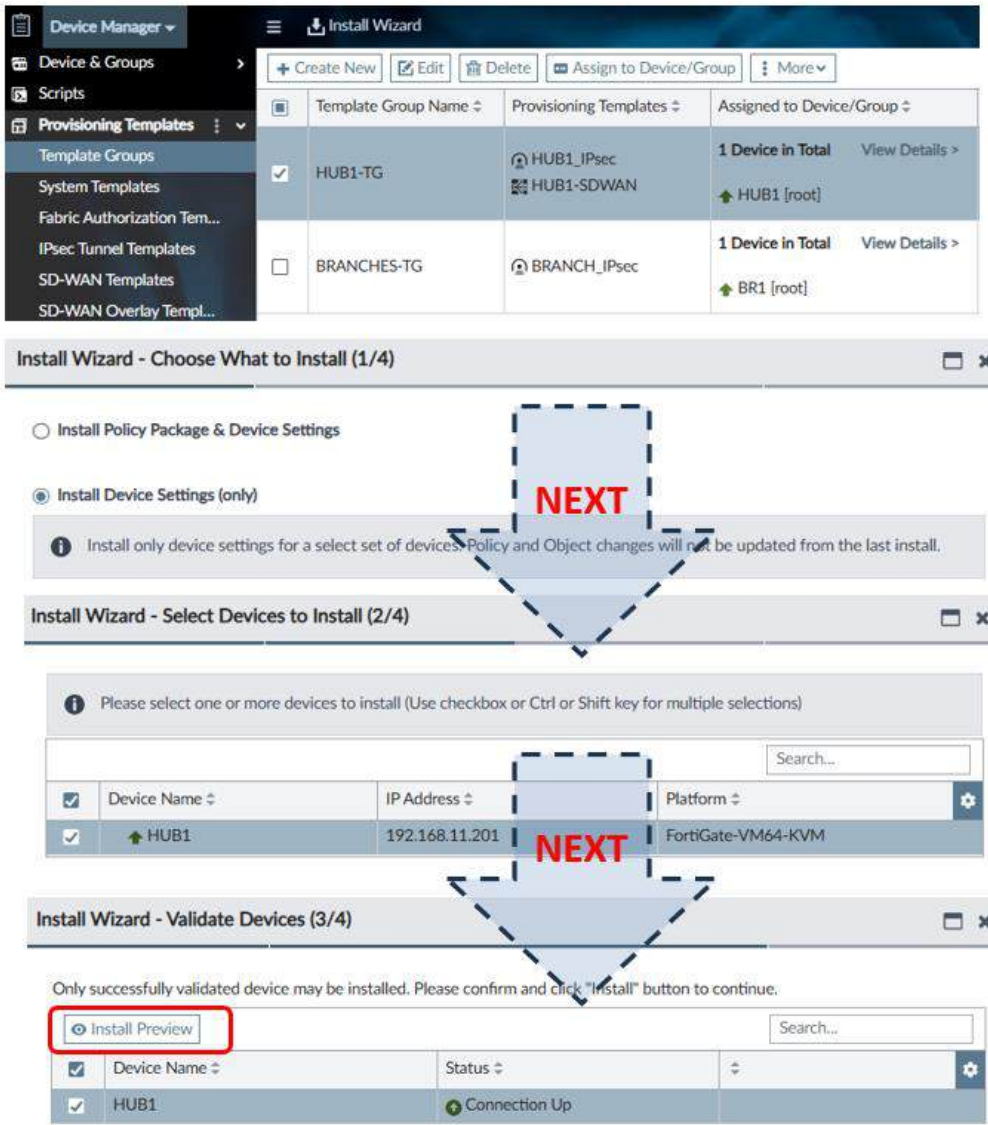
Name	Type	Outgoing Interface
VPN1	Dynamic	port1

Edit IPsec Tunnel - VPN1

- Tunnel Name:** VPN1
- Network:**
 - Routing:** Manual
 - Remote Device:** IP Address, Dynamic DNS, Dynamic
 - Outgoing Interface:** port1
 - Local ID:** \$(local_id)_Tun1
 - Network Overlay:** (checked)
 - Network ID:** 1
 - IPv4 Start IP:** 10.50.1.10
 - IPv4 End IP:** 10.50.1.250
 - IPv4 Netmask:** 255.255.255.0
 - Proposal:** aes256-sha256
 - FEC Health Check:** (checked)
- Authentication:**
 - Authentication Method:** Pre-shared Key, Signature
 - Pre-shared Key:** (masked)
- Tunnel Interface Setup:**
 - IP:** 10.50.1.1/32
 - Remote IP:** 10.50.1.253/24
- Phase2 Interface:**

Name	Keep Alive	Key Life	Proposal
VPN1	(disabled)	43200	aes256-sha256

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.

The image displays the Fortinet IPsec VPN configuration interface on the left and the corresponding CLI commands on the right. The interface is titled "Edit IPsec Tunnel - VPN1" and includes sections for Tunnel Name, Network, Routing, Remote Device, Outgoing Interface, Local ID, Network Overlay, Network ID, IPv4 Start/End IP/Netmask, Proposal, FEC Health Check, Authentication, Tunnel Interface Setup, and Phase2 Interface. A table at the bottom shows the configuration for the VPN1 tunnel.

The CLI commands on the right are as follows:

```
1 config vpn ipsec phase1-interface
2   edit "VPN1"
3     set type dynamic
4     set interface "port1"
5     set ike-version 2
6     set dpd on-idle
7     set comments "VPN: VPN1 [Created by IPSEC Template]"
8     set proposal aes256-sha256
9     set peertype any
10    set mode-cfg enable
11    set localid "Nairobi_Tun1"
12    set dpd-retryinterval 60
13    set net-device disable
14    set add-route disable
15    set auto-discovery-sender enable
16    set ipv4-start-ip 10.50.1.10
17    set ipv4-end-ip 10.50.1.250
18    set ipv4-netmask 255.255.255.0
19    set psksecret ENC Z8Zpc/bwU2j1HxCFsp0zLVsmPXEWQvc6J
20    set network-overlay enable
21    set network-id 1
22  next
23 end
24 config system interface
25   edit "VPN1"
26     set vdom "root"
27     set ip 10.50.1.1 255.255.255.255
28     set type tunnel
29     set remote-ip 10.50.1.253 255.255.255.0
30     set snmp-index 116
31     set interface "port1"
32  next
33 end
34 config vpn ipsec phase2-interface
35   edit "VPN1"
36     set phaseiname "VPN1"
37     set proposal aes256-sha256
38     set comments "VPN: VPN1 [Created by IPSEC Template]"
39  next
40 end
```

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.

IPsec Template - HUB1_IPsec

Name: HUB1_IPsec
Description:

+ Create New | Edit | Delete | More

<input checked="" type="checkbox"/>	Name	Type	Outgoing Interface
<input checked="" type="checkbox"/>	VPN1	Dynamic	port1

IPsec Template - HUB1_IPsec

Name: HUB1_IPsec
Description:

+ Create New | Edit | Delete | More

<input type="checkbox"/>	Name	Type	Outgoing Interface
<input type="checkbox"/>	VPN1	Dynamic	port1
<input type="checkbox"/>	VPN2	Dynamic	port2

Clone IPsec Tunnel - Clone_VPN1

Tunnel Name: VPN2

Network

Routing: **Manual** Automatic

Remote Device: IP Address Dynamic DNS **Dynamic**

Outgoing Interface: port2

Local ID: \${local_id}_Tun2

Network Overlay:

Network ID: 2

IPv4 Start IP: 10.50.2.10

IPv4 End IP: 10.50.2.250

IPv4 Netmask: 255.255.255.0

Proposal: aes256-sha256

FEC Health Check:

Authentication

Authentication Method: **Pre-shared Key** Signature

Pre-shared Key:

Tunnel Interface Setup

IP: 10.50.2.1/32

Remote IP: 10.50.2.253/24

Phase2 Interface

+ Create New | Edit | Delete

<input type="checkbox"/>	Name	Keep Alive
<input type="checkbox"/>	VPN2	<input type="checkbox"/>

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.

Edit IPsec Tunnel - VPN1

Tunnel Name: VPN1

Network

Routing: Manual Automatic

Remote Device: IP Address Dynamic DNS Dynamic

Outgoing Interface: port1

Local ID: \${local_id}_Tun1

Network Overlay:

Network ID: 1

IPv4 Start IP: 10.50.1.10

IPv4 End IP: 10.50.1.10

IPv4 Netmask: 255.255.255.0

Proposal: aes256-sha256

FEC Health Check:

Authentication

Authentication Method: Pre-shared Key Signature

Pre-shared Key:

Tunnel Interface Setup

IP: 10.50.1.1/32

Remote IP: 10.50.1.253/24

Phase2 Interface

<input type="checkbox"/>	Name ↕	Keep Alive ↕	Key Life ↕	Proposal ↕
<input type="checkbox"/>	VPN1	<input checked="" type="checkbox"/>	43200	aes256-sha256

Edit IPsec Tunnel - VPN2

Tunnel Name: VPN2

Network

Routing: Manual Automatic

Remote Device: IP Address Dynamic DNS Dynamic

Outgoing Interface: port2

Local ID: \${local_id}_Tun2

Network Overlay:

Network ID: 2

IPv4 Start IP: 10.50.2.10

IPv4 End IP: 10.50.2.250

IPv4 Netmask: 255.255.255.0

Proposal: aes256-sha256

FEC Health Check:

Authentication

Authentication Method: Pre-shared Key Signature

Pre-shared Key:

Tunnel Interface Setup

IP: 10.50.2.1/32

Remote IP: 10.50.2.253/24

Phase2 Interface

<input type="checkbox"/>	Name ↕	Keep Alive ↕	Key Life ↕	Proposal ↕
<input type="checkbox"/>	VPN2	<input checked="" type="checkbox"/>	43200	aes256-sha256

HUB1 IPsec VPN2 Config Preview

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

Edit IPsec Tunnel - VPN2

Tunnel Name: VPN2

Network

Routing: Manual Automatic

Remote Device: IP Address Dynamic DNS Dynamic

Outgoing Interface: port2

Local ID: \$(local_id)_Tun2

Network Overlay:

Network ID: 2

IPv4 Start IP: 10.50.2.10

IPv4 End IP: 10.50.2.250

IPv4 Netmask: 255.255.255.0

Proposal: aes256-sha256

FEC Health Check:

Authentication

Authentication Method: Pre-shared Key Signature

Pre-shared Key:

Tunnel Interface Setup

IP: 10.50.2.1/32

Remote IP: 10.50.2.253/24

Phase2 Interface

<input type="checkbox"/>	Name ⇅	Keep Alive ⇅	Key Life ⇅	Proposal ⇅
<input type="checkbox"/>	VPN2	<input type="checkbox"/>	43200	aes256-sha256

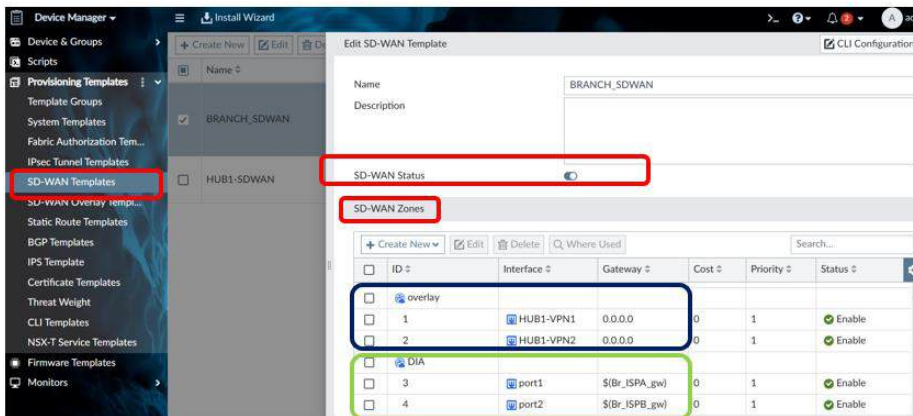
```
1 config vpn ipsec phase1-interface
2   edit "VPN2"
3     set type dynamic
4     set interface "port2"
5     set ike-version 2
6     set dpd on-idle
7     set comments "VPN: VPN2 [Created by IPSEC Template]"
8     set proposal aes256-sha256
9     set peertype any
10    set mode-cfg enable
11    set localid "Nairobi_Tun2"
12    set dpd-retryinterval 60
13    set net-device disable
14    set add-route disable
15    set auto-discovery-sender enable
16    set ipv4-start-ip 10.50.2.10
17    set ipv4-end-ip 10.50.2.250
18    set ipv4-netmask 255.255.255.0
19    set psksecret ENC Z8Zpc/bwU2j1HxCfSp0zLVsmPXEWQQvc6JV
20    set network-overlay enable
21    set network-id 2
22  next
23 end
24 config system interface
25   edit "VPN2"
26     set vdom "root"
27     set ip 10.50.2.1 255.255.255.255
28     set type tunnel
29     set remote-ip 10.50.2.253 255.255.255.0
30     set snmp-index 117
31     set interface "port2"
32  next
33 end
34 config vpn ipsec phase2-interface
35   edit "VPN2"
36     set phaseiname "VPN2"
37     set proposal aes256-sha256
38     set comments "VPN: VPN2 [Created by IPSEC Template]"
39  next
40 end
```

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.

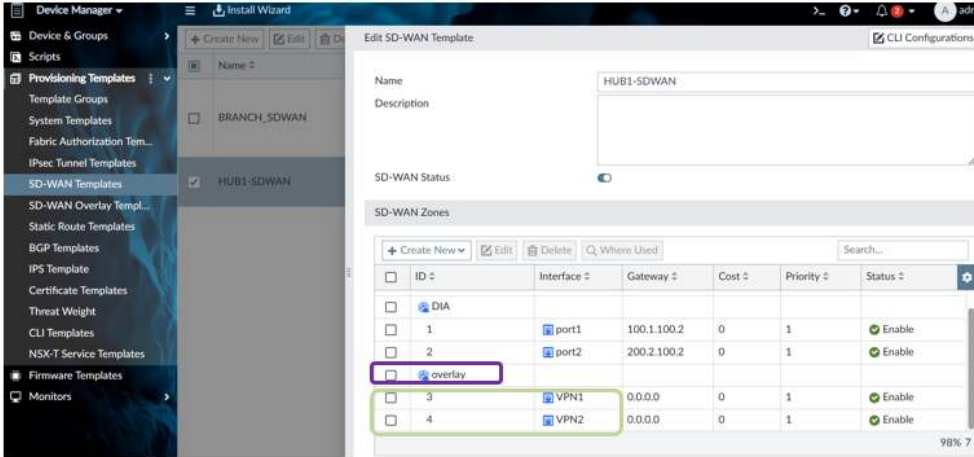
When adding the underlay network as SD-WAN interfaces, you must add the gateway ip addresses when defining them as sdwan member interfaces. Without gateways explicitly defined, the remote devices will not have default routes, any static default routes already configured will not be present in the routing 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.



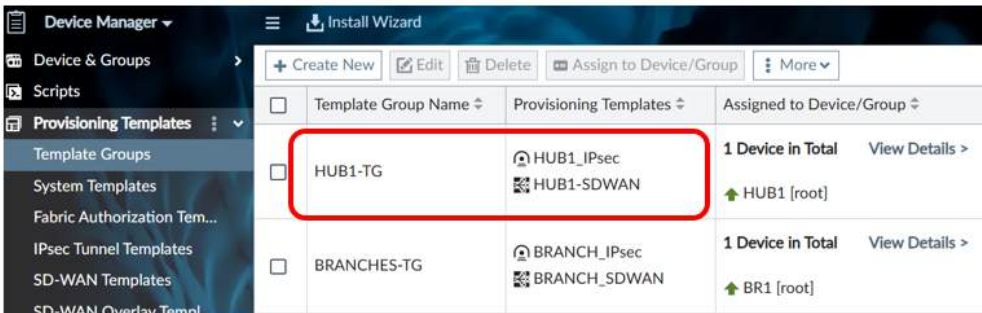
```
1 config system sdwan
2   config zone
3     edit "overlay"
4     next
5     edit "DIA"
6     next
7   end
8   config members
9     edit 1
10      set interface "HUB1-VPN1"
11      set zone "overlay"
12    next
13    edit 2
14      set interface "HUB1-VPN2"
15      set zone "overlay"
16    next
17    edit 3
18      set interface "port1"
19      set zone "DIA"
20      set gateway 100.1.1.2
21    next
22    edit 4
23      set interface "port2"
24      set zone "DIA"
25      set gateway 200.2.1.2
26    next
27  end
28 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.

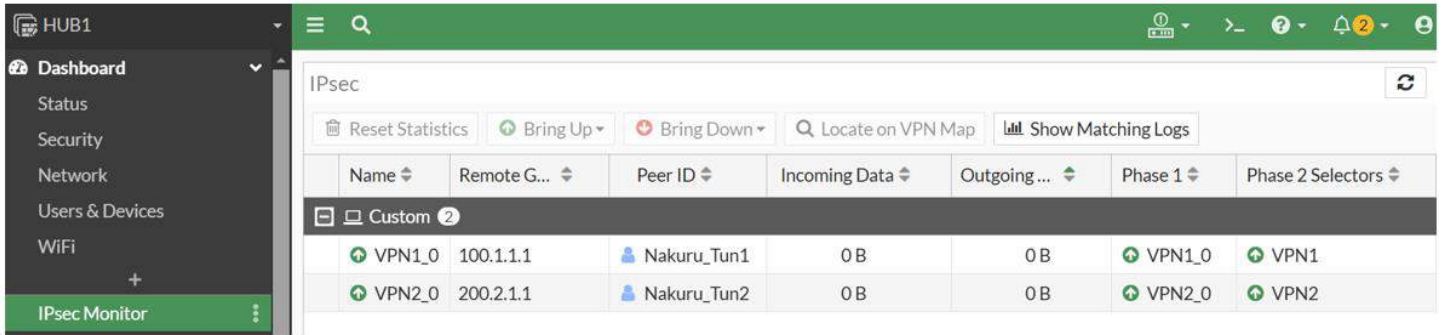


```
1 config system sdwan
2     config zone
3         edit "overlay"
4         next
5     end
6     config members
7         edit 3
8             set interface "VPN1"
9             set zone "overlay"
10        next
11       edit 4
12           set interface "VPN2"
13           set zone "overlay"
14       next
15   end
16 end
```



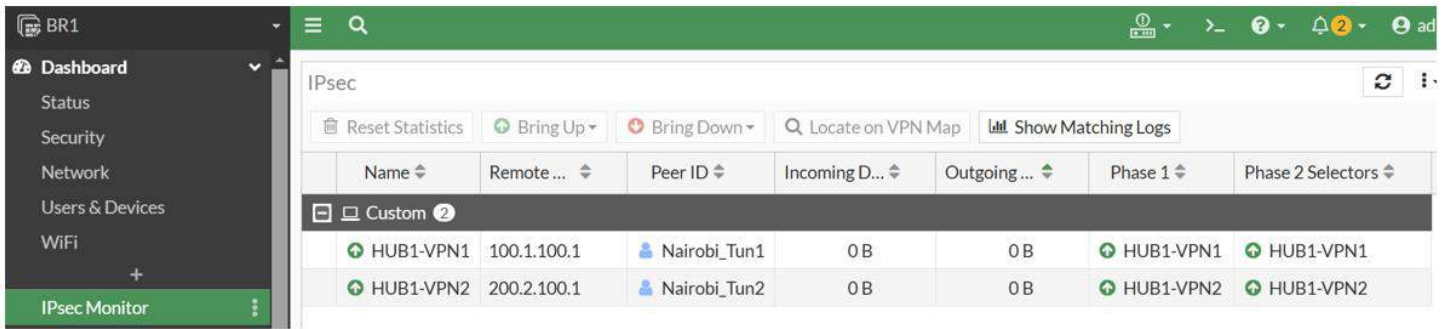
Requirement for firewall policies at the HUB and Branches

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.



The screenshot shows the IPsec Monitor interface for HUB1. The left sidebar contains a navigation menu with options: Dashboard, Status, Security, Network, Users & Devices, WiFi, and IPsec Monitor (highlighted). The main panel displays the IPsec status for two tunnels:

Name	Remote G...	Peer ID	Incoming Data	Outgoing ...	Phase 1	Phase 2 Selectors
VPN1_0	100.1.1.1	Nakuru_Tun1	0 B	0 B	VPN1_0	VPN1
VPN2_0	200.2.1.1	Nakuru_Tun2	0 B	0 B	VPN2_0	VPN2

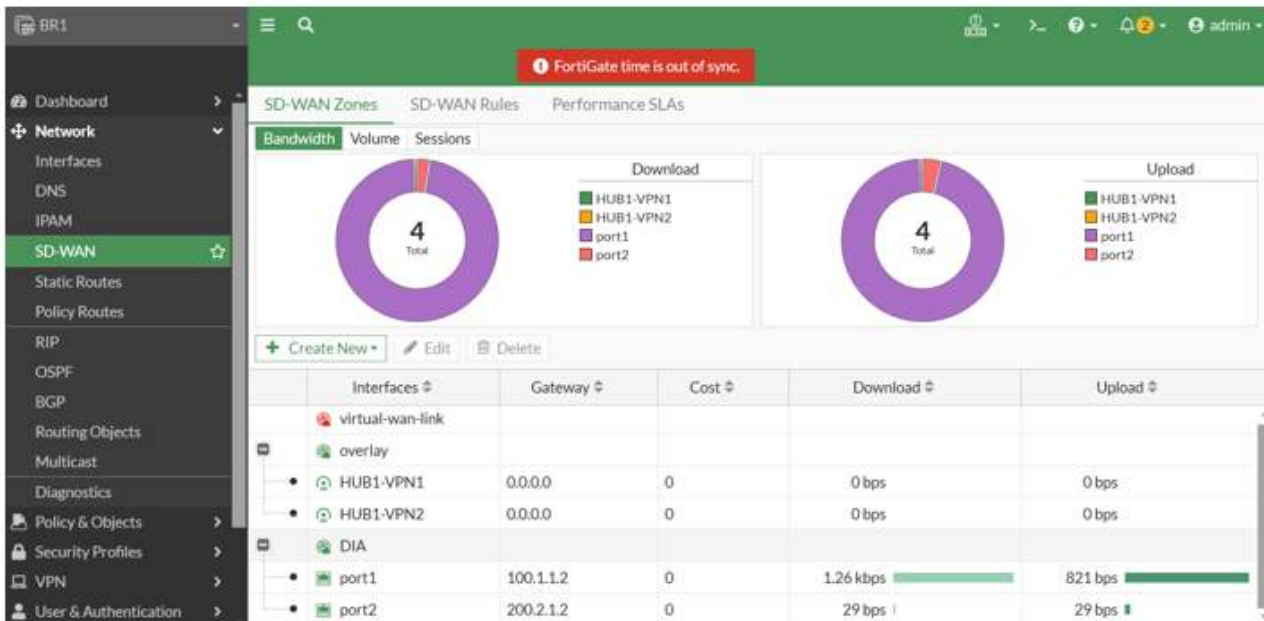
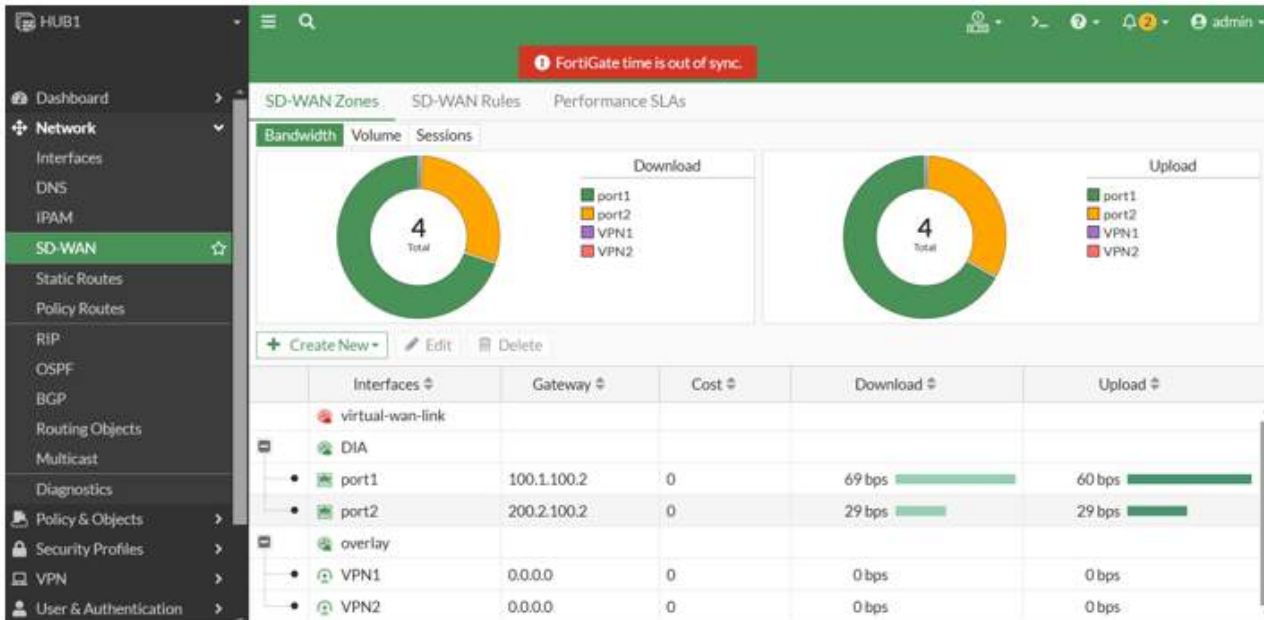


The screenshot shows the IPsec Monitor interface for BR1. The left sidebar contains a navigation menu with options: Dashboard, Status, Security, Network, Users & Devices, WiFi, and IPsec Monitor (highlighted). The main panel displays the IPsec status for two tunnels:

Name	Remote ...	Peer ID	Incoming D...	Outgoing ...	Phase 1	Phase 2 Selectors
HUB1-VPN1	100.1.100.1	Nairobi_Tun1	0 B	0 B	HUB1-VPN1	HUB1-VPN1
HUB1-VPN2	200.2.100.1	Nairobi_Tun2	0 B	0 B	HUB1-VPN2	HUB1-VPN2

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/reference 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.

The screenshot shows the Fortinet SD-WAN configuration interface. The 'Policy & Objects' menu is open, and the 'Normalized Interface' option is selected. The 'Edit Normalized Interface - LAN' dialog is displayed, showing the following fields:

- Name: LAN
- Description: (empty)
- Color: (Change button)
- Wildcard: (toggle off)
- Per-Platform Mapping: (dropdown arrow)
- Per-Device Mapping: (dropdown arrow)

Below the dialog, the 'Edit Normalized Interface - LAN' section shows the following fields:

- Name: LAN
- Description: (empty)
- Color: (Change button)
- Wildcard: (toggle off)

The 'Per-Platform Mapping' and 'Per-Device Mapping' sections are currently empty.

At the bottom, a table lists the mapped devices:

	Mapped Device	Details	Type	Addressing Mode	IP/Netmask	Shaping Profile
<input type="checkbox"/>	BR1 [root]	port3	Physical	Manual	0.0.0.0/0.0.0.0	
<input type="checkbox"/>	BR2 [root]	port3	Physical	Manual	0.0.0.0/0.0.0.0	
<input type="checkbox"/>	HUB1 [root]	port3	Physical	Manual	10.10.100.1/2...	

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.

The 'create New Address' dialog box shows the following configuration:

- Category: Address
- Name: Data
- Color: Change
- Type: Subnet
- IP/Netmask: 10.1.\$(branch_id).0/24
- Interface: any

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/address/Data/ : invalid subnet ip or mask

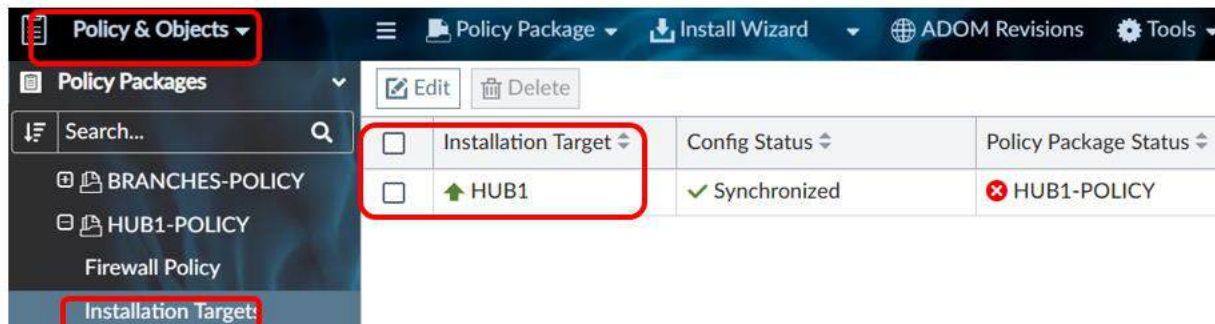
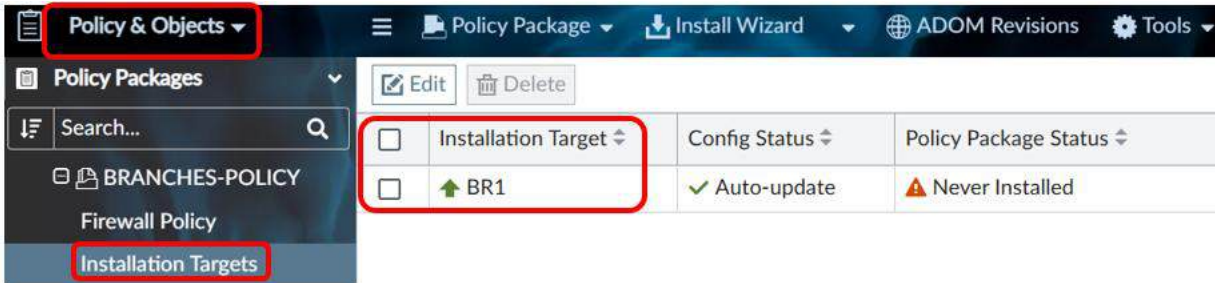
The 'Edit Metadata Variables - branch_id' dialog box shows the following configuration:

- Name: branch_id
- Description: (empty)
- Default Value: 222

Name	Type	Details
FortiManager	Address	IP/Netmask: 192.168.11.15/255.255.255.255
Data	Address	IP/Netmask: 10.1.\$(branch_id).0/255.255.255.0

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

The screenshot shows the Fortinet Policy & Objects interface with a list of firewall policies. The table below represents the data shown in the interface:

<input type="checkbox"/>	#	Name	From	To	Source	Destinatic	Schedule	Service	NAT	Action
<input type="checkbox"/>	1	LAN-TO-HQ&BRANCHES	LAN	overlay	all	all	always	ALL	Disabled	Accept
<input type="checkbox"/>	2	HQ&BRANCHES-TO-LAN	overlay	LAN	all	all	always	ALL	Disabled	Accept
<input type="checkbox"/>	3	DIA	LAN	DIA	all	all	always	ALL	Enabled	Accept
<input type="checkbox"/>	Implicit (4/4 Total:1)									
<input type="checkbox"/>	4	Implicit Deny	any	any	all	all	always	ALL		Deny

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

The top-left screenshot shows the 'Create New CLI Template' dialog in FortiManager. The 'Template Name' is 'HUB-CLI' and the 'Type' is 'CLI Script'. The 'Script Details' section contains the following CLI commands:

```

1 config system interface
2   edit "HUB_LOOP"
3     set vdom "root"
4     set ip 10.100.100.100 255.255.255.255
5     set allowaccess ping
6     set type loopback
7   next
8 end
  
```

The top-right screenshot shows the 'Install Preview of HUB1' dialog. The 'Assigned Devices' list includes 'HUB1'. The 'Search...' field contains the same CLI script as shown in the previous screenshot.

The bottom screenshot shows the 'Provisioning Templates' table in FortiManager. The 'HUB1-TG' group is selected, and the 'HUB-CLI' template is highlighted with a red box. The table shows that 'HUB-CLI' is assigned to 'HUB1 [root]'. Other templates shown include 'HUB1_IPsec', 'HUB1-SDWAN', 'BRANCH_IPsec', and 'BRANCH_SDWAN'.

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.

The top-left screenshot shows the 'Device Manager' interface in FortiManager. The 'Network' section is expanded, and the 'Loopback (1)' group is selected. The 'HUB_LOOP' interface is highlighted with a red box. The 'Normalized Interface' column is also highlighted with a red box.

The top-right screenshot shows the 'Edit Interface Map' dialog. The 'Loopback (1)' group is selected, and the 'HUB_LOOP' interface is highlighted. The 'Edit Interface Map' button is highlighted with a red box.

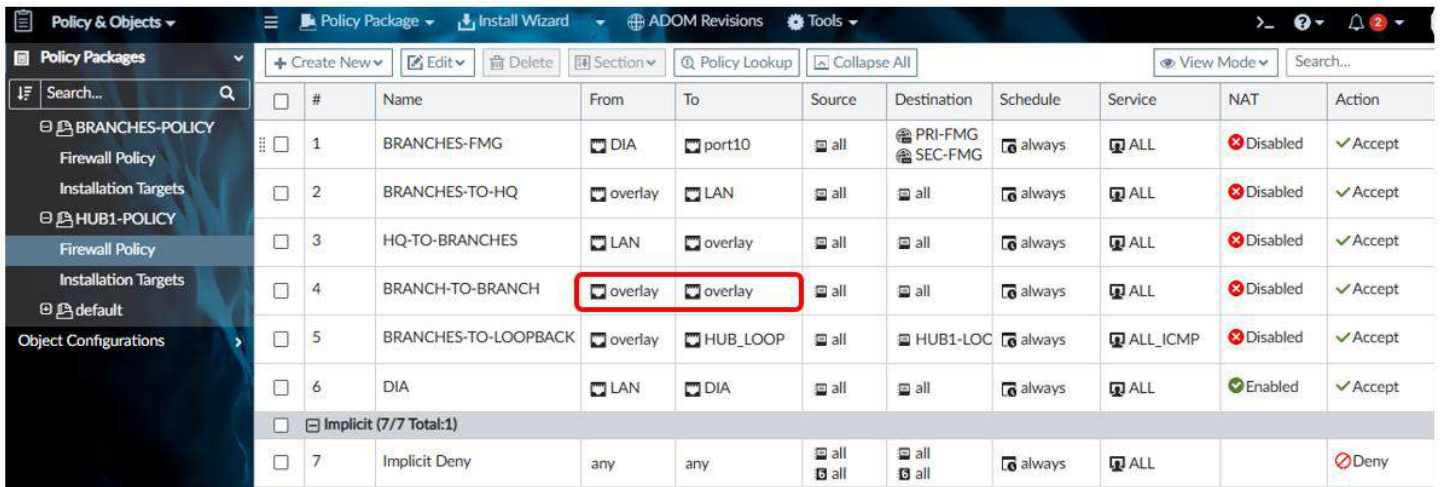
The bottom-left screenshot shows the 'create New Normalized Interface' dialog. The 'Name' field is set to 'HUB_LOOP'.

The bottom-right screenshot shows the 'Normalized Interface' table in FortiManager. The 'HUB_LOOP' interface is highlighted with a red box. The table shows that 'HUB_LOOP' is assigned to 'HUB_LOOP'.

HUB Firewall Policies

Six key policies are required at HUB1.

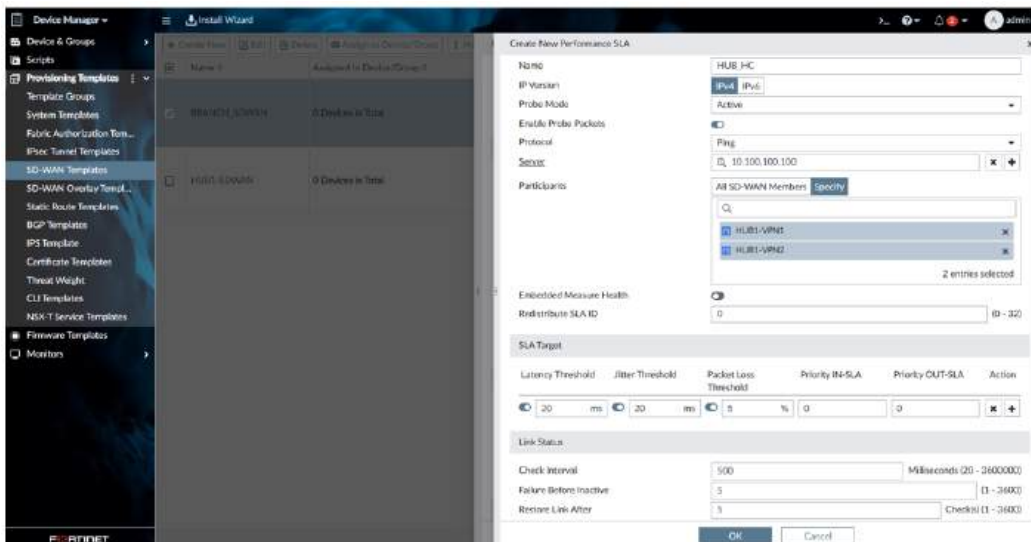
- FortiManager access from the Branches
- Branches to HQ LAN inbound on the overlay network
- Outbound rule to allow HQ to access branches on the overlay network
- Branch to Branch traffic hair pinned at the HUB **i.e. source and destination interfaces are the overlay network.**
- Branches to LOOPBACK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



#	Name	From	To	Source	Destination	Schedule	Service	NAT	Action
1	BRANCHES-FMG	DIA	port10	all	PRI-FMG SEC-FMG	always	ALL	Disabled	Accept
2	BRANCHES-TO-HQ	overlay	LAN	all	all	always	ALL	Disabled	Accept
3	HQ-TO-BRANCHES	LAN	overlay	all	all	always	ALL	Disabled	Accept
4	BRANCH-TO-BRANCH	overlay	overlay	all	all	always	ALL	Disabled	Accept
5	BRANCHES-TO-LOOPBACK	overlay	HUB_LOOP	all	HUB1-LOC	always	ALL_ICMP	Disabled	Accept
6	DIA	LAN	DIA	all	all	always	ALL	Enabled	Accept
Implicit (7/7 Total:1)									
7	Implicit Deny	any	any	all	all	always	ALL		Deny

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.



```
1 config system sdwan
2   config health-check
3     edit "HUB_HC"
4       set server 10.100.100.100
5       set members 2 1
6     config sla
7       edit 1
8         set latency-threshold 20
9         set jitter-threshold 20
10        set packetloss-threshold 5
11      next
12    end
13  next
14 end
15 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.

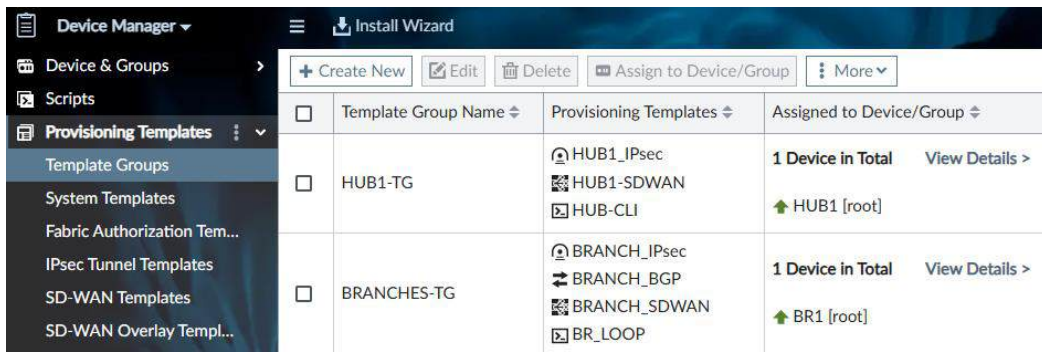
BGP is also used to define the networks to be advertised to the peers. iBGP is configured between the HUB and branches. At the point of activation, ADVPN is configured for BGP. The template is attached to the Branches template Group.

The screenshot shows the Fortinet Device Manager interface. In the left sidebar, 'BGP Templates' is highlighted. The main area displays a table of BGP templates:

Name	Assigned to Device/Group
HUB_BGP_Recommended	0 Devices in Total
BRANCH_BGP_Recommended	0 Devices in Total

The 'BRANCH_BGP_Recommended' template is selected, and a context menu is open over it, with the 'Activate' option highlighted. Below this, the 'Activate BRANCH_BGP_Recommended' dialog box is shown with the following configuration:

- Template Name: BRANCH_BGP
- Enable ADVPN: (highlighted with a red box)
- Local AS: 65001
- Router ID: 10.100.100.\${branch_id}
- Neighbor:
 - IP: 10.50.1.1
 - Remote AS: 65001
- Networks:
 - Prefix: 10.10.\${branch_id}.0/24



Branch1 BGP Neighbor1 Configuration Preview

Edit BGP Template ✕

Name:

Description:

Local As:

Router ID:

Neighbors

IP	Remote AS
<input type="text" value="10.50.1.1"/>	<input type="text" value="65001"/>

Networks

IP/Netmask	Action
<input type="text" value="10.10.\${branch_id}.0/24"/>	<input type="button" value="✕"/>

IPv6 Networks:

```

1 config router bgp
2   set as 65001
3   set router-id 10.100.100.1
4   set ibgp-multipath enable
5   set additional-path enable
6   set graceful-restart enable
7   set additional-path-select 255
8   config neighbor
9     edit "10.50.1.1"
10      set advertisement-interval 1
11      set capability-graceful-restart enable
12      set link-down-failover enable
13      set soft-reconfiguration enable
14      set description "HUB1-VPN1"
15      set interface "HUB1-VPN1"
16      set remote-as 65001
17      set connect-timer 10
18      set additional-path receive
19    next
20  end
21 config network
22   edit 1
23     set prefix 10.10.1.0 255.255.255.0
24   next
25 end
26 set recursive-next-hop enable
27 end

```

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.

Edit Neighbor

IP: 10.50.1.1
Remote AS: 65001
Password: *****
Interface: HUB1-VPN1
Update Source:
Graceful Restart Time: 0
Advertisement Interval: 1
Route Map Out Preferable: None
Additional Path: Receive
Connect Timer: 10
Bidirectional Forwarding Detection:
Link Down Failover:
Shutdown Enable:
IPv4 Filtering:
Filter List In:
Filter List Out:
Distribute List In:
Distribute List Out:
Prefix List In:
Prefix List Out:
Route Map In:
Route Map Out:

Edit Route Map

Name: **IN-SLA-TUNE**
Comments:
Rules:
+ Create New | Edit | Delete | Permit | Deny | Search...

ID	Action	Interface	Match IP Rules	Match Next Hop Rules
1	Permit			

Edit Route Map Rule
ID: 1
Action: Deny | **Permit**
IP Address Rule Variables:
Match interface:
Match IP address:
Match IPv6 address:
Match next hop router IP address:
Match next hop router IPv6 address:
Set next hop router IP address:
Set next hop router local IPv6 address:
AS Path Rule Variables:
Match AS path list: Set AS path: Action:
Community Rule Variables:
Match community:
Set community delete:
Set Community: 65001:1

Edit Neighbor

IP: 10.50.1.1
Remote AS: 65001
Password: *****
Interface: HUB1-VPN1
Update Source:
Graceful Restart Time: 0
Advertisement Interval: 1
Route Map Out Preferable: IN-SLA-TUNE1
Additional Path: Receive
Connect Timer: 10
Bidirectional Forwarding Detection:
Link Down Failover:
Shutdown Enable:
IPv4 Filtering:
Filter List In:
Filter List Out:
Distribute List In:
Distribute List Out:
Prefix List In:
Prefix List Out:
Route Map In:
Route Map Out: OUT-OF-SLA
Allow AS in:

Edit Route Map

Name: **OUT-OF-SLA**
Comments:
Rules:
+ Create New | Edit | Delete | Permit | Deny | Search...

ID	Action	Interface	Match IP Rules	Match Next Hop Rules
1	Permit			

Edit Route Map Rule
ID: 1
Action: Deny | **Permit**
IP Address Rule Variables:
Match interface:
Match IP address:
Match IPv6 address:
Match next hop router IP address:
Match next hop router IPv6 address:
Set next hop router IP address:
Set next hop router local IPv6 address:
AS Path Rule Variables:
Match AS path list: Set AS path: Action:
Community Rule Variables:
Match community:
Set community delete:
Set Community: 65001:1

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

The image shows a configuration interface for a BGP neighbor. On the left, the 'Edit Neighbor' panel is visible, with the following settings:

- IP: 10.50.1.1
- Remote AS: 65001
- Password: [REDACTED]
- Interface: HUB1-VPN1
- Update Source: [OFF]
- Graceful Restart Time: 0
- Advertisement Interval: 1
- Route Map Out Preferable: IN-SLA-TUN1
- Additional Path: Receive
- Connect Timer: 10
- Bidirectional Forwarding Detection: [OFF]
- Link Down Failover: [ON]
- Shutdown Enable: [OFF]
- IPv4 Filtering: [ON]
- Filter List In: [OFF]
- Filter List Out: [OFF]
- Distribute List In: [OFF]
- Distribute List Out: [OFF]
- Prefix List In: [OFF]
- Prefix List Out: [OFF]
- Route Map In: [OFF]
- Route Map Out: OUT-OF-SLA
- Allow AS In: [OFF]

On the right, the CLI configuration is shown, corresponding to the settings in the interface:

```
1 config router route-map
2   edit "IN-SLA-TUN1"
3     config rule
4       edit 1
5         set set-community "65001:1"
6       next
7     end
8   next
9   edit "OUT-OF-SLA"
10    config rule
11      edit 1
12        set set-community "65001:11"
13      next
14    end
15  next
16 end
17 config router bgp
18   config neighbor
19     edit "10.50.1.1"
20       set route-map-out "OUT-OF-SLA"
21       set route-map-out-preferable "IN-SLA-TUN1"
22     next
23   end
24 end
25
```

A dashed blue box highlights the 'Route Map Out' setting in the interface, which is set to 'OUT-OF-SLA'. A blue arrow points from this box to line 14 of the CLI code, which is the 'end' statement for the 'OUT-OF-SLA' rule.

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.

Create New SD-WAN Neighbor

IP: 10.50.1.1

Interface Member: HUB1-VPN1 (1 entry selected)

Performance SLA: HUB_HC

SLA: 1

Role: Standalone

Neighbor	Role	Interface Member	Performance SLA	SLA
10.50.1.1	Standalone	HUB1-VPN1	HUB_HC	1

```
1 config system sdwan
2   config neighbor
3     edit "10.50.1.1"
4       set health-check "HUB_HC"
5       set sla-id 1
6       set member 1
7     next
8   end
9 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.

Device Manager - Install Wizard

Provisioning Templates: HUB1_BGP_Recommended (Active), BRANCH1_BGP_Recommended, BRANCH1_BGP

Activate HUB1_BGP_Recommended

Template Name: HUB1_BGP

Enable ADVPN:

Local AS: 65001

Router ID: 10.100.100.100

Neighbor: Remote AS: 65001

Neighbor Group: Remote AS: 65001

Neighbor Range: Prefix: 10.50.1.0/24

Networks: Prefix: 10.10.0/24

Template Group Name	Provisioning Templates	Assigned to Device/Group
HUB1-TG	HUB1_IPsec, HUB1_BGP, HUB1-SDWAN, HUB-CLI	1 Device in Total: HUB1 [root]

```
1 config router bgp
2   set as 65001
3   set router-id 10.100.100.100
4   set ibgp-multipath enable
5   set additional-path enable
6   set graceful-restart enable
7   set additional-path-select 255
8   config neighbor-group
9     edit "VPN1"
10      set capability-graceful-restart enable
11      set link-down-failover enable
12      set next-hop-self enable
13      set soft-reconfiguration enable
14      set remote-as 65001
15      set additional-path send
16      set route-reflector-client enable
17    next
18  end
19  config neighbor-range
20    edit 1
21      set prefix 10.50.1.0 255.255.255.0
22      set neighbor-group "VPN1"
23    next
24  end
25  config network
26    edit 1
27      set prefix 10.10.100.0 255.255.255.0
28    next
29  end
30 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.

The image displays the configuration interface for a BGP Neighbor Group, a Route Map Rule, and a Community List, along with the corresponding CLI configuration.

Edit Neighbor Group: Name: VPN1, Remote AS: 65001, Interface: [Search], Activate IPv4: [On], Filter List In: [Off], Filter List Out: [Off], Distribute List In: [Off], Distribute List Out: [Off], Prefix List In: [Off], Prefix List Out: [Off], Route Map In: [On] (Route-Map-In).

Edit Route Map Rule: Match community: 65001:1, Match community exact: [Disable] [Enable], Set community delete: [Off], Set Community: [Off], Set extended community target: [Off], Set extended community site-of-origin: [Off], Other Rule Variables: Match metric, Match origin (Egp, Igp, Incomplete, None), Set origin (Egp, Igp, Incomplete, None), Match route type (External-type1, External-type2), Set metric type (External-type1, External-type2), Set metric, Set tag value: 1, Set route tag: 1.

Create New Community List: Name: 65001:1, Type: Expanded [Standard], Rules: [Create New] [Edit] [Delete] [Permit] [Deny], ID: 1, Action: [Deny] [Permit], Match: 65001:1.

CLI Configuration:

```
1 config router community-list
2   edit "65001:1"
3     config rule
4       edit 1
5         set action permit
6         set match "65001:1"
7       next
8     end
9   next
10 end
11 config router route-map
12   edit "Route-Map-In"
13     config rule
14       edit 1
15         set match-community "65001:1"
16         set set-route-tag "1"
17       next
18     end
19   next
20 end
21 config router bgp
22   config neighbor-group
23     edit "VPN1"
24       set route-map-in "Route-Map-In"
25     next
26   end
27 end
```

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.

Edit BGP Template

Name: BRANCH_BGP

Local As: 65001

Router ID: 10.100.100.\$(branch_id)

Neighbors:

IP	Remote AS
10.50.1.1	65001

Clone

Create New Route Map Rule

ID: 1

Action: Deny Permit

IP Address Rule Variables:

Match interface:

Match IP address:

Match IPv6 address:

Match next hop router IP address:

Match next hop router IPv6 address:

Set next hop router IP address:

Set next hop router local IPv6 address:

AS Path Rule Variables:

Match AS path list:

Set AS path:

Community Rule Variables:

Match community:

Set community delete:

Set Community: 65001:2

Edit Neighbor

IP: 10.50.2.1

Remote AS: 65001

Password:

Interface: HUB1-VPN2

Update Source:

Graceful Restart Time: 0

Advertisement Interval: 1

Route Map Out Preferabl: IN-SLA-TUN2

Additional Path: Receive

Connect Timer: 10

Bidirectional Forwarding Detection:

Link Down Failover:

Shutdown Enable:

IPv4 Filtering:

Filter List In:

Filter List Out:

Distribute List In:

Distribute List Out:

Prefix List In:

Prefix List Out:

Route Map In:

Route Map Out: OUT-OF-SLA

Create New Route Map

Name: IN-SLA-TUN2

Rules:

ID	Action	Interface	Match IP Rules
1	Permit		null

Neighbors:

IP	Remote AS
10.50.1.1	65001
10.50.2.1	65001

SDWAN Branch1 HUB1-VPN2

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

Performance SLA

Name	Detect Server	Packet Loss	Latency	Jitter	Failure Threshold	Recovery Threshold
HUB_HC	10.100.100.100	5	20	20	5	5

Neighbor

Neighbor	Role	Interface Member	Performance SLA	SLA
10.50.1.1	Standalone	HUB1-VPN1	HUB_HC	1
10.50.2.1	Standalone	HUB1-VPN2	HUB_HC	1

Branch1 BGP Neighbor2 Configuration Preview

Create New Route Map Rule		Edit Neighbor	
ID	1	IP	10.50.2.1
Action	Deny Permit	Remote AS	65001
IP Address Rule Variables		Password	*****
Match interface	<input checked="" type="checkbox"/>	Interface	HUB1-VPN2
Match IP address	<input checked="" type="checkbox"/>	Update Source	<input checked="" type="checkbox"/>
Match IPv6 address	<input checked="" type="checkbox"/>	Graceful Restart Time	0
Match next hop router IP address	<input checked="" type="checkbox"/>	Advertisement Interval	↓
Match next hop router IPv6 address	<input checked="" type="checkbox"/>	Route Map Out Preferable	IN-SLA-TUN2
Set next hop router IP address	<input type="text"/>	Additional Path	Receive
Set next hop router local IPv6 address	<input type="text"/>	Connect Timer	10
AS Path Rule Variables		Bidirectional Forwarding Detection	<input checked="" type="checkbox"/>
Match AS path list	<input checked="" type="checkbox"/>	Link Down Failover	<input checked="" type="checkbox"/>
Set AS path	<input type="text"/>	Shutdown Enable	<input checked="" type="checkbox"/>
Community Rule Variables		IPv4 Filtering	<input checked="" type="checkbox"/>
Match community	<input checked="" type="checkbox"/>	Filter List In	<input checked="" type="checkbox"/>
Set community delete	<input checked="" type="checkbox"/>	Filter List Out	<input checked="" type="checkbox"/>
Set Community	65001:2	Distribute List In	<input checked="" type="checkbox"/>
Create New Route Map		Distribute List Out	<input checked="" type="checkbox"/>
Name	IN-SLA-TUN2	Prefix List In	<input checked="" type="checkbox"/>
Comments		Prefix List Out	<input checked="" type="checkbox"/>
		Route Map In	<input checked="" type="checkbox"/>
		Route Map Out	OUT-OF-SLA

```

1 config router route-map
2   edit "IN-SLA-TUN2"
3     config rule
4       edit 1
5         set set-community "65001:2"
6       next
7     end
8   next
9 end
10 config router bgp
11   config neighbor
12     edit "10.50.2.1"
13       set advertisement-interval 1
14       set capability-graceful-restart enable
15       set link-down-failover enable
16       set soft-reconfiguration enable
17       set description "HUB1-VPN1"
18       set interface "HUB1-VPN2"
19       set remote-as 65001
20       set route-map-out "OUT-OF-SLA"
21       set connect-timer 10
22       set additional-path receive
23       set route-map-out-preferable "IN-SLA-TUN2"
24     next
25   end
26 end
27 config system sdwan
28   config neighbor
29     edit "10.50.2.1"
30       set health-check "HUB_HC"
31       set sla-id 1
32       set member 2
33     next
34   end
35 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.

Edit BGP Template

+ Create New Edit Delete More ▾

<input checked="" type="checkbox"/>	Name ▾	Clone	Remote AS ▾
<input checked="" type="checkbox"/>	VPN1		65001

Edit Neighbor Group

Name:

Remote AS:

Interface:

Activate IPv4:

Neighbor Ranges

+ Create New Edit Delete More ▾ Search...

<input checked="" type="checkbox"/>	Prefix ▾	Neighbor Group ▾	Maximum Neighbor Number ▾
<input checked="" type="checkbox"/>	10.50.1.0/255.255.255.0	VPN1	0

Edit Neighbor Range

Prefix:

Neighbor Group:

Max neighbor number:

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.

Create New Community List

Name: 65001:2
Type: Expanded Standard

Rules

ID	Action	Regular Expression
1	Permit	

Edit Route Map Rule

Match community: 65001:2

Match community exact: Distort Invert

Set community delete:

Set Community:

Set extended community target:

Set extended community site-of-origin:

Other Rule Variables

Match metric:

Match origin:

Set origin:

Match route type:

Set metric type:

Set metric:

Set tag value:

Set route tag: 2

Community list rule edit

ID: 1
Action: Deny Permit
Match: 65001:2

Advanced Options ->

Edit Neighbor Group

Name: VPN2

Remote AS: 65001

Interface:

Activate IPv4:

Filter List In:

Filter List Out:

Distribute List In:

Distribute List Out:

Prefix List In:

Prefix List Out:

Route Map In: Route-Map-In

Route Map Out:

Allow AS In:

Graceful Restart Time: 0

Max Prefix:

HUB1 VPN2 Configuration Preview.

```

1 config router community-list
2   edit "65001:11"
3     config rule
4       edit 1
5         set action permit
6         set match "65001:11"
7       next
8     end
9   next
10  edit "65001:2"
11    config rule
12      edit 1
13        set action permit
14        set match "65001:2"
15      next
16    end
17  next
18 end
19 config router route-map
20   edit "Route-Map-In"
21   config rule

```

```

22     edit 2
23       set match-community "65001:2"
24       set set-route-tag "2"
25     next
26     edit 11
27       set match-community "65001:11"
28       set set-route-tag "11"
29     next
30   end
31 next
32 end
33 config router bgp
34   config neighbor-group
35     edit "VPN2"
36       set capability-graceful-restart enable
37       set link-down-failover enable
38       set next-hop-self enable
39       set soft-reconfiguration enable
40       set remote-as 65001
41       set route-map-in "Route-Map-In"
42       set additional-path send

```

```

43       set route-reflector-client enable
44     next
45   end
46   config neighbor-range
47     edit 2
48       set prefix 10.50.2.0 255.255.255.0
49       set neighbor-group "VPN2"
50   next
51 end
52 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

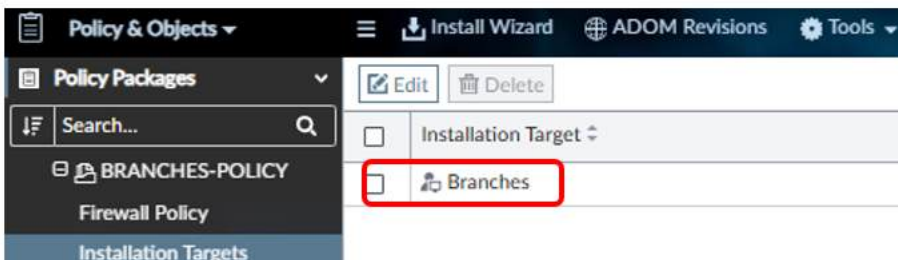
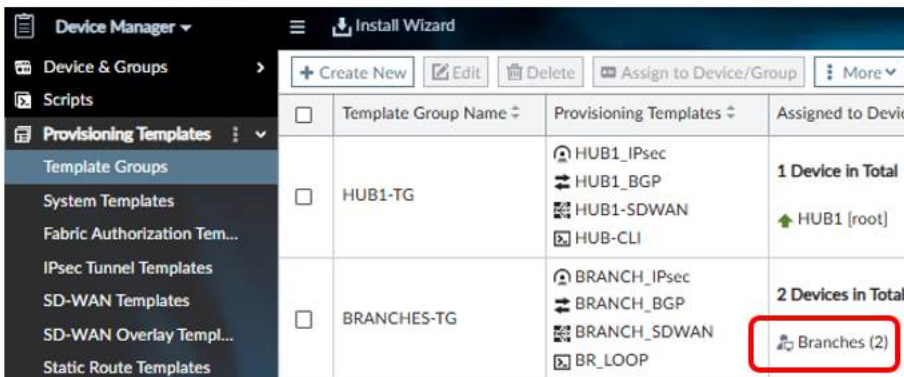
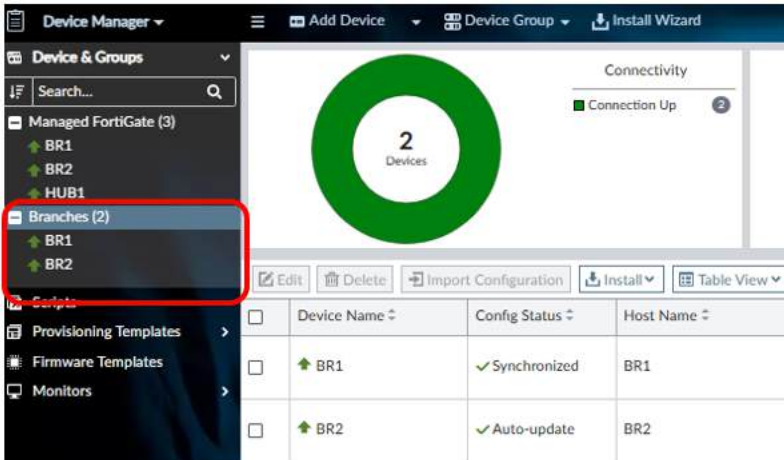
Original VRF 0
Local
 10.50.1.1 from 10.50.1.1 (10.100.100.100)
  Origin IGP metric 0, localpref 100, valid, internal, best
  Receive Path ID: 1
  Advertised Path ID: 1
  Last update: Wed Oct 30 12:57:26 2024
```

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.

```
1 config system sdwan
2   config health-check
3     delete "Default_Office_365"
4     delete "Default_google Search"
5     delete "Default_gmail"
6     delete "Default_FortiGuard"
7     delete "Default_DNS"
8     delete "Default_AWS"
9   end
10 end
11 config vpn ipsec phase1-interface
12   edit "HUB1-VPN1"
13     set interface "port1"
14     set ike-version 2
15     set comments "VPN: HUB1-VPN1 [Created by IPSEC Template]"
16     set proposal aes256-sha256
17     set peertype any
18     set mode-cfg enable
19     set localid "Mombasa_Tun1"
20     set remote-gw 100.1.100.1
21     set idle-timeout enable
22     set net-device enable
23     set add-route disable
24     set auto-discovery-receiver enable
25     set psksecret ENC Uf9K7GT61jLI6XwvrngnX2qW326yfJkLIgWRsz/6HS
26     set network-overlay enable
27     set network-id 1
28     set auto-discovery-shortcuts dependent
29   next
30 end
31 config system interface
32   edit "HUB1-VPN1"
33     set vdom "root"
34     set type tunnel
35     set snmp-index 116
36     set interface "port1"
37   next
38 end
39 config vpn ipsec phase1-interface
40   edit "HUB1-VPN2"
41     set interface "port2"
42     set ike-version 2
```

```
43   set comments "VPN: HUB1-VPN2 [Created by IPSEC Template]"
44   set proposal aes256-sha256
45   set peertype any
46   set mode-cfg enable
47   set localid "Mombasa_Tun2"
48   set remote-gw 200.2.100.1
49   set idle-timeout enable
50   set net-device enable
51   set add-route disable
52   set auto-discovery-receiver enable
53   set psksecret ENC Uf9K7GT61jLI6XwvrngnX2qW326yfJkLIgWRsz/6I
54   set network-overlay enable
55   set network-id 2
56   set auto-discovery-shortcuts dependent
57   next
58 end
59 config system interface
60   edit "HUB1-VPN2"
61     set vdom "root"
62     set type tunnel
63     set snmp-index 117
64     set interface "port2"
65   next
66   edit "HUB_LOOP"
67     set vdom "root"
68     set ip 10.100.100.2 255.255.255.255
69     set allowaccess ping
70     set type loopback
71     set snmp-index 118
72   next
73 end
74 config vpn ipsec phase2-interface
75   edit "HUB1-VPN1"
76     set phaseiname "HUB1-VPN1"
77     set proposal aes256-sha256
78     set auto-negotiate enable
79     set comments "VPN: HUB1-VPN1 [Created by IPSEC Template]"
80   next
81   edit "HUB1-VPN2"
82     set phaseiname "HUB1-VPN2"
83     set proposal aes256-sha256
84     set auto-negotiate enable
```

```
85   set comments "VPN: HUB1-VPN2 [Created by IPSEC Template]"
86   next
87 end
88 config system sdwan
89   set status enable
90   config zone
91     edit "overlay"
92     next
93     edit "DIA"
94     next
95   end
96   config members
97     edit 1
98       set interface "HUB1-VPN1"
99       set zone "overlay"
100   next
101   edit 2
102     set interface "HUB1-VPN2"
103     set zone "overlay"
104   next
105   edit 3
106     set interface "port1"
107     set zone "DIA"
108     set gateway 100.1.2.2
109   next
110   edit 4
111     set interface "port2"
112     set zone "DIA"
113     set gateway 200.2.2.2
114   next
115 end
116 config health-check
117   edit "HUB_HC"
118     set server 10.100.100.100
119     set members 2 1
120     config sla
121       edit 1
122         set latency-threshold 20
123         set jitter-threshold 20
124         set packetloss-threshold 5
125     next
```

```

126     end
127     next
128   end
129 end
130 config router route-map
131   edit "OUT-OF-SLA"
132     config rule
133       edit 1
134         set set-community "65001:11"
135       next
136     end
137   next
138   edit "IN-SLA-TUN1"
139     config rule
140       edit 1
141         set set-community "65001:1"
142       next
143     end
144   next
145 end
146 config router bgp
147   config neighbor
148     edit "10.50.1.1"
149       set advertisement-interval 1
150       set capability-graceful-restart enable
151       set link-down-failover enable
152       set soft-reconfiguration enable
153       set description "HUB1-VPN1"
154       set interface "HUB1-VPN1"
155       set remote-as 65001
156       set route-map-out "OUT-OF-SLA"
157       set connect-timer 10
158       set additional-path receive
159       set route-map-out-preferable "IN-SLA-TUN1"
160     next
161   end
162 end
163 config system sdwan
164   config neighbor
165     edit "10.50.1.1"
166       set health-check "HUB_HC"
167     set sla-id 1

```

```

167     set sla-id 1
168     set member 1
169     next
170   end
171 end
172 config router route-map
173   edit "IN-SLA-TUN2"
174     config rule
175       edit 1
176         set set-community "65001:2"
177       next
178     end
179   next
180 end
181 config router bgp
182   config neighbor
183     edit "10.50.2.1"
184       set advertisement-interval 1
185       set capability-graceful-restart enable
186       set link-down-failover enable
187       set soft-reconfiguration enable
188       set description "HUB1-VPN1"
189       set interface "HUB1-VPN2"
190       set remote-as 65001
191       set route-map-out "OUT-OF-SLA"
192       set connect-timer 10
193       set additional-path receive
194       set route-map-out-preferable "IN-SLA-TUN2"
195     next
196   end
197 end
198 config system sdwan
199   config neighbor
200     edit "10.50.2.1"
201       set health-check "HUB_HC"
202     set sla-id 1
203     set member 2
204   next
205 end
206 end
207 config router bgp

```

```

208   set as 65001
209   set router-id 10.100.100.2
210   set ibgp-multipath enable
211   set additional-path enable
212   set graceful-restart enable
213   set additional-path-select 255
214   config network
215     edit 1
216       set prefix 10.10.2.0 255.255.255.0
217     next
218   end
219   set recursive-next-hop enable
220 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

```

```

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>
*>i 10.10.2.0/24 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>
*>i 10.10.100.0/24 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 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 50 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

```

```

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
*>i 10.10.1.0/24 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>
*>i 10.10.100.0/24 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    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    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 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/PfxRcv
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

Name	Remote Gat...	Peer ID	Incoming Data	Outgoing Data	Phase 1	Phase 2 Selectors
VPN1_0	100.1.1.1	Nakuru_Tun1	315.70 kB	317.89 kB	VPN1_0	VPN1
VPN1_1	100.1.2.1	Mombasa_Tun1	314.69 kB	317.16 kB	VPN1_1	VPN1
VPN2_0	200.2.1.1	Nakuru_Tun2	316.65 kB	319.27 kB	VPN2_0	VPN2
VPN2_1	200.2.2.1	Mombasa_Tun2	315.18 kB	316.99 kB	VPN2_1	VPN2

```
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.

The screenshot displays the Fortinet SDWAN configuration interface. On the left, a sidebar shows a list of SDWAN templates: 'BRANCH_SDWAN' and 'HUB1-SDWAN'. The main area is titled 'Edit SD-WAN Template' and shows a list of links with columns for ID, Name, Source, Destination, Criteria, Members, Status, and Performance SLA. The links are: 'virtual-wan-link', 'DIA', '1' (port1, 100.1.100.2), '2' (port2, 200.2.100.2), 'overlay', and '3' (VPN1, 0.0.0.0). Below this is the 'SD-WAN Rules' section, which contains a table of rules. The 'Destination' column for rules 1, 2, and 3 is highlighted with a red box, showing 'Route Tag: 1', 'Route Tag: 2', and 'Route Tag: 11' respectively. The 'sd-wan' rule has a destination of 'ALL'. To the right, the CLI configuration for the SDWAN system is shown, including service definitions for TUN1, TUN2, and TUN3.

ID	Name	Source	Destination	Criteria	Members	Status	Performance SLA
1	TUN1	Data-Networks	Route Tag: 1		VPN1	Enable	
2	TUN2	Data-Networks	Route Tag: 2		VPN2	Enable	
3	TUN3	Data-Networks	Route Tag: 11		VPN1, VPN2	Enable	
	sd-wan	ALL	ALL	Source IP	ALL		

```
1 config system sdwan
2   config service
3     edit 1
4       set name "TUN1"
5       set route-tag 1
6       set src "Data-Networks"
7       set priority-members 3
8     next
9     edit 2
10      set name "TUN2"
11      set route-tag 2
12      set src "Data-Networks"
13      set priority-members 4
14    next
15    edit 3
16      set name "TUN3"
17      set route-tag 11
18      set src "Data-Networks"
19      set priority-members 3 4
20    next
21  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.

FortiGate BR1 IPsec Monitor screenshot. A red banner at the top indicates "FortiGate time is out of sync." The IPsec table shows three tunnels. The first tunnel, HUB1-VPN2_0, is highlighted with a red box. It has a remote address of 200.2.2.1, peer ID of Mombasa_Tun2, and shows incoming and outgoing data. The Phase 2 selectors are HUB1-VPN2_0 and HUB1-VPN2.

Name	Remote...	Peer ID	Incoming Data	Outgoing...	Phase 1	Phase 2 Selectors
HUB1-VPN2_0	200.2.2.1	Mombasa_Tun2	960 B	756 B	HUB1-VPN2_0	HUB1-VPN2 HUB1-VPN2
HUB1-VPN1	100.1.100.1	Nairobi_Tun1	422.32 kB	421.80 kB	HUB1-VPN1	HUB1-VPN1
HUB1-VPN2	200.2.100.1	Nairobi_Tun2	424.23 kB	425.24 kB	HUB1-VPN2	HUB1-VPN2

FortiGate BR2 IPsec Monitor screenshot. A red banner at the top indicates "FortiGate time is out of sync." The IPsec table shows three tunnels. The first tunnel, HUB1-VPN2_0, is highlighted with a red box. It has a remote address of 200.2.1.1, peer ID of Nakuru_Tun2, and shows incoming and outgoing data. The Phase 2 selectors are HUB1-VPN2_0 and HUB1-VPN2.

Name	Remot...	Peer ID	Incoming Data	Outgoing Data	Phase 1	Phase 2 Selectors
HUB1-VPN1	100.1.100.1	Nairobi_Tun1	432.44 kB	431.84 kB	HUB1-VPN1	HUB1-VPN1
HUB1-VPN2	200.2.100.1	Nairobi_Tun2	432.86 kB	432.43 kB	HUB1-VPN2	HUB1-VPN2
HUB1-VPN2_0	200.2.1.1	Nakuru_Tun2	756 B	960 B	HUB1-VP...	HUB1-VPN2 HUB1-VPN2

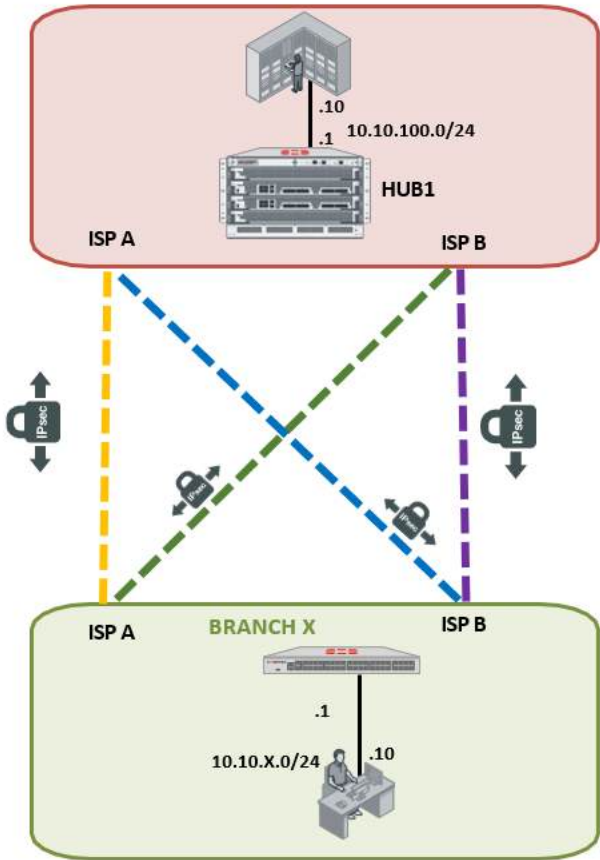
```
BR1 # diagnose sys sdwan health-check
Health Check(HUB_HC):
Seq(2 HUB1-VPN2): state(alive), packet-loss(0.000%) latency(4.618), jitter(2.992), mos(4.399), bandwidth-up(999999), ba
ndwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(2 HUB1-VPN2_0): state(dead), packet-loss(100.000%) sla_map=0x0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



TUNNEL	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.

The image shows two screenshots from the Fortinet SD-WAN configuration interface. The left screenshot displays the 'IPsec Template - HUB1_IPsec' configuration page. It includes a 'Name' field with 'HUB1_IPsec' and a 'Description' field. Below is a table of existing VPN tunnels with a 'Clone' button highlighted. The right screenshot shows the 'Clone IPsec Tunnel - Clone_VPN1' configuration page. It includes fields for Tunnel Name (VPN3), Remote Device (IP Address), Outgoing Interface (port1), Local ID (\$local_id_Tun3), Network ID (3), IPv4 Start IP (10.50.3.10), IPv4 End IP (10.50.3.250), IPv4 Netmask (255.255.255.0), Proposal (aes256-sha256), Authentication Method (Pre-shared Key), and Tunnel Interface Setup (IP: 10.50.3.1/32, Remote IP: 10.50.3.253/24). A Phase2 Interface table at the bottom shows a new entry for VPN3.

TUNNEL	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)

Name	Keep Alive	Key Life
VPN3	<input type="checkbox"/>	43200

HUB1 IPSEC Provisioning for VPN4

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

Clone IPsec Tunnel - Clone_VPN2

Tunnel Name: VPN4

Network

Routing: Manual Automatic

Remote Device: IP Address Dynamic DNS Dynamic

Outgoing Interface: port2

Local ID: \$(local_id)_Tun4

Network Overlay:

Network ID: 4

IPv4 Start IP: 10.50.4.10

IPv4 End IP: 10.50.4.250

IPv4 Netmask: 255.255.255.0

Proposal: aes256-sha256 (1 entry selected)

FEC Health Check

Authentication

Authentication Method: Pre-shared Key Signature

Pre-shared Key:

Tunnel Interface Setup

IP: 10.50.4.1/32

Remote IP: 10.50.4.253/24

Phase2 Interface

Name	Keep Alive	Key Life	Proposal
<input type="checkbox"/> VPN4	<input checked="" type="checkbox"/>	43200	aes256-sha256

IPsec Template - HUB1_IPsec

Name: HUB1_IPsec

Description:

Name	Type	Outgoing Interface
<input type="checkbox"/> VPN1	Dynamic	port1
<input type="checkbox"/> VPN2	Dynamic	port2
<input type="checkbox"/> VPN3	Dynamic	port1
<input type="checkbox"/> VPN4	Dynamic	port2

TUNNEL	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

```
1 config vpn ipsec phase1-interface
2   edit "VPN3"
3     set type dynamic
4     set interface "port1"
5     set ike-version 2
6     set dpd on-idle
7     set comments "VPN: VPN3 [Created by IPSEC Template]"
8     set proposal aes256-sha256
9     set peertype any
10    set mode-cfg enable
11    set localid "Nairobi_Tun3"
12    set dpd-retryinterval 60
13    set net-device disable
14    set add-route disable
15    set auto-discovery-sender enable
16    set ipv4-start-ip 10.50.3.10
17    set ipv4-end-ip 10.50.3.250
18    set ipv4-netmask 255.255.255.0
19    set psksecret ENC Z8Zpc/bwU2j1HxCFsp0zLVsmPXEWQvc6JVoYq8gt:
20    set network-overlay enable
21    set network-id 3
22  next
23 end
24 config system interface
25   edit "VPN3"
26     set vdom "root"
27     set ip 10.50.3.1 255.255.255.255
28     set type tunnel
29     set remote-ip 10.50.3.253 255.255.255.0
30     set snmp-index 119
31     set interface "port1"
32  next
33 end
34 config vpn ipsec phase1-interface
35   edit "VPN4"
36     set type dynamic
37     set interface "port2"
38     set ike-version 2
```

```
39     set dpd on-idle
40     set comments "VPN: VPN4 [Created by IPSEC Template]"
41     set proposal aes256-sha256
42     set peertype any
43     set mode-cfg enable
44     set localid "Nairobi_Tun4"
45     set dpd-retryinterval 60
46     set net-device disable
47     set add-route disable
48     set auto-discovery-sender enable
49     set ipv4-start-ip 10.50.4.10
50     set ipv4-end-ip 10.50.4.250
51     set ipv4-netmask 255.255.255.0
52     set psksecret ENC Z8Zpc/bwU2j1HxCFsp0zLVsmPXEWQvc6JVoYq8gt3RKcH0GzUF
53     set network-overlay enable
54     set network-id 4
55  next
56 end
57 config system interface
58   edit "VPN4"
59     set vdom "root"
60     set ip 10.50.4.1 255.255.255.255
61     set type tunnel
62     set remote-ip 10.50.4.253 255.255.255.0
63     set snmp-index 120
64     set interface "port2"
65  next
66 end
67 config vpn ipsec phase2-interface
68   edit "VPN3"
69     set phaseiname "VPN3"
70     set proposal aes256-sha256
71     set comments "VPN: VPN3 [Created by IPSEC Template]"
72  next
73   edit "VPN4"
74     set phaseiname "VPN4"
75     set proposal aes256-sha256
76     set comments "VPN: VPN4 [Created by IPSEC Template]"
```

```
77 next
78 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.

IPsec Template - BRANCH_IPsec

Name: BRANCH_IPsec

Description:

	Name	Type	Outgoing Interface
<input type="checkbox"/>	HUB1-VPN1	Static	port1
<input checked="" type="checkbox"/>	HUB1-VPN2	Static	port2

Clone

TUNNEL	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)

Clone IPsec Tunnel - Clone_HUB1-VPN2

Tunnel Name: HUB1-VPN3

Network:

Routing: Manual Automatic

Remote Device: IP Address Dynamic DNS Dynamic

Remote Gateway (IP Address): \$(hub1_ISPA)

Outgoing Interface: port2

Local ID: \$(local_id)_Tun3

Network Overlay:

Network ID: 3

Proposal: aes256-sha256

FEC Health Check:

Authentication: Pre-shared Key Signature

Pre-shared Key:

Tunnel Interface Setup:

IP:

Remote IP:

Phase2 Interface:

	Name	Keep Alive
<input checked="" type="checkbox"/>	HUB1-VPN3	<input type="checkbox"/>

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.

IPsec Template - BRANCH_IPsec

Name: BRANCH_IPsec

Description:

Actions: + Create New, Edit, Delete, More

Name	Type	Outgoing Interface	Local Interface
HUB1-VPN1	Static	port1	
HUB1-VPN2	Static	port2	
HUB1-VPN3	Static	port2	
HUB1-VPN4	Static	port1	

Clone IPsec Tunnel - Clone_HUB1-VPN1

Tunnel Name: HUB1-VPN4

Network:

Routing: Manual Automatic

Remote Device: IP Address Dynamic DNS Dynamic

Remote Gateway (IP Address): \$(hub1_ISPB)

Outgoing Interface: port1

Local ID: \$(local_id)_Tun4

Network Overlay:

Network ID: 4

Proposal: aes256-sha256

FEC Health Check:

Authentication: Pre-shared Key Signature

Pre-shared Key:

Tunnel Interface Setup:

IP: Remote IP:

Phase2 Interface:

Name	Keep Alive
HUB1-VPN4	<input type="checkbox"/>

TUNNEL	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.

```
1 config vpn ipsec phase1-interface
2   edit "HUB1-VPN3"
3     set interface "port2"
4     set ike-version 2
5     set comments "VPN: HUB1-VPN3 [Created by IPSEC Template]"
6     set proposal aes256-sha256
7     set peertype any
8     set mode-cfg enable
9     set localid "Nakuru_Tun3"
10    set remote-gw 100.1.100.1
11    set idle-timeout enable
12    set net-device enable
13    set add-route disable
14    set auto-discovery-receiver enable
15    set psksecret ENC Z8Zpc/bwU2j1HxCFsp0zLVsmpXEMQQvc6JVoYq8gt3R
16    set network-overlay enable
17    set network-id 3
18    set auto-discovery-shortcuts dependent
19  next
20 end
21 config system interface
22   edit "HUB1-VPN3"
23     set vdom "root"
24     set type tunnel
25     set snmp-index 119
26     set interface "port2"
27  next
28 end
29 config vpn ipsec phase1-interface
30   edit "HUB1-VPN4"
31     set interface "port1"
32     set ike-version 2
33     set comments "VPN: HUB1-VPN4 [Created by IPSEC Template]"
34     set proposal aes256-sha256
35     set peertype any
36     set mode-cfg enable
37     set localid "Nakuru_Tun4"
38     set remote-gw 200.2.100.1
39     set idle-timeout enable
40     set net-device enable
41     set add-route disable
42     set auto-discovery-receiver enable
43     set psksecret ENC Z8Zpc/bwU2j1HxCFsp0zLVsmpXEMQQvc6JVoYq8g
44     set network-overlay enable
45     set network-id 4
46
47     set auto-discovery-shortcuts dependent
48   next
49 end
50 config system interface
51   edit "HUB1-VPN4"
52     set vdom "root"
53     set type tunnel
54     set snmp-index 120
55     set interface "port1"
56  next
57 end
58 config vpn ipsec phase2-interface
59   edit "HUB1-VPN3"
60     set phaseName "HUB1-VPN3"
61     set proposal aes256-sha256
62     set auto-negotiate enable
63     set comments "VPN: HUB1-VPN3 [Created by IPSEC Template]"
64  next
65   edit "HUB1-VPN4"
66     set phaseName "HUB1-VPN4"
67     set proposal aes256-sha256
68     set auto-negotiate enable
69     set comments "VPN: HUB1-VPN4 [Created by IPSEC Template]"
70  next
71 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.

The image shows two screenshots of the FortiGate IPsec monitoring interface. The top screenshot is for BR1, and the bottom is for BR2. Both show a table of IPsec tunnels with columns for Name, Remote Gateway, Peer ID, Incoming Data, Outgoing Data, Phase 1, and Phase 2 Selectors. In BR1, HUB1-VPN3 and HUB1-VPN4 are marked with red status icons, while HUB1-VPN1 and HUB1-VPN2 are green. In BR2, HUB1-VPN1 and HUB1-VPN2 are green, while HUB1-VPN3 and HUB1-VPN4 are red. A red box highlights the red status icons for HUB1-VPN3 and HUB1-VPN4 in both dashboards.

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

The image shows the FortiGate SD-WAN configuration interface. On the left, the 'SD-WAN Zones' table is displayed with columns for ID, Interface, Gateway, Cost, Priority, and Status. The 'overlay' zone is highlighted with a red box. To the right, a CLI configuration snippet is shown, including the following commands:

```

1 config system sdwan
2   config members
3     edit 5
4       set interface "VPN3"
5       set zone "overlay"
6     next
7     edit 6
8       set interface "VPN4"
9       set zone "overlay"
10    next
11  end
12 end

```

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

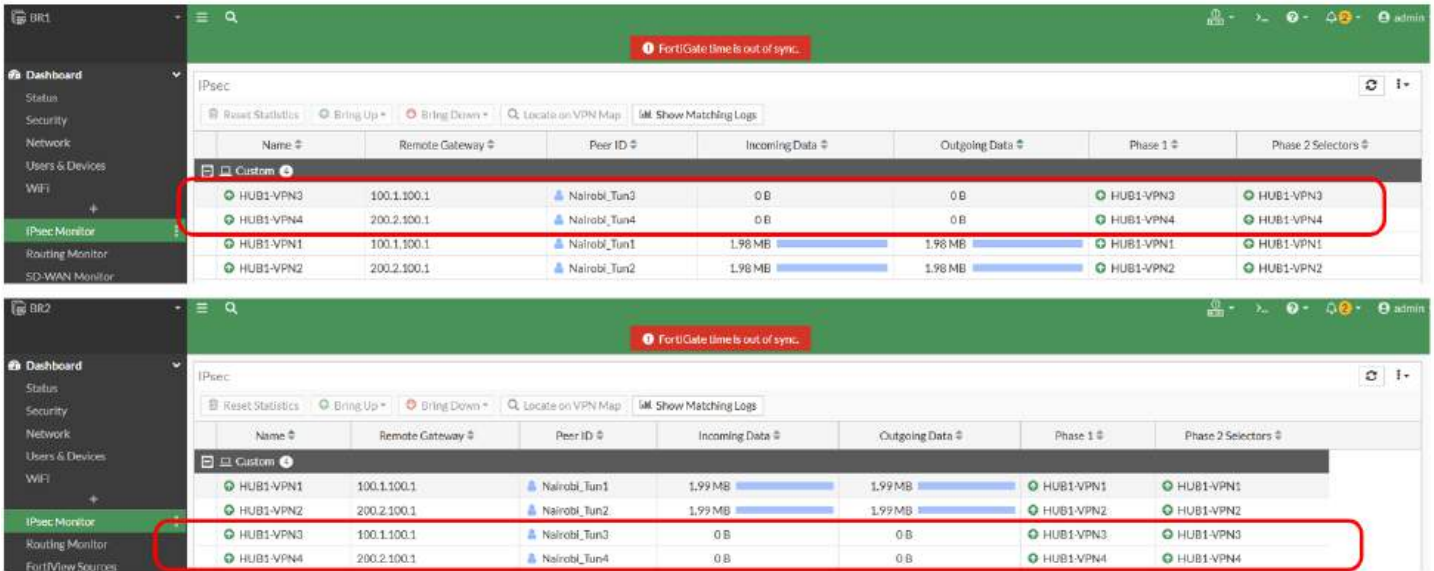
The image shows the FortiGate SD-WAN configuration interface. On the left, the 'SD-WAN Zones' table is displayed with columns for ID, Interface, Gateway, Cost, Priority, and Status. The 'HUB1-VPN3' and 'HUB1-VPN4' zones are highlighted with a red box. To the right, a CLI configuration snippet is shown, including the following commands:

```

1 config system sdwan
2   config members
3     edit 5
4       set interface "HUB1-VPN3"
5       set zone "overlay"
6     next
7     edit 6
8       set interface "HUB1-VPN4"
9       set zone "overlay"
10    next
11  end
12 end

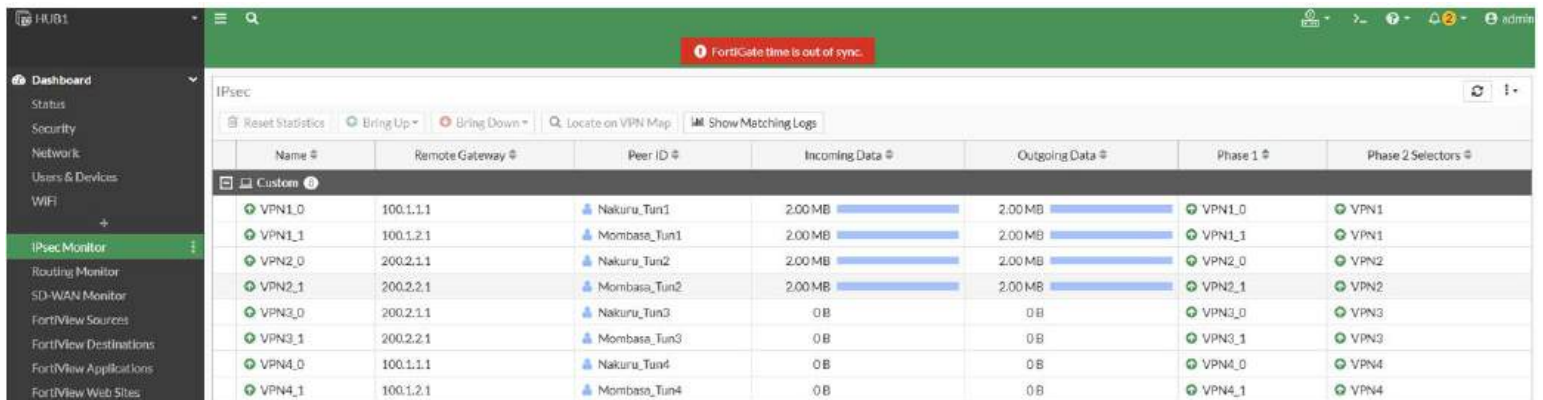
```

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

Community list rule edit

ID: 1
Action: Deny Permit
Match: 65001:3
Advanced Options >

Create New Community List

Name: 65001:3
Type: Expanded Standard

Rules

ID	Action	Regular Expression
1	Permit	

Neighbor Group

Name	Remote AS
VPN1	65001
VPN2	65001
VPN3	65001
VPN4	65001

Create New Route Map Rule

Community Rule Variables

Match community: 65001:3
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: Egp Igp Incomplete None
Match route type: External-type1 External-type2 None
Set metric type: External-type1 External-type2 None
Set metric:
Set tag value:
Set route tag: 3

Edit Route Map

Name: Route-Map-In
Comments:

Rules

ID	Action	Interface
1	Permit	
2	Permit	
3	Permit	
11	Permit	

Neighbor Group

+ Create New Edit Delete More Search...

<input type="checkbox"/>	Name ⇅	Remote AS ⇅	⚙
<input checked="" type="checkbox"/>	VPN1	65001	
<input type="checkbox"/>	VPN2	65001	
<input type="checkbox"/>	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...

<input type="checkbox"/>	ID ⇅	Action ⇅	Regular Expression ⇅	⚙
<input type="checkbox"/>	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: Egp Igp Incomplete None

Match route type: External-type1 External-type2 None

Set metric type: External-type1 External-type2 None

Set metric:

Set tag value:

Set route tag: 4

Edit Route Map

Name: Route-Map-In

Comments:

Rules

+ Create New Edit Delete Permit Deny Search...

<input type="checkbox"/>	ID ⇅	Action ⇅	Interface ⇅
<input checked="" type="checkbox"/>	1	Permit	
<input type="checkbox"/>	2	Permit	
<input type="checkbox"/>	3	Permit	
<input type="checkbox"/>	4	Permit	
<input type="checkbox"/>	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.

Neighbor Ranges

	Prefix ⇅	Group ⇅	Maximum Neighbor Number ⇅	
<input checked="" type="checkbox"/>	10.50.1.0/255.255.255.0	VPN1	0	

Edit Neighbor Range

Prefix: 10.50.3.0/255.255.255.0

Neighbor Group: VPN3

Max neighbor number: 0

Edit Neighbor Range

Prefix: 10.50.4.0/255.255.255.0

Neighbor Group: VPN4

Max neighbor number: 0

Neighbor Ranges

	Prefix ⇅	Neighbor Group ⇅	Maximum Neighbor Number ⇅	
<input type="checkbox"/>	10.50.1.0/255.255.255.0	VPN1	0	
<input type="checkbox"/>	10.50.2.0/255.255.255.0	VPN2	0	
<input type="checkbox"/>	10.50.3.0/255.255.255.0	VPN3	0	
<input type="checkbox"/>	10.50.4.0/255.255.255.0	VPN4	0	

HUB1 BGP Provisioning Preview for VPN3 & VPN4

```
1 config router community-list
2   edit "65001:3"
3     config rule
4       edit 1
5         set action permit
6         set match "65001:3"
7       next
8     end
9   next
10  edit "65001:4"
11    config rule
12      edit 1
13        set action permit
14        set match "65001:4"
15      next
16    end
17  next
18 end
19 config router route-map
20   edit "Route-Map-In"
21     config rule
22       edit 3
23         set match-community "65001:3"
24         set set-route-tag "3"
25       next
26       edit 4
27         set match-community "65001:4"
28         set set-route-tag "4"
29       next
30     end
31   next
32 end
33 config router bgp
34   config neighbor-group
35     edit "VPN3"
36       set capability-graceful-restart enable
37       set link-down-fallover enable
38       set next-hop-self enable
39       set soft-reconfiguration enable
40       set remote-as 65001
41       set route-map-in "Route-Map-In"
42       set additional-path send
43       set route-reflector-client enable
44     next
```

```
45     edit "VPN4"
46       set capability-graceful-restart enable
47       set link-down-fallover enable
48       set next-hop-self enable
49       set soft-reconfiguration enable
50       set remote-as 65001
51       set route-map-in "Route-Map-In"
52       set additional-path send
53       set route-reflector-client enable
54     next
55   end
56   config neighbor-range
57     edit 3
58       set prefix 10.50.3.0 255.255.255.0
59       set neighbor-group "VPN3"
60     next
61     edit 4
62       set prefix 10.50.4.0 255.255.255.0
63       set neighbor-group "VPN4"
64     next
65   end
66 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.

The image shows three screenshots from the Fortinet SD-WAN configuration interface:

- Neighbors:** A table showing existing neighbors. The 'Clone' button is highlighted in red. The table contains:

IP	Remote AS
10.50.1.1	65001
10.50.2.1	65001
- Create New Route Map:** A form where the name 'IN-SLA-TUN3' is entered and highlighted in red.
- Create New Route Map Rule:** A form where the ID is '1', Action is 'Deny/Permit', and 'Set Community' is set to '65001:3' (highlighted in red).
- Edit Neighbor:** A form where the IP is '10.50.3.1', Remote AS is '65001', Interface is 'HUB1-VPN3', Route Map Out Preferable is 'IN-SLA-TUN3', and Route Map Out is 'OUT-OF-SLA' (all highlighted in red).

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.

The image shows three screenshots from the Fortinet SD-WAN configuration interface:

- Create New Route Map:** A form where the name 'IN-SLA-TUN4' is entered and highlighted in red.
- Create New Route Map Rule:** A form where the ID is '1', Action is 'Deny/Permit', and 'Set Community' is set to '65001:4' (highlighted in red).
- Edit Neighbor:** A form where the IP is '10.50.4.1', Remote AS is '65001', Interface is 'HUB1-VPN4', Route Map Out Preferable is 'IN-SLA-TUN4', and Route Map Out is 'OUT-OF-SLA' (all highlighted in red).

Below the configuration screenshots is a screenshot of the 'Neighbors' table, which now includes the newly configured neighbor:

IP	Remote AS
10.50.1.1	65001
10.50.2.1	65001
10.50.3.1	65001
10.50.4.1	65001

BRANCH PROVISIONING Preview FOR NEIGHBORS 3 & 4

```
1 config router route-map
2   edit "IN-SLA-TUN3"
3     config rule
4       edit 1
5         set set-community "65001:3"
6       next
7     end
8   next
9   edit "IN-SLA-TUN4"
10    config rule
11      edit 1
12        set set-community "65001:4"
13      next
14    end
15  next
16 end
17 config router bgp
18   config neighbor
19     edit "10.50.3.1"
20       set advertisement-interval 1
21       set capability-graceful-restart enable
22       set link-down-failover enable
23       set soft-reconfiguration enable
24       set description "HUB1-VPN1"
25       set interface "HUB1-VPN3"
26       set remote-as 65001
27       set route-map-out "OUT-OF-SLA"
28       set connect-timer 10
29       set additional-path receive
30       set route-map-out-preferable "IN-SLA-TUN3"
31     next
32     edit "10.50.4.1"
33       set advertisement-interval 1
34       set capability-graceful-restart enable
35       set link-down-failover enable
36       set soft-reconfiguration enable
37       set description "HUB1-VPN1"
38       set interface "HUB1-VPN4"
39       set remote-as 65001
40       set route-map-out "OUT-OF-SLA"
41       set connect-timer 10
42       set additional-path receive
43       set route-map-out-preferable "IN-SLA-TUN4"
44     next
45   end
46 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 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.

Create New SD-WAN Neighbor

IP: 10.50.3.1

Interface Member: HUB1-VPN3

Performance SLA: HUB_HC

SLA: 1

Role: Standalone

Advanced Options >

Edit SD-WAN Neighbor - 10.50.4.1

IP: 10.50.4.1

Interface Member: HUB1-VPN4

Performance SLA: HUB_HC

SLA: 1

Role: Standalone

Advanced Options >

Neighbor

Neighbor	Role	Interface Member	Performance SLA	SLA
10.50.1.1	Standalone	HUB1-VPN1	HUB_HC	1
10.50.2.1	Standalone	HUB1-VPN2	HUB_HC	1
10.50.3.1	Standalone	HUB1-VPN3	HUB_HC	1
10.50.4.1	Standalone	HUB1-VPN4	HUB_HC	1

```
1 config system sdwan
2     config neighbor
3         edit "10.50.3.1"
4             set health-check "HUB_HC"
5             set sla-id 1
6             set member 5
7         next
8         edit "10.50.4.1"
9             set health-check "HUB_HC"
10            set sla-id 1
11            set member 6
12        next
13    end
14 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           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
S      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]
S      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
S      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]
S      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]

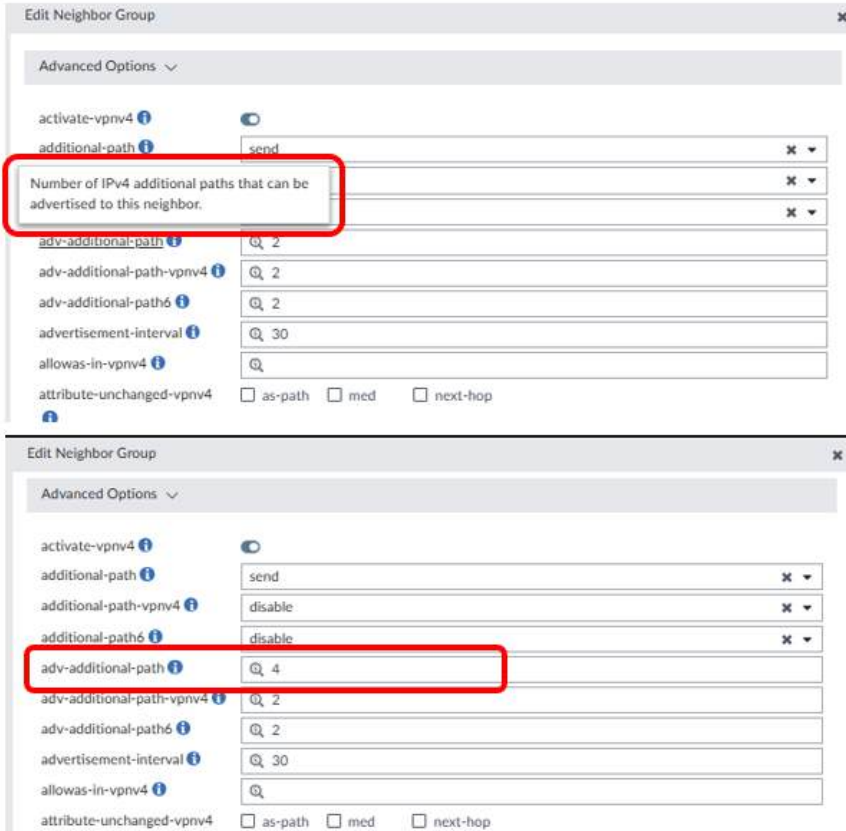
BR2 # get router info bgp summary

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.



```

1 config router bgp
2     config neighbor-group
3         edit "VPN1"
4             set adv-additional-path 4
5         next
6         edit "VPN2"
7             set adv-additional-path 4
8         next
9         edit "VPN3"
10            set adv-additional-path 4
11        next
12        edit "VPN4"
13            set adv-additional-path 4
14        next
15    end
16 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.

The image shows two screenshots of the Fortinet SD-WAN configuration interface. The top screenshot is for HUB1 and the bottom is for BR1. Both show a table of Firewall Policies.

Name	From	To	Source	Destination	Schedule	Service	Action	NAT	Security Profiles	Log	Bytes
BRANCHES-FMG	DIA	port10	all	PR1-FMG SEC-FMG	always	ALL	ACCEPT	Disabled	no-inspection	All	3.21 MB
BRANCHES-TO-HQ	overlay	port3	all	all	always	ALL	ACCEPT	Disabled	no-inspection	All	2.90 kB
HQ-TO-BRANCHES	port3	overlay	all	all	always	ALL	ACCEPT	Disabled	no-inspection	All	1.18 kB
BRANCH-TO-BRANCH	overlay	overlay	all	all	always	ALL	ACCEPT	Disabled	no-inspection	All	84 B
BRANCHES-TO-LOOPBACK	overlay	HUB_LOOP	all	HUB1-LOOPBACK	always	ALL_ICMP	ACCEPT	Disabled	no-inspection	All	25.41 MB
DIA	port3	DIA	all	all	always	ALL	ACCEPT	Enabled	no-inspection	All	683.68 MB
Implicit Deny	any	any	all	all	always	ALL	DENY			Disabled	24.40 kB

Name	From	To	Source	Destination	Schedule	Service	Action	NAT	Security Profiles	Log	Bytes
LAN-TO-HQ&BRANCHES	port3	overlay	all	all	always	ALL	ACCEPT	Disabled	no-inspection	All	6.91 kB
HQ&BRANCHES-TO-LAN	overlay	port3	all	all	always	ALL	ACCEPT	Disabled	no-inspection	All	336 B
DIA	port3	DIA	all	all	always	ALL	ACCEPT	Enabled	no-inspection	All	1.18 kB
Implicit Deny	any	any	all	all	always	ALL	DENY			Disabled	5.54 kB

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.

The screenshot shows the SD-WAN Rules configuration table. The table has columns for ID, Name, Source, Destination, Criteria, Members, Status, Performance SLA, and Port. The rules are as follows:

ID	Name	Source	Destination	Criteria	Members	Status	Performance SLA	Port
1	TUN1	Data-Networks	Route Tag: 1		VPN1	Enable		
2	TUN2	Data-Networks	Route Tag: 2		VPN2	Enable		
5	TUN4	Data-Networks	Route Tag: 4		VPN4	Enable		
4	TUN3	Data-Networks	Route Tag: 3		VPN3	Enable		
3	TUN11	Data-Networks	Route Tag: 11		VPN1 VPN2 VPN3 VPN4	Enable		
	sd-wan	ALL	ALL	Source IP	ALL			

```

1 config system sdwan
2   config service
3     edit 5
4       set name "TUN4"
5       set route-tag 4
6       set src "Data-Networks"
7       set priority-members 6
8     next
9     edit 4
10      set name "TUN3"
11      set route-tag 3
12      set src "Data-Networks"
13      set priority-members 5
14    next
15    edit 3
16      set priority-members 3 4 5 6
17    next
18    move 3 after 4
19  end
20 end
  
```

The screenshot shows the Fortinet SD-WAN configuration for HUB1, specifically the SD-WAN Zones table. The table has columns for ID, Name, Source, Destination, Criteria, Members, Hit Count, and Last Used. The zones are as follows:

ID	Name	Source	Destination	Criteria	Members	Hit Count	Last Used
1	TUN1	Data-Networks	Route tag: 1		VPN1	0	19 hours ago
2	TUN2	Data-Networks	Route tag: 2		VPN2	0	19 hours ago
5	TUN4	Data-Networks	Route tag: 4		VPN4	0	12 minutes ago
4	TUN3	Data-Networks	Route tag: 3		VPN3	0	12 minutes ago
3	TUN11	Data-Networks	Route tag: 11		VPN1 VPN2 VPN3 VPN4	0	19 hours ago
	sd-wan	all	all	Source IP	any		

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.

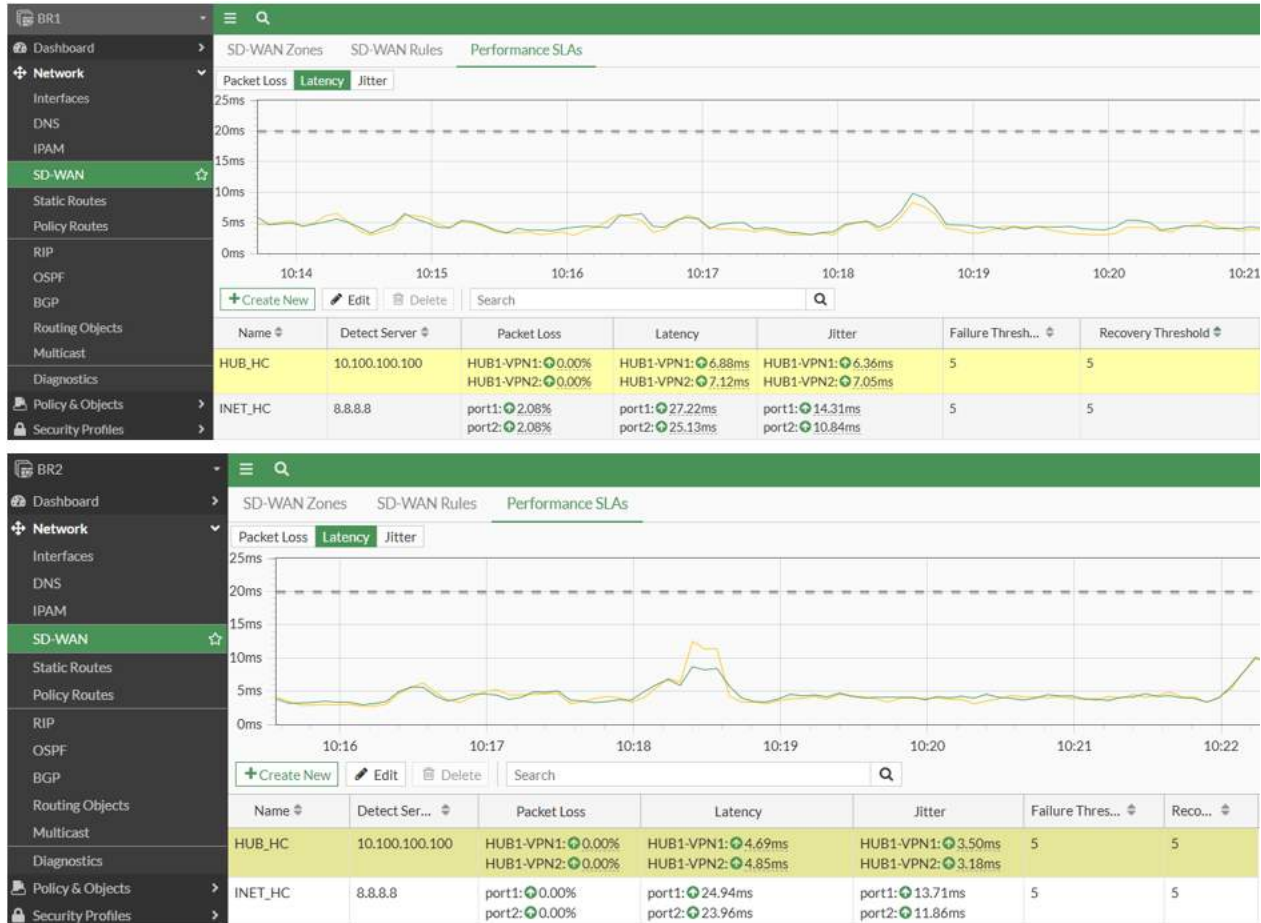
Interface membership is restricted, i.e. Internet access to use DIA (port1 and port2) while the overlay tunnels are used access to the HUB networks.

ID	Name	Source	Destination	Criteria	Members	Status	Performance
1	HQ-BEST	all	Data-Networks	Latency	HUB1-VPN1 HUB1-VPN2 HUB1-VPN3 HUB1-VPN4	Enable	HUB_HC
2	INTERNET	Data-Networks	all	Latency	port1 port2	Enable	INET_HC
	sd-wan	ALL	ALL	Source IP	ALL		

```
1 config system sdwan
2   config health-check
3     edit "INET_HC"
4       set server 8.8.8.8
5       set members 4 3
6     config sla
7       edit 1
8         set latency-threshold 50
9         set jitter-threshold 50
10        set packetloss-threshold 5
11      next
12    end
13  next
14 end
15 config service
16   edit 1
17     set name "HQ-BEST"
18     set mode priority
19     set dst "Data-Networks"
20     set src "all"
21     set health-check "HUB_HC"
22     set priority-members 1 2 5 6
23   next
24   edit 2
25     set name "INTERNET"
26     set mode priority
27     set dst "all"
28     set src "Data-Networks"
29     set health-check "INET_HC"
30     set priority-members 3 4
31   next
32 end
33 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.

Edit Performance SLA - HUB_HC

Name: HUB_HC

IP Version: IPv4

Probe Mode: Active

Enable Probe Packets:

Protocol: Ping

Server: 10.100.100.100

Participants:

- HUB1-VPN1
- HUB1-VPN2
- HUB1-VPN3
- HUB1-VPN4

4 entries selected

```

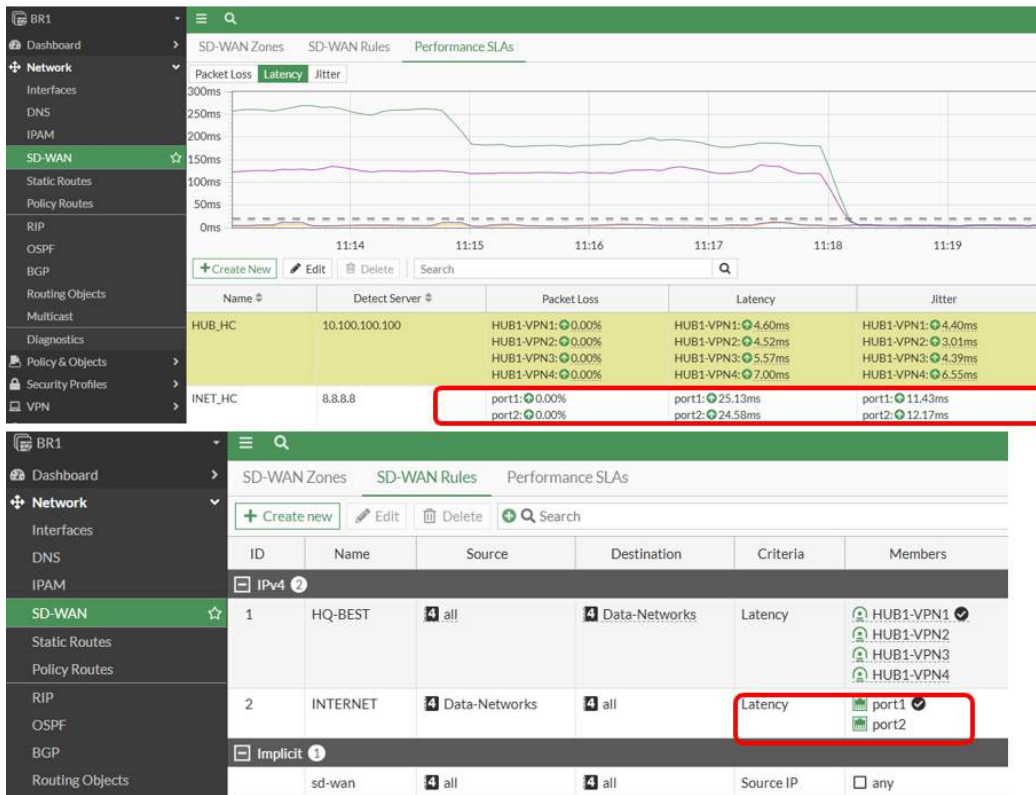
1 config system sdwan
2   config health-check
3     edit "HUB_HC"
4       set members 6 5 1 2
5     next
6   end
7 end
                    
```

All overlay interfaces are now used with the SLA.



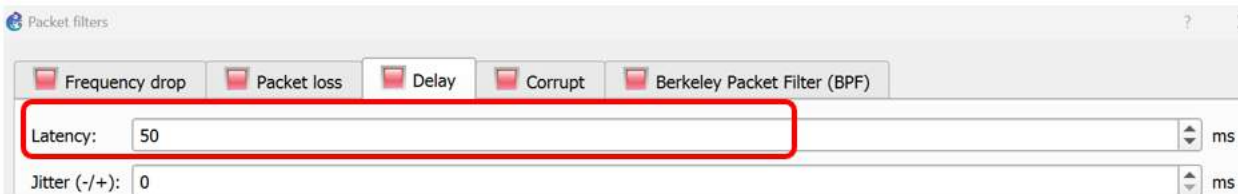
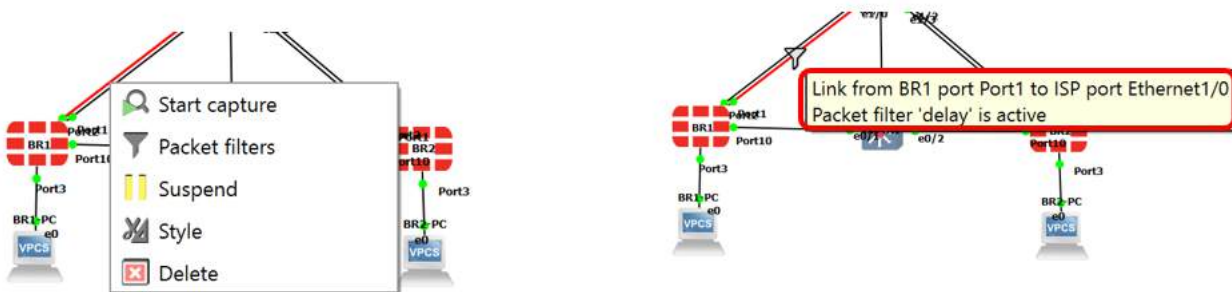
LINK Performance Fluctuation

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

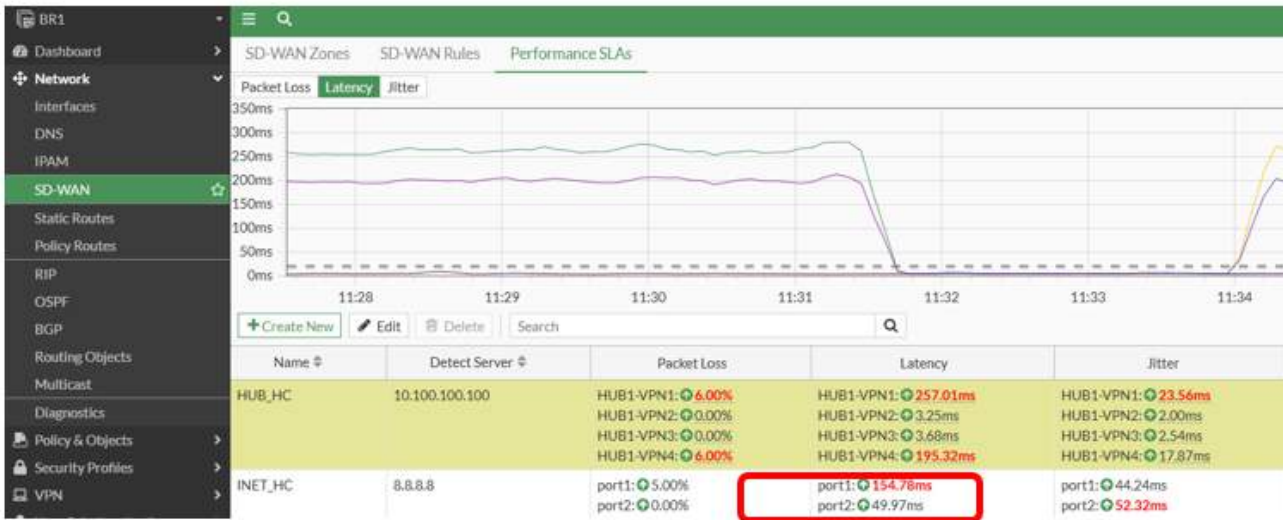


GNS3 packet filters allow manipulation of link quality.

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.



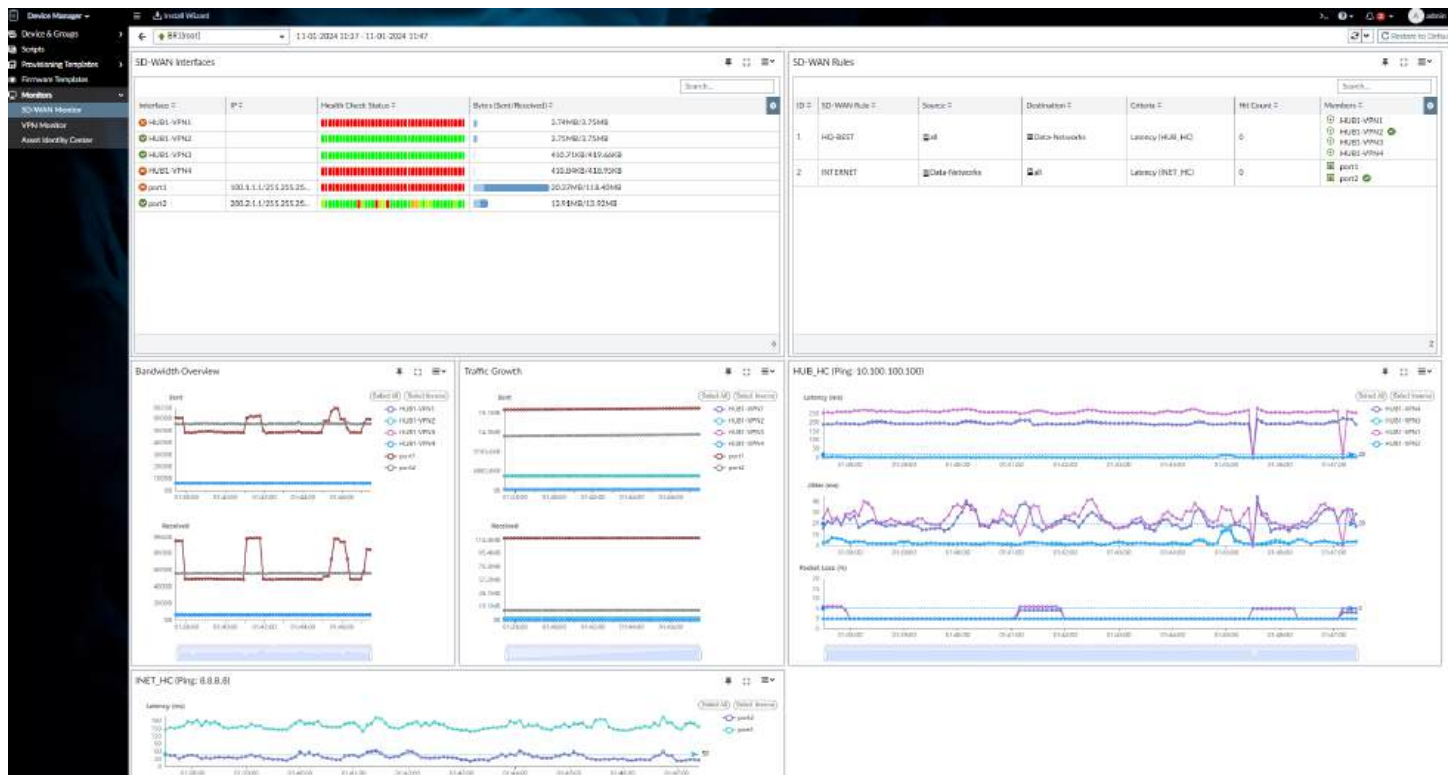
The screenshot displays the SD-WAN Rules configuration page for BR1. It shows a table with columns for ID, Name, Source, Destination, Criteria, and Members. Two rules are visible: HQ-BEST and INTERNET. The INTERNET rule has a latency criterion and its members are port1 and port2, with port2 being selected and circled in red.

ID	Name	Source	Destination	Criteria	Members
1	HQ-BEST	all	Data-Networks	Latency	HUB1-VPN1 HUB1-VPN2 HUB1-VPN3 HUB1-VPN4
2	INTERNET	Data-Networks	all	Latency	port1 port2

SDWAN MONITORING From FortiManager

FortiManager offers monitoring options to have visibility of all available tunnels

Device	SD-WAN Interface	Upload	Download
BR1[root]	HUB1-VPN1	0% 653 bps/0 bp	0% 646 bps/0 bps
	HUB1-VPN2	0% 661 bps/0 bp	0% 664 bps/0 bps
	HUB1-VPN3	0% 641 bps/0 bp	0% 644 bps/0 bps
	HUB1-VPN4	0% 653 bps/0 bp	0% 653 bps/0 bps
	port1	0% 4.9 Kbps/0 bp	0% 4.9 Kbps/0 bps
	port2	0% 5.7 Kbps/0 bp	0% 5.7 Kbps/0 bps
BR2[root]	HUB1-VPN1	0% 653 bps/0 bp	0% 653 bps/0 bps
	HUB1-VPN2	0% 653 bps/0 bp	0% 653 bps/0 bps
	HUB1-VPN3	0% 653 bps/0 bp	0% 653 bps/0 bps
	HUB1-VPN4	0% 653 bps/0 bp	0% 653 bps/0 bps
	port1	0% 4.9 Kbps/0 bp	0% 4.9 Kbps/0 bps
	port2	0% 4.9 Kbps/0 bp	0% 4.9 Kbps/0 bps
HUB1[root]	port1	0% 7.9 Kbps/0 bp	0% 7.9 Kbps/0 bps
	port2	0% 7.9 Kbps/0 bp	0% 7.9 Kbps/0 bps
	VPN1	0% 1.3 Kbps/0 bp	0% 1.3 Kbps/0 bps
	VPN2	0% 1.3 Kbps/0 bp	0% 1.3 Kbps/0 bps
	VPN3	0% 1.3 Kbps/0 bp	0% 1.3 Kbps/0 bps
	VPN4	0% 1.3 Kbps/0 bp	0% 1.3 Kbps/0 bps



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

The screenshot displays the Fortinet SD-WAN Monitor interface. At the top, there are navigation tabs for 'Device Manager', 'Install Wizard', and 'VPN Monitor'. The main area shows a map of Kenya with three orange circular icons representing VPN hubs: one in the north (Nairobi), one in the center (Nairobi), and one in the south (Mombasa). Green lines represent VPN tunnels connecting these hubs. Below the map is a table with the following columns: Device, Name, Type, Remote Gateway, Peer ID, Incoming Data, Outgoing Data, Phase 1, and Phase 2 Selectors.

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 3 questions

- How many FortiManager instances can a FortiGate unit reference at the same time
 - 2
 - 5
 - 10
- Which configuration should be sent to the FortiGate units first during the initial design phase
 - Device settings
 - Policy package
 - Both at the same time
- 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?
 - Configure interfaces into SDWAN zones at the branch and HUB
 - Configure outbound overlay policies at the branch and inbound policies at the HUB.
 - Any of the above
- 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?
 - Missing firewall policies at the branch.
 - Network outage at the ISP level.
 - Underlay interfaces configured with missing gateways
- The error below was shown when using a Metadata variable in an address object. What is the likely cause

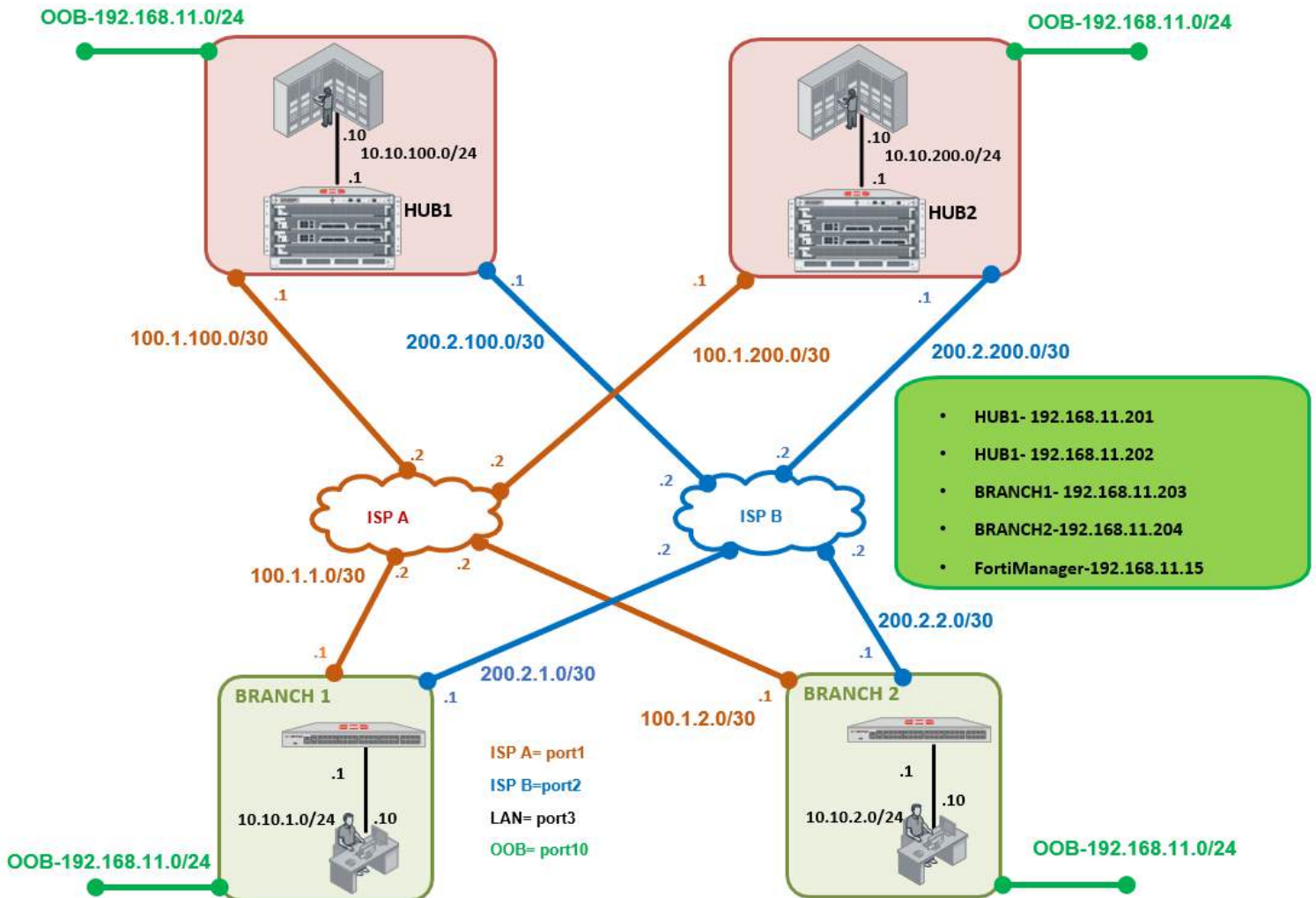
firewall/address/Data/ : invalid subnet ip or mask ✘

 - Duplicate Metadata entries
 - Invalid IP address.
 - Missing default value in the Metadata Variable.
- 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:
 - Route map out
 - Route map out preferable
 - Route map in
- Why is a loopback interface preferred for tracking at the HUB and not a server in the DC.
 - It is more stable with guaranteed uptime
 - It is not subject to interference from HQ LAN delays
 - It is easier to advertise via IBGP
 - 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:

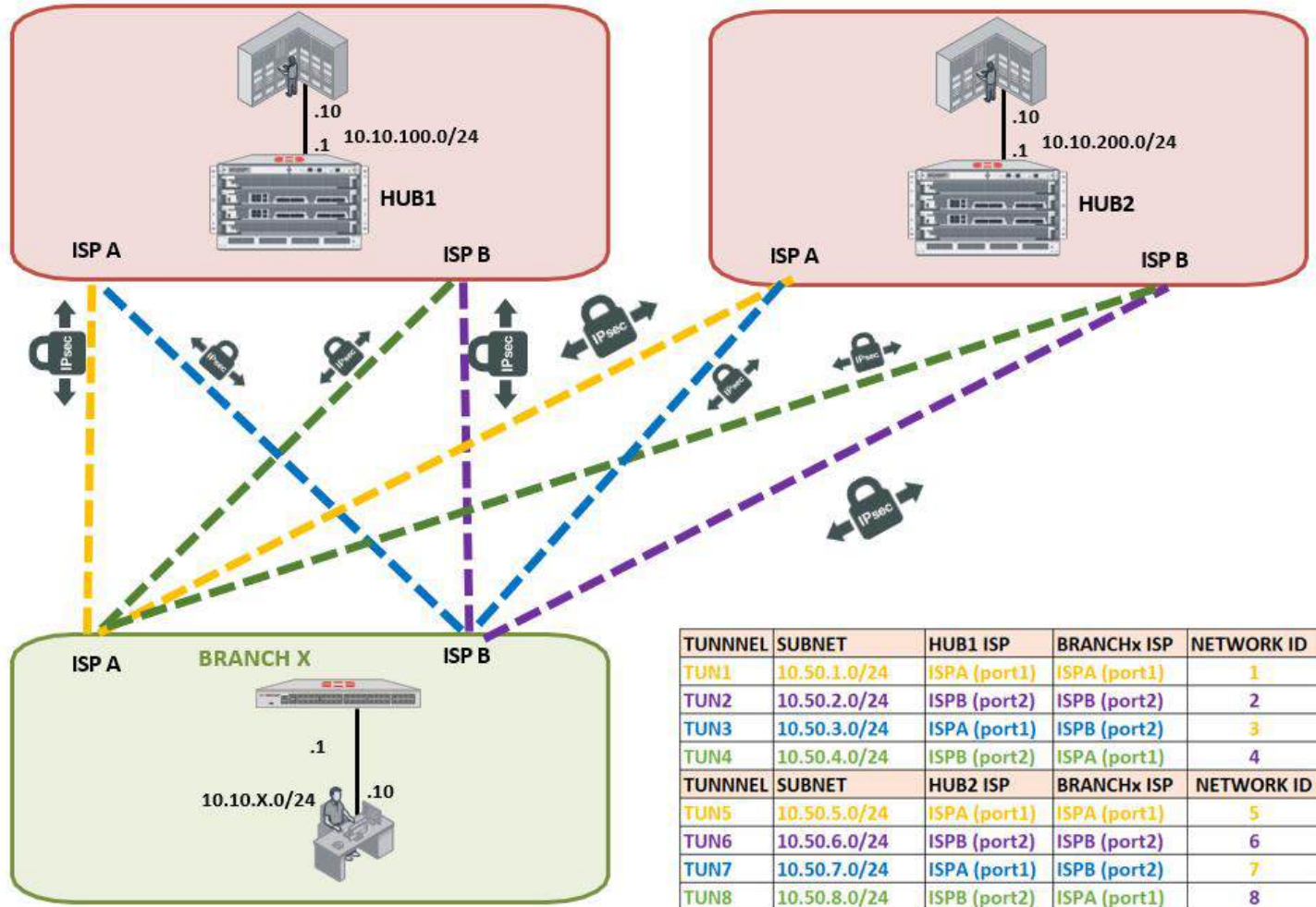


IPSEC Redundancy Between a Branch and 2 HUBS

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 will be active in HUB1 or HUB2 is determined by other factors e.g. DNS.

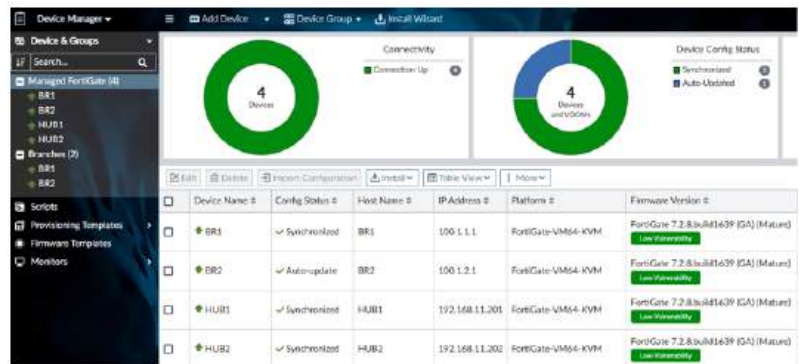
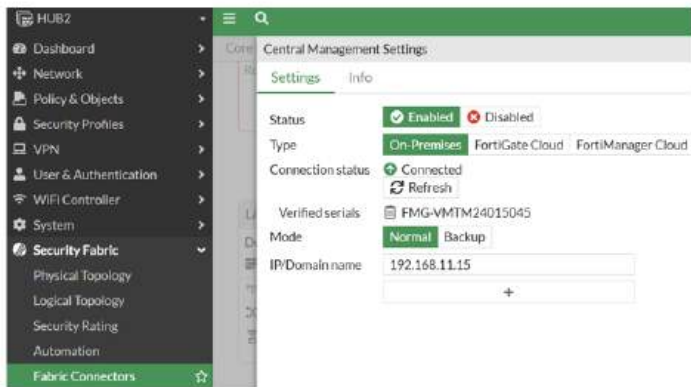
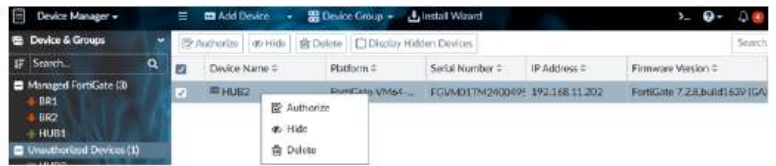
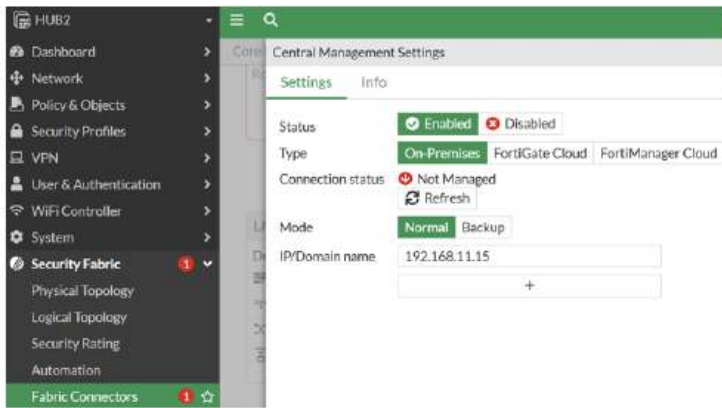
Since unique **network IDs** and subnets are required per tunnel, Tunnel assignment is extended from 5-8 as the unique identifiers. It is not mandatory to configure 4 tunnels per HUB, if link resilience is guaranteed, 2 tunnels could be configured per HUB for a total of 4 per branch.

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



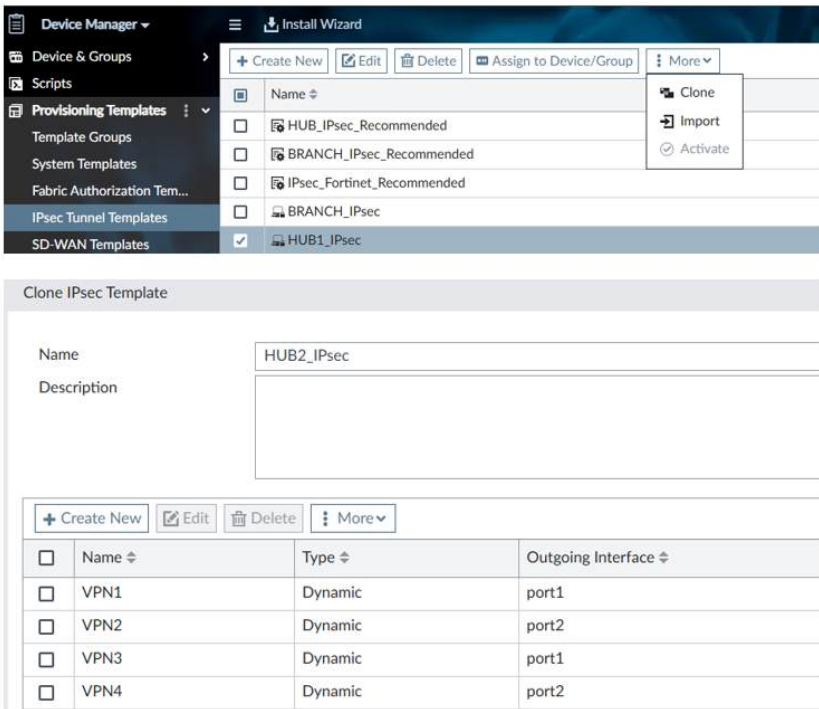
Register HUB2 To FortiManager

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



IPSEC HUB2 Configuration

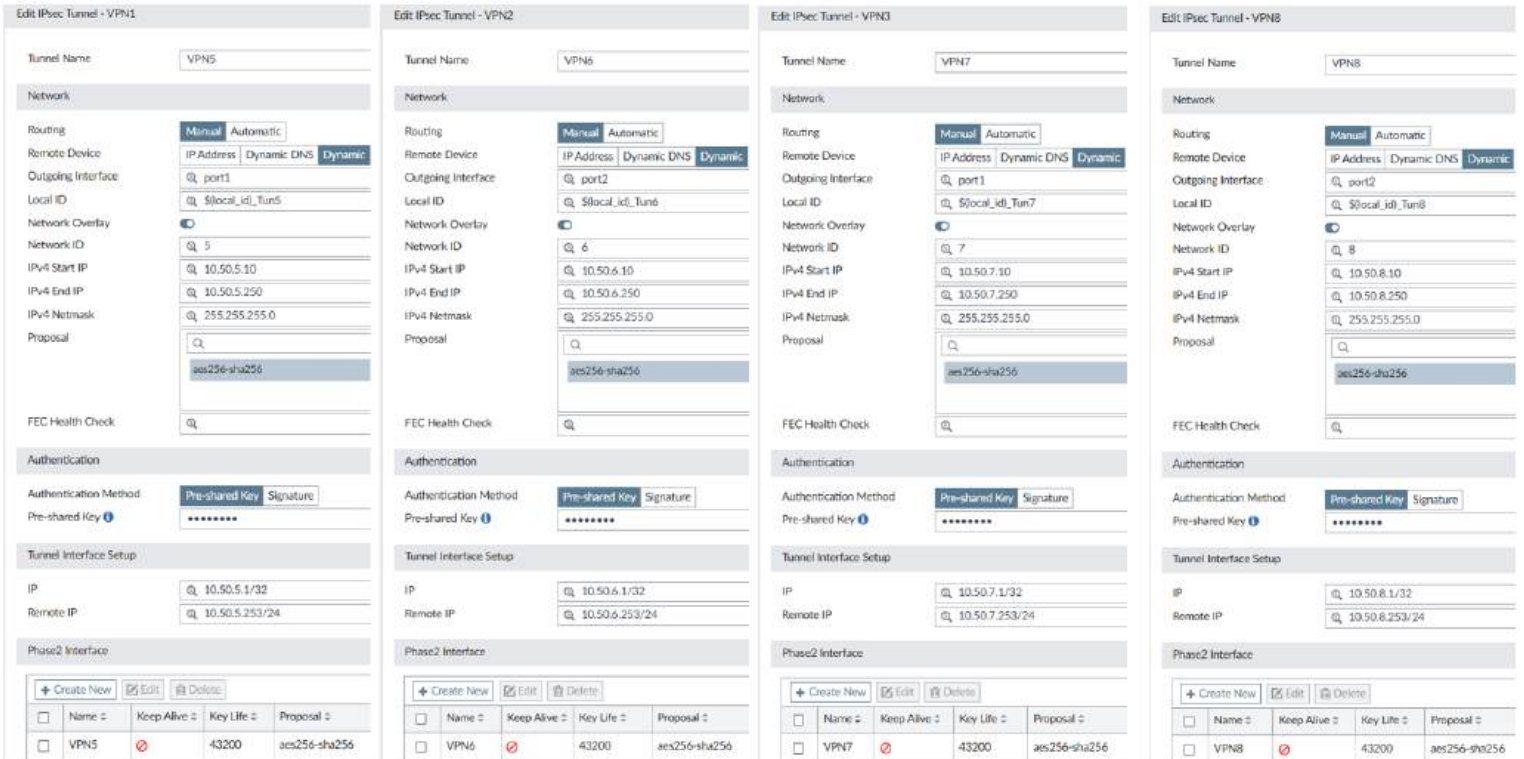
To create VPN5 – VPN8, clone the existing HUB1_IPsec template and then clone the already configured tunnels for HUB1. The Template for HUB2 will have VPNs 5-8 only with configurations changed to match the new tunnels.



VPN	Type	Outgoing Interface
VPN1	Dynamic	port1
VPN2	Dynamic	port2
VPN3	Dynamic	port1
VPN4	Dynamic	port2

TUNNEL	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
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



VPN	Name	Keep Alive	Key Life	Proposal
VPN5	VPN5	43200	43200	aes256-sha256
VPN6	VPN6	43200	43200	aes256-sha256
VPN7	VPN7	43200	43200	aes256-sha256
VPN8	VPN8	43200	43200	aes256-sha256

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

The top row shows two configuration windows. The left window, 'IPsec Template - HUB2_IPsec', has 'Name' set to 'HUB2_IPsec'. Below it is a table of VPN configurations:

Name	Type	Outgoing Interface	Local Interface
VPN5	Dynamic	port1	
VPN6	Dynamic	port2	
VPN7	Dynamic	port1	
VPN8	Dynamic	port2	

The right window, 'Edit Template Group - HUB2-TG', shows 'Name' as 'HUB2-TG' and 'Provisioning Templates' with 'HUB2_IPsec' selected. A note below states: '* Only one template can be selected for each template type.'

The bottom row shows two 'Device Manager' screenshots. The left one shows the 'Provisioning Templates' list with 'HUB2_IPsec' highlighted. The right one shows the 'Template Groups' list with 'HUB2-TG' highlighted, showing it is assigned to 'HUB2 [root]'.

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.

The left screenshot, 'Edit Metadata Variables - branch_id', shows 'Name' as 'branch_id' and 'Default Value' as '222'. The 'Per-Device Mapping' table is as follows:

Mapped Device	Value
BR1 [root]	1
BR2 [root]	2
HUB1 [root]	100
HUB2 [root]	200

The right screenshot, 'Edit Metadata Variables - local_id', shows 'Name' as 'local_id' and 'Default Value' as an empty field. The 'Per-Device Mapping' table is as follows:

Mapped Device	Value
BR1 [root]	Nakuru
BR2 [root]	Mombasa
HUB1 [root]	Nairobi
HUB2 [root]	Thika

IPSEC HUB2 Configuration Preview

```
1 config vpn ipsec phase1-interface
2   edit "VPN5"
3     set type dynamic
4     set interface "port1"
5     set ike-version 2
6     set dpd on-idle
7     set comments "VPN: VPN5 [Created by IPSEC Template]"
8     set proposal aes256-sha256
9     set peertype any
10    set mode-cfg enable
11    set localid "Thika_Tun5"
12    set dpd-retryinterval 60
13    set net-device disable
14    set add-route disable
15    set auto-discovery-sender enable
16    set ipv4-start-ip 10.50.5.10
17    set ipv4-end-ip 10.50.5.250
18    set ipv4-netmask 255.255.255.0
19    set psksecret ENC Z8Zpc/bwU2j1HXCFsp0zLVsmPXEHQqvc6JVo
20    set network-overlay enable
21    set network-id 5
22  next
23 end
24 config system interface
25   edit "VPN5"
26     set vdom "root"
27     set ip 10.50.5.1 255.255.255.255
28     set type tunnel
29     set remote-ip 10.50.5.253 255.255.255.0
30     set snmp-index 116
31     set interface "port1"
32  next
33 end
34 config vpn ipsec phase1-interface
35   edit "VPN6"
36     set type dynamic
37     set interface "port2"
38     set ike-version 2
39     set dpd on-idle
40     set comments "VPN: VPN6 [Created by IPSEC Template]"
41     set proposal aes256-sha256
```

```
42     set peertype any
43     set mode-cfg enable
44     set localid "Thika_Tun6"
45     set dpd-retryinterval 60
46     set net-device disable
47     set add-route disable
48     set auto-discovery-sender enable
49     set ipv4-start-ip 10.50.6.10
50     set ipv4-end-ip 10.50.6.250
51     set ipv4-netmask 255.255.255.0
52     set psksecret ENC Z8Zpc/bwU2j1HXCFsp0zLVsmPXEHQqvc6JVo
53     set network-overlay enable
54     set network-id 6
55  next
56 end
57 config system interface
58   edit "VPN6"
59     set vdom "root"
60     set ip 10.50.6.1 255.255.255.255
61     set type tunnel
62     set remote-ip 10.50.6.253 255.255.255.0
63     set snmp-index 117
64     set interface "port2"
65  next
66 end
67 config vpn ipsec phase1-interface
68   edit "VPN7"
69     set type dynamic
70     set interface "port1"
71     set ike-version 2
72     set dpd on-idle
73     set comments "VPN: VPN7 [Created by IPSEC Template]"
74     set proposal aes256-sha256
75     set peertype any
76     set mode-cfg enable
77     set localid "Thika_Tun7"
78     set dpd-retryinterval 60
79     set net-device disable
80     set add-route disable
81     set auto-discovery-sender enable
82     set ipv4-start-ip 10.50.7.10
83     set ipv4-end-ip 10.50.7.250
```

```
84     set ipv4-netmask 255.255.255.0
85     set psksecret ENC Z8Zpc/bwU2j1HXCFsp0zLVsmPXEHQqvc6JVo
86     set network-overlay enable
87     set network-id 7
88  next
89 end
90 config system interface
91   edit "VPN7"
92     set vdom "root"
93     set ip 10.50.7.1 255.255.255.255
94     set type tunnel
95     set remote-ip 10.50.7.253 255.255.255.0
96     set snmp-index 118
97     set interface "port1"
98  next
99 end
100 config vpn ipsec phase1-interface
101   edit "VPN8"
102     set type dynamic
103     set interface "port2"
104     set ike-version 2
105     set dpd on-idle
106     set comments "VPN: VPN8 [Created by IPSEC Template]"
107     set proposal aes256-sha256
108     set peertype any
109     set mode-cfg enable
110     set localid "Thika_Tun8"
111     set dpd-retryinterval 60
112     set net-device disable
113     set add-route disable
114     set auto-discovery-sender enable
115     set ipv4-start-ip 10.50.8.10
116     set ipv4-end-ip 10.50.8.250
117     set ipv4-netmask 255.255.255.0
118     set psksecret ENC Z8Zpc/bwU2j1HXCFsp0zLVsmPXEHQqvc6JVo
119     set network-overlay enable
120     set network-id 8
121  next
122 end
123 config system interface
124   edit "VPN8"
125     set vdom "root"
```

```
126     set ip 10.50.8.1 255.255.255.255
127     set type tunnel
128     set remote-ip 10.50.8.253 255.255.255.0
129     set snmp-index 119
130     set interface "port2"
131     next
132 end
133 config vpn ipsec phase2-interface
134     edit "VPN5"
135         set phase1name "VPN5"
136         set proposal aes256-sha256
137         set comments "VPN: VPN5 [Created by IPSEC Template]"
138     next
139     edit "VPN6"
140         set phase1name "VPN6"
141         set proposal aes256-sha256
142         set comments "VPN: VPN6 [Created by IPSEC Template]"
143     next
144     edit "VPN7"
145         set phase1name "VPN7"
146         set proposal aes256-sha256
147         set comments "VPN: VPN7 [Created by IPSEC Template]"
148     next
149     edit "VPN8"
150         set phase1name "VPN8"
151         set proposal aes256-sha256
152         set comments "VPN: VPN8 [Created by IPSEC Template]"
153     next
154 end
155
```

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 Template - BRANCH_IPsec

Name: BRANCH_IPsec

Description:

+ Create New | Edit | Delete | More

<input type="checkbox"/>	Name	Type	Outgoing Interface
<input type="checkbox"/>	HUB1-VPN1	Static	port1
<input type="checkbox"/>	HUB1-VPN2	Static	port2
<input type="checkbox"/>	HUB1-VPN3	Static	port2
<input type="checkbox"/>	HUB1-VPN4	Static	port1

Policy & Objects

Object Configurations

<input type="checkbox"/>	Name	Default Value
<input type="checkbox"/>	Br_ISPA_gw	
<input type="checkbox"/>	Br_ISPB_gw	
<input type="checkbox"/>	branch_id	222
<input type="checkbox"/>	hub1_ISPA	100.1.100.1
<input type="checkbox"/>	hub1_ISPB	200.2.100.1
<input type="checkbox"/>	hub2_ISPA	100.1.200.1
<input type="checkbox"/>	hub2_ISPB	200.2.200.1
<input type="checkbox"/>	local_id	
<input type="checkbox"/>	vm_interface_number	1

IPSEC BRANCH HUB2-VPN5-8 CONFIGURATION

Clone IPsec Tunnel - Clone_HUB1-VPN1

Tunnel Name: HUB2-VPN5

Network

Routing: Manual Automatic

Remote Device: IP Address Dynamic DNS Dynamic

Remote Gateway (IP Address): \$hub2_ISPA

Outgoing Interface: port1

Local ID: \$local_id_Tun5

Network Overlay: Network Overlay

Network ID: 5

Proposal: aes256-sha256

FEC Health Check

Authentication: Pre-shared Key Signature

Pre-shared Key: *****

Tunnel Interface Setup

IP: Remote IP

Phase2 Interface

<input type="checkbox"/>	Name	Keep Alive	Key Life	Proposal
<input type="checkbox"/>	HUB2-VPN5		43200	aes256-sha256

Clone IPsec Tunnel - Clone_HUB1-VPN2

Tunnel Name: HUB2-VPN6

Network

Routing: Manual Automatic

Remote Device: IP Address Dynamic DNS Dynamic

Remote Gateway (IP Address): \$hub2_ISPB

Outgoing Interface: port2

Local ID: \$local_id_Tun6

Network Overlay: Network Overlay

Network ID: 6

Proposal: aes256-sha256

FEC Health Check

Authentication: Pre-shared Key Signature

Pre-shared Key: *****

Tunnel Interface Setup

IP: Remote IP

Phase2 Interface

<input type="checkbox"/>	Name	Keep Alive	Key Life	Proposal
<input type="checkbox"/>	HUB2-VPN6		43200	aes256-sha256

Clone IPsec Tunnel - Clone_HUB1-VPN3

Tunnel Name: HUB2-VPN7

Network

Routing: Manual Automatic

Remote Device: IP Address Dynamic DNS Dynamic

Remote Gateway (IP Address): \$hub2_ISPA

Outgoing Interface: port2

Local ID: \$local_id_Tun7

Network Overlay: Network Overlay

Network ID: 7

Proposal: aes256-sha256

FEC Health Check

Authentication: Pre-shared Key Signature

Pre-shared Key: *****

Tunnel Interface Setup

IP: Remote IP

Phase2 Interface

<input type="checkbox"/>	Name	Keep Alive	Key Life	Proposal
<input type="checkbox"/>	HUB2-VPN7		43200	aes256-sha256

Clone IPsec Tunnel - Clone_HUB1-VPN4

Tunnel Name: HUB2-VPN8

Network

Routing: Manual Automatic

Remote Device: IP Address Dynamic DNS Dynamic

Remote Gateway (IP Address): \$hub2_ISPB

Outgoing Interface: port1

Local ID: \$local_id_Tun8

Network Overlay: Network Overlay

Network ID: 8

Proposal: aes256-sha256

FEC Health Check

Authentication: Pre-shared Key Signature

Pre-shared Key: *****

Tunnel Interface Setup

IP: Remote IP

Phase2 Interface

<input type="checkbox"/>	Name	Keep Alive	Key Life	Proposal
<input type="checkbox"/>	HUB2-VPN8		43200	aes256-sha256

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.

The screenshot shows the Fortinet Device Manager interface. The top part displays a list of Provisioning Templates:

Template Group Name	Provisioning Templates	Assigned to Device/Group
HUB1-TG	HUB1_IPsec HUB1_BGP HUB1-SDWAN HUB-CLI	1 Device in Total HUB1 [root]
BRANCHES-TG	BRANCH_IPsec BRANCH_BGP BRANCH_SDWAN BR_LOOP	2 Devices in Total Branches (2)
HUB2-TG	HUB2_IPsec	1 Device in Total HUB2 [root]

The bottom part shows the configuration for the 'BRANCH_IPsec' template:

Name: BRANCH_IPsec

Description: [Empty]

Name	Type	Outgoing Interface	Local Interface
HUB1-VPN1	Static	port1	
HUB1-VPN2	Static	port2	
HUB1-VPN3	Static	port2	
HUB1-VPN4	Static	port1	
HUB2-VPN5	Static	port1	
HUB2-VPN6	Static	port2	
HUB2-VPN7	Static	port2	
HUB2-VPN8	Static	port1	

The first four rows of the table (HUB1-VPN1 to HUB1-VPN4) are highlighted with a red circle.

IPSEC BRANCH HUB2-VPN5-8 Configuration Preview

```

1 config vpn ipsec phase1-interface
2   edit "HUB2-VPN5"
3     set interface "port1"
4     set ike-version 2
5     set comments "VPN: HUB2-VPN5 [Created by IPSEC Template]"
6     set proposal aes256-sha256
7     set peertype any
8     set mode-cfg enable
9     set localid "Nakuru_Tun5"
10    set remote-gw 100.1.200.1
11    set idle-timeout enable
12    set net-device enable
13    set add-route disable
14    set auto-discovery-receiver enable
15    set psksecret ENC 282pc/bwU2j1HxCFsp0zLVsmPXEwQvc6Jvovq8gt3R
16    set network-overlay enable
17    set network-id 5
18    set auto-discovery-shortcuts dependent
19  next
20 end
21 config system interface
22   edit "HUB2-VPN5"
23     set vdom "root"
24     set type tunnel
25     set snmp-index 121
26     set interface "port1"
27  next
28 end
29 config vpn ipsec phase1-interface
30   edit "HUB2-VPN6"
31     set interface "port2"
32     set ike-version 2
33     set comments "VPN: HUB2-VPN6 [Created by IPSEC Template]"
34     set proposal aes256-sha256
35     set peertype any
36     set mode-cfg enable
37     set localid "Nakuru_Tun6"
38     set remote-gw 200.2.200.1
39     set idle-timeout enable
40     set net-device enable
41     set add-route disable
42     set auto-discovery-receiver enable
43
44     set psksecret ENC 282pc/bwU2j1HxCFsp0zLVsmPXEwQvc6Jvovq8gt3R
45     set network-overlay enable
46     set network-id 6
47     set auto-discovery-shortcuts dependent
48  next
49 end
50 config system interface
51   edit "HUB2-VPN6"
52     set vdom "root"
53     set type tunnel
54     set snmp-index 122
55     set interface "port2"
56  next
57 end
58 config vpn ipsec phase1-interface
59   edit "HUB2-VPN7"
60     set interface "port2"
61     set ike-version 2
62     set comments "VPN: HUB2-VPN7 [Created by IPSEC Template]"
63     set proposal aes256-sha256
64     set peertype any
65     set mode-cfg enable
66     set localid "Nakuru_Tun7"
67     set remote-gw 100.1.200.1
68     set idle-timeout enable
69     set net-device enable
70     set add-route disable
71     set auto-discovery-receiver enable
72     set psksecret ENC 282pc/bwU2j1HxCFsp0zLVsmPXEwQvc6Jvovq8gt3R
73     set network-overlay enable
74     set network-id 7
75     set auto-discovery-shortcuts dependent
76  next
77 end
78 config system interface
79   edit "HUB2-VPN7"
80     set vdom "root"
81     set type tunnel
82     set snmp-index 123
83     set interface "port2"
84  next
85 end
86 config vpn ipsec phase1-interface
87   edit "HUB2-VPN8"
88     set interface "port1"
89     set ike-version 2
90     set comments "VPN: HUB2-VPN8 [Created by IPSEC Template]"
91     set proposal aes256-sha256
92     set peertype any
93     set mode-cfg enable
94     set localid "Nakuru_Tun8"
95     set remote-gw 200.2.200.1
96     set idle-timeout enable
97     set net-device enable
98     set add-route disable
99     set auto-discovery-receiver enable
100    set psksecret ENC 282pc/bwU2j1HxCFsp0zLVsmPXEwQvc6Jvovq8gt3R
101    set network-overlay enable
102    set network-id 8
103    set auto-discovery-shortcuts dependent
104  next
105 end
106 config system interface
107   edit "HUB2-VPN8"
108     set vdom "root"
109     set type tunnel
110     set snmp-index 124
111     set interface "port1"
112  next
113 end
114 config vpn ipsec phase2-interface
115   edit "HUB2-VPN5"
116     set phaseName "HUB2-VPN5"
117     set proposal aes256-sha256
118     set auto-negotiate enable
119     set comments "VPN: HUB2-VPN5 [Created by IPSEC Template]"
120  next
121   edit "HUB2-VPN6"
122     set phaseName "HUB2-VPN6"
123     set proposal aes256-sha256
124     set auto-negotiate enable
125     set comments "VPN: HUB2-VPN6 [Created by IPSEC Template]"
126  next
127   edit "HUB2-VPN7"
128     set phaseName "HUB2-VPN7"
129     set proposal aes256-sha256
130     set auto-negotiate enable
131     set comments "VPN: HUB2-VPN7 [Created by IPSEC Template]"
132  next
133   edit "HUB2-VPN8"
134     set phaseName "HUB2-VPN8"
135     set proposal aes256-sha256
136     set auto-negotiate enable
137     set comments "VPN: HUB2-VPN8 [Created by IPSEC Template]"
138  next
139 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.

Name	Remote Gateway	Peer ID	Incoming Data	Outgoing Data	Phase 1	Phase 2 Selectors
HUB2-VPN7	100.1.200.1		0 B	0 B	HUB2-VPN7	HUB2-VPN7
HUB2-VPN8	200.2.200.1		0 B	0 B	HUB2-VPN8	HUB2-VPN8
HUB2-VPN5	100.1.200.1		0 B	0 B	HUB2-VPN5	HUB2-VPN5
HUB2-VPN6	200.2.200.1		0 B	0 B	HUB2-VPN6	HUB2-VPN6
HUB1-VPN1	100.1.100.1	Nairobi_Tun1	26.2.1 kB	267.54 kB	HUB1-VPN1	HUB1-VPN1
HUB1-VPN4	200.2.100.1	Nairobi_Tun4	264.43 kB	263.25 kB	HUB1-VPN4	HUB1-VPN4
HUB1-VPN2	200.2.100.1	Nairobi_Tun2	342.61 kB	341.50 kB	HUB1-VPN2	HUB1-VPN2
HUB1-VPN3	100.1.100.1	Nairobi_Tun3	342.60 kB	341.60 kB	HUB1-VPN3	HUB1-VPN3

Name	Remote Gateway	Peer ID	Incoming Data	Outgoing Data	Phase 1	Phase 2 Selectors
HUB1-VPN1	100.1.100.1	Nairobi_Tun1	356.08 kB	354.04 kB	HUB1-VPN1	HUB1-VPN1
HUB1-VPN2	200.2.100.1	Nairobi_Tun2	354.08 kB	354.86 kB	HUB1-VPN2	HUB1-VPN2
HUB1-VPN3	100.1.100.1	Nairobi_Tun3	270.83 kB	269.86 kB	HUB1-VPN3	HUB1-VPN3
HUB1-VPN4	200.2.100.1	Nairobi_Tun4	356.72 kB	355.06 kB	HUB1-VPN4	HUB1-VPN4
HUB2-VPN5	100.1.200.1		0 B	0 B	HUB2-VPN5	HUB2-VPN5
HUB2-VPN6	200.2.200.1		0 B	0 B	HUB2-VPN6	HUB2-VPN6
HUB2-VPN7	100.1.200.1		0 B	0 B	HUB2-VPN7	HUB2-VPN7
HUB2-VPN8	200.2.200.1		0 B	0 B	HUB2-VPN8	HUB2-VPN8

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.

ID	Interface	Gateway	Cost	Priority	Status
1	port1	100.1.200.2	0	1	Enable
2	port2	200.2.200.2	0	1	Enable
3	VPN5	0.0.0.0	0	1	Enable
4	VPN6	0.0.0.0	0	1	Enable
5	VPN7	0.0.0.0	0	1	Enable
6	VPN8	0.0.0.0	0	1	Enable

```

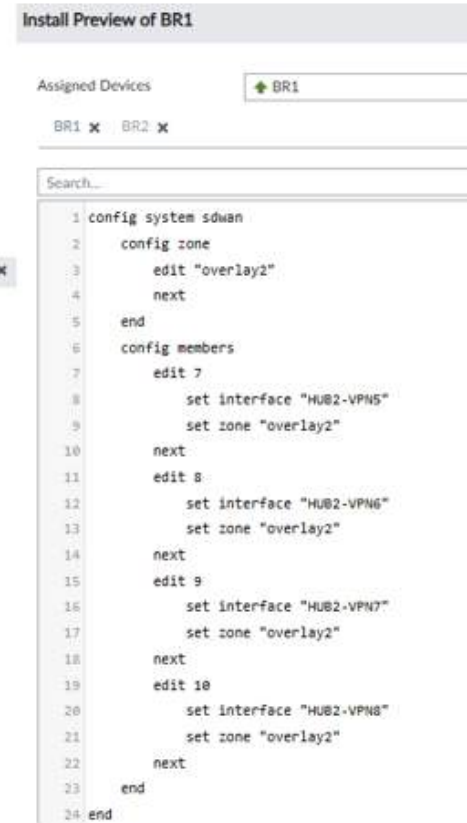
1 config system sdwan
2   set status enable
3   config zone
4     edit "DIA"
5     next
6     edit "overlay"
7     next
8   end
9   config members
10  edit 1
11    set interface "port1"
12    set zone "DIA"
13    set gateway 100.1.200.2
14  next
15  edit 2
16    set interface "port2"
17    set zone "DIA"
18    set gateway 200.2.200.2
19  next
20  edit 3
21    set interface "VPN5"
22    set zone "overlay"
23  next
24  edit 4
25    set interface "VPN6"
26    set zone "overlay"
27  next
28  edit 5
29    set interface "VPN7"
30    set zone "overlay"
31  next
32  edit 6
33    set interface "VPN8"
34    set zone "overlay"
35  next
36 end
37 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.

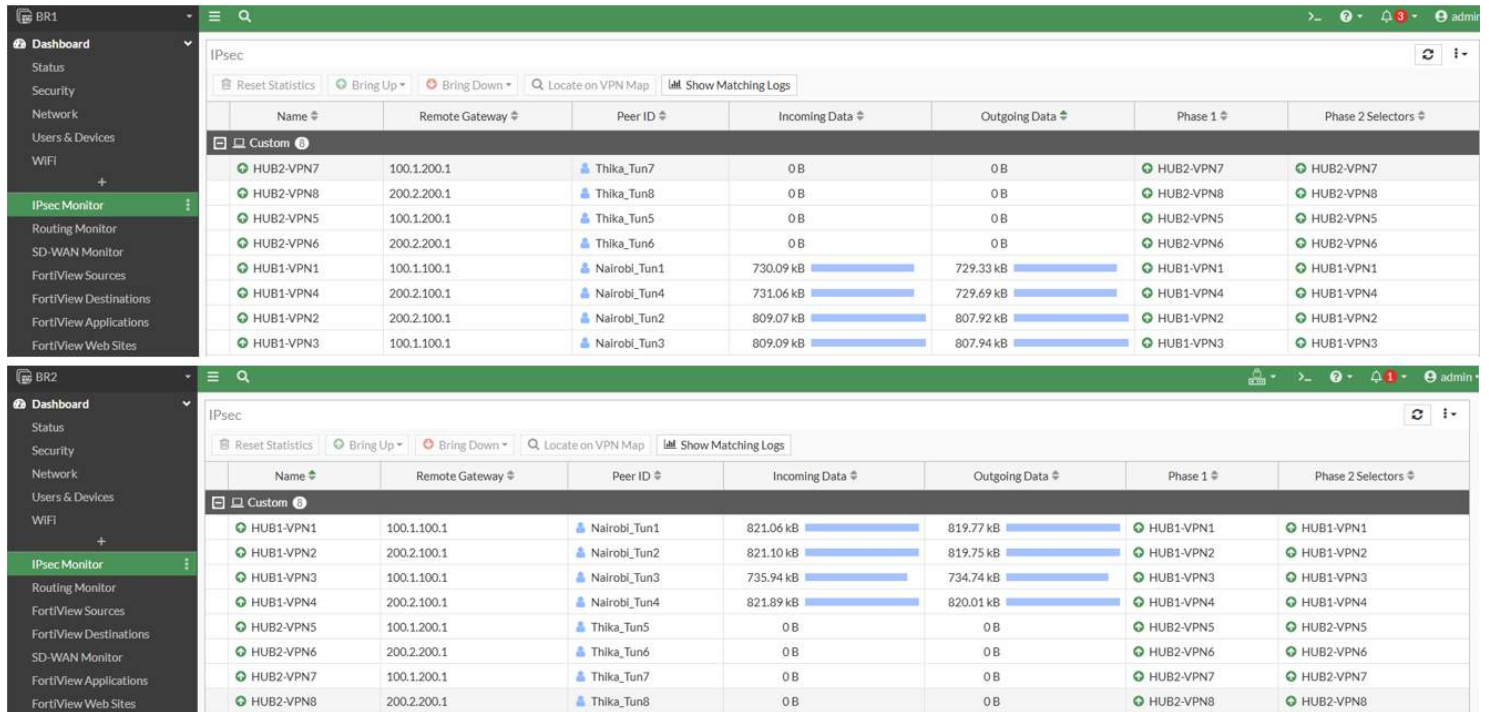


Zone	VPN	Remote Gateway	Peer ID	Phase 1	Phase 2	Status
overlay2	7	HUB2-VPN5	0.0.0.0	0	1	Enable
overlay2	8	HUB2-VPN6	0.0.0.0	0	1	Enable
overlay2	9	HUB2-VPN7	0.0.0.0	0	1	Enable
overlay2	10	HUB2-VPN8	0.0.0.0	0	1	Enable



```
1 config system sdwan
2   config zone
3     edit "overlay2"
4       next
5     end
6   config members
7     edit 7
8       set interface "HUB2-VPN5"
9       set zone "overlay2"
10    next
11   edit 8
12     set interface "HUB2-VPN6"
13     set zone "overlay2"
14   next
15   edit 9
16     set interface "HUB2-VPN7"
17     set zone "overlay2"
18   next
19   edit 10
20     set interface "HUB2-VPN8"
21     set zone "overlay2"
22   next
23 end
24 end
```

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



Name	Remote Gateway	Peer ID	Incoming Data	Outgoing Data	Phase 1	Phase 2 Selectors
HUB2-VPN7	100.1.200.1	Thika_Tun7	0B	0B	HUB2-VPN7	HUB2-VPN7
HUB2-VPN8	200.2.200.1	Thika_Tun8	0B	0B	HUB2-VPN8	HUB2-VPN8
HUB2-VPN5	100.1.200.1	Thika_Tun5	0B	0B	HUB2-VPN5	HUB2-VPN5
HUB2-VPN6	200.2.200.1	Thika_Tun6	0B	0B	HUB2-VPN6	HUB2-VPN6
HUB1-VPN1	100.1.100.1	Nairobi_Tun1	730.09 kB	729.33 kB	HUB1-VPN1	HUB1-VPN1
HUB1-VPN4	200.2.100.1	Nairobi_Tun4	731.06 kB	729.69 kB	HUB1-VPN4	HUB1-VPN4
HUB1-VPN2	200.2.100.1	Nairobi_Tun2	809.07 kB	807.92 kB	HUB1-VPN2	HUB1-VPN2
HUB1-VPN3	100.1.100.1	Nairobi_Tun3	809.09 kB	807.94 kB	HUB1-VPN3	HUB1-VPN3

Name	Remote Gateway	Peer ID	Incoming Data	Outgoing Data	Phase 1	Phase 2 Selectors
HUB1-VPN1	100.1.100.1	Nairobi_Tun1	821.06 kB	819.77 kB	HUB1-VPN1	HUB1-VPN1
HUB1-VPN2	200.2.100.1	Nairobi_Tun2	821.10 kB	819.75 kB	HUB1-VPN2	HUB1-VPN2
HUB1-VPN3	100.1.100.1	Nairobi_Tun3	735.94 kB	734.74 kB	HUB1-VPN3	HUB1-VPN3
HUB1-VPN4	200.2.100.1	Nairobi_Tun4	821.89 kB	820.01 kB	HUB1-VPN4	HUB1-VPN4
HUB2-VPN5	100.1.200.1	Thika_Tun5	0B	0B	HUB2-VPN5	HUB2-VPN5
HUB2-VPN6	200.2.200.1	Thika_Tun6	0B	0B	HUB2-VPN6	HUB2-VPN6
HUB2-VPN7	100.1.200.1	Thika_Tun7	0B	0B	HUB2-VPN7	HUB2-VPN7
HUB2-VPN8	200.2.200.1	Thika_Tun8	0B	0B	HUB2-VPN8	HUB2-VPN8

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.

Name	Remote Gateway	Peer ID	Incoming Data	Outgoing Data	Phase 1	Phase 2 Selectors
VPN5_0	100.1.2.1	Mombasa_Tun5	0 B	0 B	VPN5_0	VPN5
VPN5_1	100.1.1.1	Nakuru_Tun5	0 B	0 B	VPN5_1	VPN5
VPN6_0	200.2.2.1	Mombasa_Tun6	0 B	0 B	VPN6_0	VPN6
VPN6_1	200.2.1.1	Nakuru_Tun6	0 B	0 B	VPN6_1	VPN6
VPN7_0	200.2.2.1	Mombasa_Tun7	0 B	0 B	VPN7_0	VPN7
VPN7_1	200.2.1.1	Nakuru_Tun7	0 B	0 B	VPN7_1	VPN7
VPN8_0	100.1.2.1	Mombasa_Tun8	0 B	0 B	VPN8_0	VPN8
VPN8_1	100.1.1.1	Nakuru_Tun8	0 B	0 B	VPN8_1	VPN8

```
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
```

TUNNEL	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
TUNNEL	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

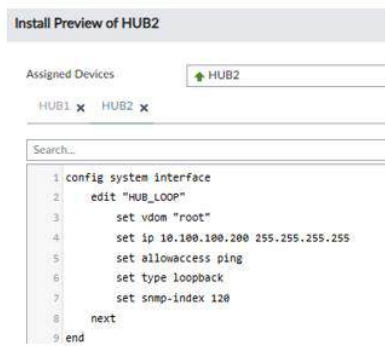
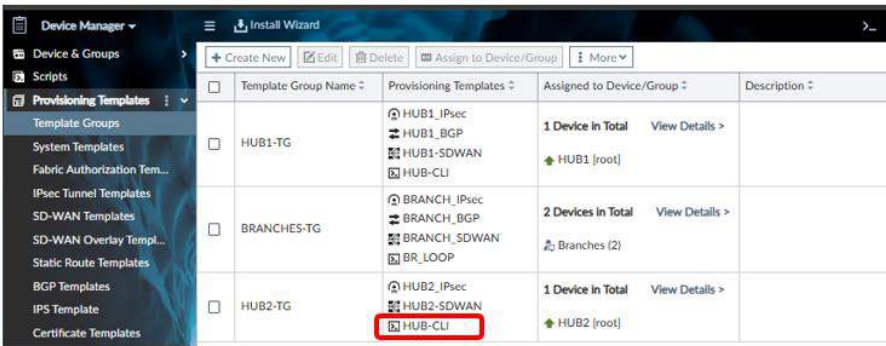
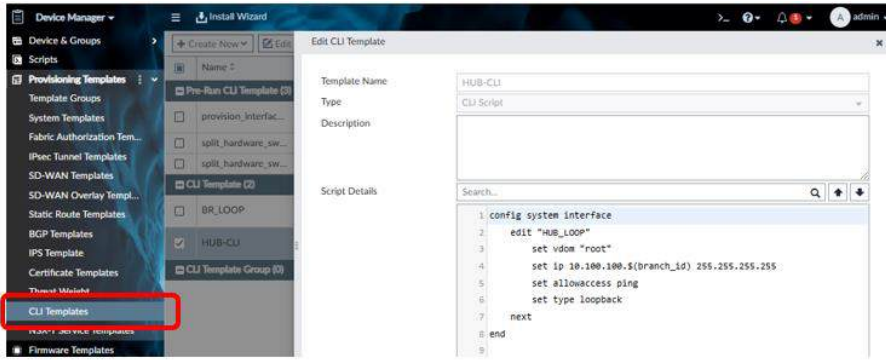
HUB2 Normalized Interface

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.

Mapped Device	Details	Type	Addressing Mode	IP/Netmask	Shaping Profile
BR1 [root]	port3	Physical	Manual	10.10.1.1/255.255.255.0	
BR2 [root]	port3	Physical	Manual	10.10.2.1/255.255.255.0	
HUB1 [root]	port3	Physical	Manual	10.10.100.1/255.255.255.0	
HUB2 [root]	port3	Physical	Manual	10.10.200.1/255.255.255.0	

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

#	Name	From	To	Source	Destination	Schedule	Service	NAT	Action
1	BRANCHES-TO-HQ	overlay	LAN	all	all	always	ALL	Disabled	Accept
2	HQ-TO-BRANCHES	LAN	overlay	all	all	always	ALL	Disabled	Accept
3	BRANCH-TO-BRANCH	overlay	overlay	all	all	always	ALL	Disabled	Accept
4	BRANCHES-TO-LOOPBACK	overlay	HUB_LOOP	all	HUB2-LOOPBACK	always	ALL_ICMP	Disabled	Accept
5	DIA	LAN	DIA	all	all	always	ALL	Enabled	Accept
Implicit (6/6 Total:1)									
6	Implicit Deny	any	any	all	all	always	ALL		Deny

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.

The screenshot shows the Fortinet SD-WAN configuration interface. On the left, the 'Device Manager' sidebar is visible with 'Provisioning Templates' expanded to 'SD-WAN Templates'. The main window is titled 'Edit Performance SLA - HUB2_HC'. The 'Name' is 'HUB2_HC', 'IP Version' is 'IPv4', and 'Probe Mode' is 'Active'. The 'Server' is set to '10.100.100.200'. Under 'Participants', 'All SD-WAN Members' is selected, and a list of participants is shown: HUB2-VPN5, HUB2-VPN6, HUB2-VPN7, and HUB2-VPN8. The 'SLA Target' section shows 'Latency Threshold' at 20 ms, 'Jitter Threshold' at 20 ms, 'Packet Loss Threshold' at 5%, 'Priority IN-SLA' at 0, and 'Priority OUT-SLA' at 0. The 'Link Status' section shows 'Check Interval' at 500 milliseconds, 'Failure Before Inactive' at 5 checks, and 'Restore Link After' at 5 checks. On the right, the 'Install Preview of BR1' window shows the configuration for 'BR1' with the following commands:

```

1 config system sdwan
2   config health-check
3     edit "HUB2_HC"
4       set server 10.100.100.200
5       set update-static-route disable
6       set members 10 9 8 7
7     config sla
8       edit 1
9         set latency-threshold 20
10        set jitter-threshold 20
11        set packetloss-threshold 5
12      next
13    end
14  next
15 end
16 end
17
  
```

HUB2 LOOPBACK Health Check Verification

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

The screenshot shows the Fortinet SD-WAN Performance SLAs monitoring interface for BR1 and BR2. The top panel shows a graph of Packet Loss, Latency, and Jitter over time. The bottom panel shows a table of Performance SLAs for HUB2_HC and HUB1_HC. The table is as follows:

Name #	Detect Server #	Packet Loss	Latency	Jitter	Failure Threshold #	Recovery Threshold #
HUB2_HC	10.100.100.200	HUB2-VPN5: 0.00%	HUB2-VPN5: 6.80ms	HUB2-VPN5: 4.70ms	5	5
		HUB2-VPN6: 0.00%	HUB2-VPN6: 6.44ms	HUB2-VPN6: 3.61ms		
		HUB2-VPN7: 0.00%	HUB2-VPN7: 7.86ms	HUB2-VPN7: 5.34ms		
		HUB2-VPN8: 0.00%	HUB2-VPN8: 7.66ms	HUB2-VPN8: 4.08ms		
HUB1_HC	10.100.100.100	HUB1-VPN1: 0.00%	HUB1-VPN1: 5.96ms	HUB1-VPN1: 1.66ms	5	5
		HUB1-VPN2: 0.00%	HUB1-VPN2: 3.23ms	HUB1-VPN2: 1.41ms		
		HUB1-VPN3: 0.00%	HUB1-VPN3: 4.49ms	HUB1-VPN3: 2.02ms		
		HUB1-VPN4: 0.00%	HUB1-VPN4: 5.09ms	HUB1-VPN4: 2.35ms		

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

```

1 config router route-map
2   edit "IN-SLA-TUN5"
3     config rule
4       edit 1
5         set set-community "65001:5"
6       next
7     end
8   next
9   edit "IN-SLA-TUN6"
10    config rule
11      edit 1
12        set set-community "65001:6"
13      next
14    end
15  next
16  edit "IN-SLA-TUN7"
17    config rule
18      edit 1
19        set set-community "65001:7"
20      next
21    end
22  next
23  edit "IN-SLA-TUN8"
24    config rule
25      edit 1
26        set set-community "65001:8"
27      next
28    end
29  end
30 end
31 config router bgp
32   config neighbor
33     edit "10.50.5.1"
34       set advertisement-interval 1
35       set capability-graceful-restart enable
36       set link-down-failover enable
37       set soft-reconfiguration enable
38       set description "HUB2-VPNS"
39       set interface "HUB2-VPNS"
40       set remote-as 65001
41       set route-map-out "OUT-OF-SLA"
42       set connect-timer 10
43       set additional-path receive
44       set route-map-out-preferable "IN-SLA-TUN5"
45     next
46     edit "10.50.6.1"
47       set advertisement-interval 1
48       set capability-graceful-restart enable
49       set link-down-failover enable
50       set soft-reconfiguration enable
51       set description "HUB2-VPN6"
52       set interface "HUB2-VPN6"
53       set remote-as 65001
54       set route-map-out "OUT-OF-SLA"
55       set connect-timer 10
56       set additional-path receive
57       set route-map-out-preferable "IN-SLA-TUN6"
58     next
59     edit "10.50.7.1"
60       set advertisement-interval 1
61       set capability-graceful-restart enable
62       set link-down-failover enable
63       set soft-reconfiguration enable
64       set description "HUB2-VPN7"
65       set interface "HUB2-VPN7"
66       set remote-as 65001
67       set route-map-out "OUT-OF-SLA"
68       set connect-timer 10
69       set additional-path receive
70       set route-map-out-preferable "IN-SLA-TUN7"
71     next
72     edit "10.50.8.1"
73       set advertisement-interval 1
74       set capability-graceful-restart enable
75       set link-down-failover enable
76       set soft-reconfiguration enable
77       set description "HUB2-VPNS"
78       set interface "HUB2-VPNS"
79       set remote-as 65001
80       set route-map-out "OUT-OF-SLA"
81       set connect-timer 10
82       set additional-path receive
83       set route-map-out-preferable "IN-SLA-TUN8"
84     next

```

IP	Remote AS
<input type="checkbox"/> 10.50.1.1	65001
<input type="checkbox"/> 10.50.2.1	65001
<input type="checkbox"/> 10.50.3.1	65001
<input type="checkbox"/> 10.50.4.1	65001
<input type="checkbox"/> 10.50.5.1	65001
<input type="checkbox"/> 10.50.6.1	65001
<input type="checkbox"/> 10.50.7.1	65001
<input type="checkbox"/> 10.50.8.1	65001

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

The screenshot displays the Fortinet SD-WAN configuration interface. The top section shows the 'Device Manager' sidebar with 'BGP Templates' selected. A table lists existing templates: HUB_BGP_Recommended, BRANCH_BGP_Recommended, BRANCH_BGP, and HUB1_BGP. A 'More' menu is open over HUB1_BGP, showing 'Clone', 'Import', and 'Activate' options.

The 'Clone BGP Template' dialog is shown below, with the following fields:

- Name: HUB2_BGP
- Description: (empty)
- Local As: 65001
- Router ID: 10.100.100.200
- Neighbors: (empty)

The 'Edit Route Map' dialog is shown below, with the following fields:

- Name: Route-Map-In-2
- Comments: (empty)

The 'Rules' section contains a table with the following data:

ID	Action	Interface	Match IP Rules	Match Next Hop Rules
1	Permit			
2	Permit			
3	Permit			
4	Permit			
11	Permit			

HUB2 BGP Route-Maps Configuration

The screenshots show the configuration of four BGP Route-Maps for HUB2:

- Route Map 65001:5:** Match community 65001:5, Set route tag 5.
- Route Map 65001:6:** Match community 65001:6, Set route tag 6.
- Route Map 65001:7:** Match community 65001:7, Set route tag 7.
- Route Map 65001:8:** Match community 65001:8, Set route tag 8.

The 'Community list rule edit' section for each map shows:

- ID: 1
- Action: Deny
- Match: 65001:5 (or 6, 7, 8)

HUB2 BGP Configuration

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

The left screenshot shows the 'Edit BGP Template' window:

- Neighbor Group:**

Name	Remote AS
VPN5	65001
VPN6	65001
VPN7	65001
VPN8	65001
- Neighbor Ranges:**

Prefix	Neighbor Group	Maximum Neighbor Number
10.50.5.0/255.255.255.0	VPN5	0
10.50.6.0/255.255.255.0	VPN6	0
10.50.7.0/255.255.255.0	VPN7	0
10.50.8.0/255.255.255.0	VPN8	0

The right screenshot shows the 'Device Manager' window with the following provisioning templates:

Template Group Name	Provisioning Templates	Assigned to Device/Group
HUB1-TG	HUB1_IPsec, HUB1_BGP, HUB1-SDWAN, HUB-CLI	1 Device in Total HUB1 [root]
BRANCHES-TG	BRANCH_IPsec, BRANCH_BGP, BRANCH_SDWAN, BR_LOOP	2 Devices in Total Branches (2)
HUB2-TG	HUB2_IPsec, HUB2_BGP, HUB2-SDWAN, HUB-CLI	1 Device in Total HUB2 [root]

HUB2 BGP Configuration Preview

```

1 config router community-list
2   edit "65001:11"
3     config rule
4       edit 1
5         set action permit
6         set match "65001:11"
7       next
8     end
9   next
10  edit "65001:5"
11    config rule
12      edit 1
13        set action permit
14        set match "65001:5"
15      next
16    end
17  next
18  edit "65001:6"
19    config rule
20      edit 1
21        set action permit
22        set match "65001:6"
23      next
24    end
25  next
26  edit "65001:7"
27    config rule
28      edit 1
29        set action permit
30        set match "65001:7"
31      next
32    end
33  next
34  edit "65001:8"
35    config rule
36      edit 1
37        set action permit
38        set match "65001:8"
39      next
40    end
41  next
42 end

43 config router route-map
44   edit "Route-Map-In-2"
45     config rule
46       edit 1
47         set match-community "65001:5"
48         set set-route-tag "5"
49       next
50     edit 2
51       set match-community "65001:6"
52       set set-route-tag "6"
53     next
54     edit 3
55       set match-community "65001:7"
56       set set-route-tag "7"
57     next
58     edit 4
59       set match-community "65001:8"
60       set set-route-tag "8"
61     next
62     edit 11
63       set match-community "65001:11"
64       set set-route-tag "11"
65     next
66   end
67 next
68 end

69 config router bgp
70   set as 65001
71   set router-id 10.100.100.200
72   set ibgp-multipath enable
73   set additional-path enable
74   set graceful-restart enable
75   set additional-path-select 255
76   config neighbor-group
77     edit "VPNs"
78       set capability-graceful-restart enable
79       set link-down-failover enable
80       set next-hop-self enable
81       set soft-reconfiguration enable
82       set remote-as 65001
83       set route-map-in "Route-Map-In-2"
84       set additional-path send

85       set route-map-in "Route-Map-In-2"
86       set additional-path send
87       set route-reflector-client enable
88     next
89     edit "VPN6"
90       set capability-graceful-restart enable
91       set link-down-failover enable
92       set next-hop-self enable
93       set soft-reconfiguration enable
94       set remote-as 65001
95       set route-map-in "Route-Map-In-2"
96       set additional-path send
97       set route-reflector-client enable
98     next
99     edit "VPN7"
100      set capability-graceful-restart enable
101      set link-down-failover enable
102      set next-hop-self enable
103      set soft-reconfiguration enable
104      set remote-as 65001
105      set route-map-in "Route-Map-In-2"
106      set additional-path send
107      set route-reflector-client enable
108    next
109    edit "VPN8"
110      set capability-graceful-restart enable
111      set link-down-failover enable
112      set next-hop-self enable
113      set soft-reconfiguration enable
114      set remote-as 65001
115      set route-map-in "Route-Map-In-2"
116      set additional-path send
117      set route-reflector-client enable
118    end
119  config neighbor-range
120    edit 1
121      set prefix 10.50.5.0 255.255.255.0
122      set neighbor-group "VPNs"
123    next
124    edit 2
125      set prefix 10.50.6.0 255.255.255.0

126  next
127  edit 3
128    set prefix 10.50.7.0 255.255.255.0
129    set neighbor-group "VPN7"
130  next
131  edit 4
132    set prefix 10.50.8.0 255.255.255.0
133    set neighbor-group "VPNs"
134  next
135  end
136 config network
137   edit 1
138     set prefix 10.10.200.0 255.255.255.0
139   next
140 end
141 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.

Install Preview of BR1

Assigned Devices

BR1

BR1 x BR2 x

Search...

```

1 config system sdwan
2   config neighbor
3     edit "10.50.5.1"
4       set health-check "HUB2_HC"
5       set sla-id 1
6       set member 7
7     next
8     edit "10.50.6.1"
9       set health-check "HUB2_HC"
10      set sla-id 1
11      set member 8
12    next
13    edit "10.50.7.1"
14      set health-check "HUB2_HC"
15      set sla-id 1
16      set member 9
17    next
18    edit "10.50.8.1"
19      set health-check "HUB2_HC"
20      set sla-id 1
21      set member 10
22    next
23  end
24 end

```

Neighbor

Neighbor	Role	Interface Member	Performance SLA	SLA
<input type="checkbox"/> 10.50.3.1	Standalone	HUB1-VPN3	HUB_HC	1
<input type="checkbox"/> 10.50.4.1	Standalone	HUB1-VPN4	HUB_HC	1
<input type="checkbox"/> 10.50.5.1	Standalone	HUB2-VPN5	HUB2_HC	1
<input type="checkbox"/> 10.50.6.1	Standalone	HUB2-VPN6	HUB2_HC	1
<input type="checkbox"/> 10.50.7.1	Standalone	HUB2-VPN7	HUB2_HC	1
<input type="checkbox"/> 10.50.8.1	Standalone	HUB2-VPN8	HUB2_HC	1

98% 8

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

Edit SD-WAN Template CLI Configurations x

0% 9

SD-WAN Rules

+ Create New Edit Delete More Search...

ID	Name	Source	Destination	Criteria	Members	Status	Performance SLA	Port	F	+
1	TUN5	Data-Networks	Route Tag: 5		VPN5	Enable				any
2	TUN6	Data-Networks	Route Tag: 6		VPN6	Enable				any
3	TUN7	Data-Networks	Route Tag: 7		VPN7	Enable				any
4	TUN8	Data-Networks	Route Tag: 8		VPN8	Enable				any
	sd-wan	ALL	ALL	Source IP	ALL					any

5

```

1 config firewall address
2   edit "Data-Networks"
3     set uuid 53bf5a3e-9541-51ef-5bd8-1ca0c2e2e343
4     set subnet 10.1.0.0 255.255.0.0
5
6   next
7 end
8 config system sdwan
9   config service
10    edit 1
11      set name "TUN5"
12      set route-tag 5
13      set src "Data-Networks"
14      set priority-members 3
15
16    next
17    edit 2
18      set name "TUN6"
19      set route-tag 6
20      set src "Data-Networks"
21      set priority-members 4
22
23    next
24    edit 3
25      set name "TUN7"
26      set route-tag 7
27      set src "Data-Networks"
28      set priority-members 5
29
30    next
31    edit 4
32      set name "TUN8"
33      set route-tag 8
34      set src "Data-Networks"
35      set priority-members 6
36
37    next
38  end
39 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.

```

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
8 BGP community entries

Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
10.50.1.1 4 65001 48 25 2 0 0 00:10:23 5
10.50.2.1 4 65001 48 25 2 0 0 00:10:22 5
10.50.3.1 4 65001 49 26 2 0 0 00:10:23 5
10.50.4.1 4 65001 52 25 2 0 0 00:10:22 5
10.50.5.1 4 65001 25 20 4 0 0 00:04:28 3
10.50.6.1 4 65001 28 20 4 0 0 00:04:17 3
10.50.7.1 4 65001 32 20 4 0 0 00:04:18 3
10.50.8.1 4 65001 30 21 4 0 0 00:04:25 3

Total number of neighbors 8

BR1 # get router info routing-table bgp
Routing table for 10.10.200.0/24
0
10.10.2.0/24 [200/0] via 10.50.1.10 [4] (recursive is directly connected, HUB1-VPN1), 00:03:23, [1/0]
[200/0] via 10.50.2.10 [4] (recursive is directly connected, HUB1-VPN2), 00:03:23, [1/0]
[200/0] via 10.50.3.10 [4] (recursive is directly connected, HUB1-VPN3), 00:03:23, [1/0]
[200/0] via 10.50.4.10 [4] (recursive is directly connected, HUB1-VPN4), 00:03:23, [1/0]
[200/0] via 10.50.5.11 [4] (recursive is directly connected, HUB2-VPN5), 00:03:23, [1/0]
[200/0] via 10.50.6.10 [4] (recursive is directly connected, HUB2-VPN6), 00:03:23, [1/0]
10.10.200.0/24 [200/0] via 10.50.2.1 (recursive via HUB1-VPN2 tunnel 200.2.100.1), 00:09:56, [1/0]
[200/0] via 10.50.3.1 (recursive via HUB1-VPN3 tunnel 10.0.0.1), 00:09:56, [1/0]
[200/0] via 10.50.4.1 (recursive via HUB1-VPN4 tunnel 10.0.0.2), 00:09:56, [1/0]
10.10.200.0/24 [200/0] via 10.50.5.1 (recursive via HUB2-VPN5 tunnel 100.1.200.1), 00:03:52, [1/0]
[200/0] via 10.50.6.1 (recursive via HUB2-VPN6 tunnel 200.2.200.1), 00:03:52, [1/0]
[200/0] via 10.50.7.1 (recursive via HUB2-VPN7 tunnel 10.0.0.3), 00:03:52, [1/0]
[200/0] via 10.50.8.1 (recursive via HUB2-VPN8 tunnel 10.0.0.4), 00:03:52, [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 4
1 BGP AS-PATH entries
8 BGP community entries

Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
10.50.1.1 4 65001 49 28 1 0 0 00:10:22 5
10.50.2.1 4 65001 52 27 1 0 0 00:10:22 5
10.50.3.1 4 65001 48 25 1 0 0 00:10:24 5
10.50.4.1 4 65001 51 25 1 0 0 00:10:22 5
10.50.5.1 4 65001 26 23 3 0 0 00:04:43 3
10.50.6.1 4 65001 26 23 3 0 0 00:04:44 3
10.50.7.1 4 65001 30 23 3 0 0 00:04:41 3
10.50.8.1 4 65001 30 23 3 0 0 00:04:38 3

Total number of neighbors 8

BR2 # get router info routing-table bgp
Routing table for 10.10.1.0/24
0
10.10.1.0/24 [200/0] via 10.50.1.11 [4] (recursive is directly connected, HUB1-VPN1), 00:03:44, [1/0]
[200/0] via 10.50.2.11 [4] (recursive is directly connected, HUB1-VPN2), 00:03:44, [1/0]
[200/0] via 10.50.3.11 [4] (recursive is directly connected, HUB1-VPN3), 00:03:44, [1/0]
[200/0] via 10.50.4.11 [4] (recursive is directly connected, HUB2-VPN4), 00:03:44, [1/0]
[200/0] via 10.50.5.10 [4] (recursive is directly connected, HUB2-VPN5), 00:03:44, [1/0]
[200/0] via 10.50.6.11 [4] (recursive is directly connected, HUB2-VPN6), 00:03:44, [1/0]
10.10.100.0/24 [200/0] via 10.50.2.1 (recursive via HUB1-VPN2 tunnel 200.2.100.1), 00:09:56, [1/0]
[200/0] via 10.50.2.1 (recursive via HUB1-VPN2 tunnel 200.2.100.1), 00:09:56, [1/0]
[200/0] via 10.50.3.1 (recursive via HUB1-VPN3 tunnel 10.0.0.1), 00:09:56, [1/0]
[200/0] via 10.50.4.1 (recursive via HUB1-VPN4 tunnel 10.0.0.2), 00:09:56, [1/0]
10.10.200.0/24 [200/0] via 10.50.5.1 (recursive via HUB2-VPN5 tunnel 100.1.200.1), 00:04:13, [1/0]
[200/0] via 10.50.6.1 (recursive via HUB2-VPN6 tunnel 200.2.200.1), 00:04:13, [1/0]
[200/0] via 10.50.7.1 (recursive via HUB2-VPN7 tunnel 10.0.0.3), 00:04:13, [1/0]
[200/0] via 10.50.8.1 (recursive via HUB2-VPN8 tunnel 10.0.0.4), 00:04:13, [1/0]

```

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.

Install Preview of HUB2

Assigned Devices: HUB2

```

1 config router bgp
2     config neighbor-group
3         edit "VPNS5"
4             set adv-additional-path 4
5         next
6         edit "VPN6"
7             set adv-additional-path 4
8         next
9         edit "VPN7"
10            set adv-additional-path 4
11        next
12        edit "VPN8"
13            set adv-additional-path 4
14        next
15    end
16 end
17

```

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.

The screenshots show the Firewall Policy configuration for BR1 and BR2. The policies are configured as follows:

Name	From	To	Source	Destination	Schedule	Service	Action	NAT	Security Profiles	Log	Bytes	Type
LAN-TO-HQ&BRANCHES	port3	overlay	all	all	always	ALL	ACCEPT	Disabled	no-inspection	All	2.02 kB	Standard
HQ&BRANCHES-TO-LAN	overlay	port3	all	all	always	ALL	ACCEPT	Disabled	no-inspection	All	0 B	Standard
DIA	port3	DIA	all	all	always	ALL	ACCEPT	Enabled	no-inspection	All	588 B	Standard
Implicit Deny	any	any	all	all	always	ALL	DENY			Disabled	1.26 kB	

ADVPN with 8 Tunnels

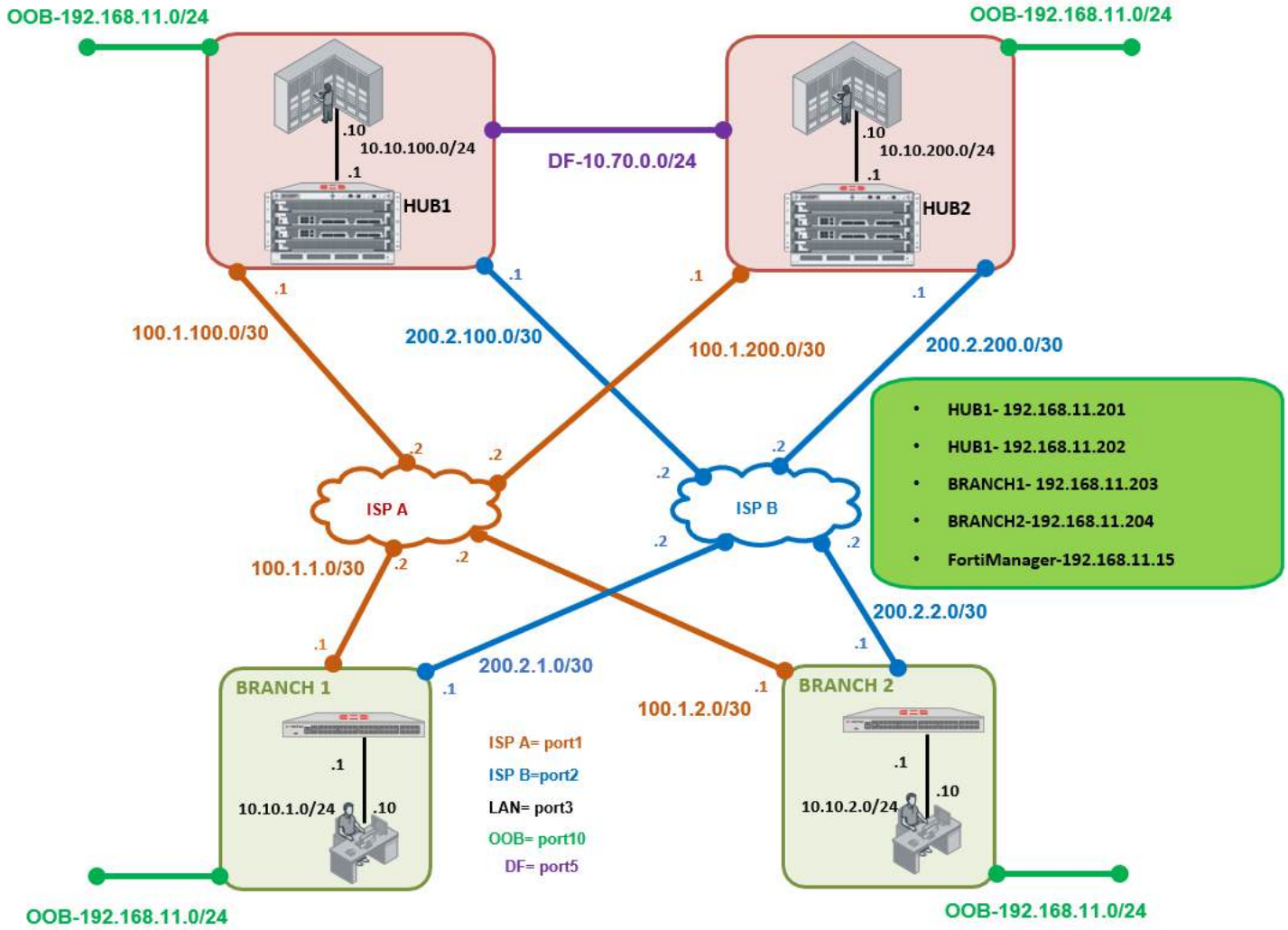
The configurations done at IPSec and BGP levels were ADVPN aware, branch to branch traffic will activate an ADVPN session, in the case shown below over HUB2-VPN6.

The screenshots show the IPsec Monitor configuration for BR1 and BR2. The active sessions are as follows:

Name	Remote Gateway	Peer ID	Incoming Data	Outgoing Data	Phase 1	Phase 2 Selectors
HUB2-VPN6_0	200.2.2.1	Mombasa_Tun6	876 B	756 B	HUB2-VPN6_0	HUB2-VPN6 HUB2-VPN6
HUB2-VPN7	100.1.200.1	Thika_Tun7	170.23 kB	169.24 kB	HUB2-VPN7	HUB2-VPN7
HUB2-VPN8	200.2.200.1	Thika_Tun8	170.27 kB	169.47 kB	HUB2-VPN8	HUB2-VPN8
HUB2-VPN5	100.1.200.1	Thika_Tun5	170.89 kB	169.98 kB	HUB2-VPN5	HUB2-VPN5
HUB2-VPN6	200.2.200.1	Thika_Tun6	171.04 kB	170.03 kB	HUB2-VPN6	HUB2-VPN6
HUB1-VPN1	100.1.100.1	Nairobi_Tun1	219.60 kB	217.52 kB	HUB1-VPN1	HUB1-VPN1
HUB1-VPN3	100.1.100.1	Nairobi_Tun3	219.34 kB	217.56 kB	HUB1-VPN3	HUB1-VPN3
HUB1-VPN4	200.2.100.1	Nairobi_Tun4	220.06 kB	217.78 kB	HUB1-VPN4	HUB1-VPN4
HUB1-VPN2	200.2.100.1	Nairobi_Tun2	219.82 kB	217.90 kB	HUB1-VPN2	HUB1-VPN2

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.

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
10.10.200.10/32	10.70.0.200	port5	Enabled

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
10.10.100.10/32	10.70.0.100	port5	Enabled

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

Name	From	To	Source	Destination	Schedule	Service	Action	NAT
BRANCHES-FMG	DIA	port10	all	PRI-FMG SEC-FMG	always	ALL	ACCEPT	Disabled
BRANCHES-TO-HQ	overlay	port3	all	all	always	ALL	ACCEPT	Disabled
HQ-TO-BRANCHES	port3	overlay	all	all	always	ALL	ACCEPT	Disabled
BRANCH-TO-BRANCH	overlay	overlay	all	all	always	ALL	ACCEPT	Disabled
BRANCHES-TO-LOOPBACK	overlay	HUB_LOOP	all	HUB1-LOOPBACK	always	ALL_ICMP	ACCEPT	Disabled
LAN-HUB2	port3	port5	all	Data-Networks	always	ALL	ACCEPT	Disabled
HUB2-LAN	port5	port3	Data-Networks	all	always	ALL	ACCEPT	Disabled
DIA	port3	DIA	all	all	always	ALL	ACCEPT	Enabled
Implicit Deny	any	any	all	all	always	ALL	DENY	

Name	From	To	Source	Destination	Schedule	Service	Action	NAT
BRANCHES-TO-HQ	overlay	port3	all	all	always	ALL	ACCEPT	Disabled
HQ-TO-BRANCHES	port3	overlay	all	all	always	ALL	ACCEPT	Disabled
BRANCH-TO-BRANCH	overlay	overlay	all	all	always	ALL	ACCEPT	Disabled
BRANCHES-TO-LOOPBACK	overlay	HUB_LOOP	all	HUB2-LOOPBACK	always	ALL_ICMP	ACCEPT	Disabled
LAN-HUB1	port3	port5	all	Data-Networks	always	ALL	ACCEPT	Disabled
HUB1-LAN	port5	port3	Data-Networks	all	always	ALL	ACCEPT	Disabled
DIA	port3	DIA	all	all	always	ALL	ACCEPT	Enabled
Implicit Deny	any	any	all	all	always	ALL	DENY	

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.

This screenshot shows the configuration for the first IPsec tunnel, HUB1-HUB2-VPN1, on HUB1. The configuration includes:

- Name:** HUB1-HUB2-VPN1
- Comments:** VPN: HUB1-HUB2-VPN1 (Created by VPN wizard) 43/255
- Network:** Remote Gateway: Static IP Address (100.1.200.1), Interface: port1
- Authentication:** Authentication Method: Pre-shared Key, IKE Version: 1, Mode: Main (ID protection)
- Phase 1 Proposal:** Algorithms: AES128-SHA256, AES256-SHA256, AES128-SHA1, AES256-SHA1; Diffie-Hellman Groups: 14, 5
- XAUTH:** Type: Disabled
- Phase 2 Selectors:**

Name	Local Address	Remote Address	Add
HUB1-HUB2-VPN1	10.10.100.0/255.255.2	10.10.200.0/255.255.2	<input type="checkbox"/>

This screenshot shows the configuration for the second IPsec tunnel, HUB1-HUB2-VPN2, on HUB1. The configuration includes:

- Name:** HUB1-HUB2-VPN2
- Comments:** VPN: HUB1-HUB2-VPN2 (Created by VPN wizard) 43/255
- Network:** Remote Gateway: Static IP Address (200.2.200.1), Interface: port2
- Authentication:** Authentication Method: Pre-shared Key, IKE Version: 1, Mode: Main (ID protection)
- Phase 1 Proposal:** Algorithms: AES128-SHA256, AES256-SHA256, AES128-SHA1, AES256-SHA1; Diffie-Hellman Groups: 14, 5
- XAUTH:** Type: Disabled
- Phase 2 Selectors:**

Name	Local Address	Remote Address	Add
HUB1-HUB2-VPN2	10.10.100.0/255.255.2	10.10.200.0/255.255.2	<input type="checkbox"/>

This screenshot shows the configuration for the first IPsec tunnel, HUB2-HUB1-VPN1, on HUB2. The configuration includes:

- Name:** HUB2-HUB1-VPN1
- Comments:** VPN: HUB2-HUB1-VPN1 (Created by VPN wizard) 43/255
- Network:** Remote Gateway: Static IP Address (100.1.100.1), Interface: port1
- Authentication:** Authentication Method: Pre-shared Key, IKE Version: 1, Mode: Main (ID protection)
- Phase 1 Proposal:** Algorithms: AES128-SHA256, AES256-SHA256, AES128-SHA1, AES256-SHA1; Diffie-Hellman Groups: 14, 5
- XAUTH:** Type: Disabled
- Phase 2 Selectors:**

Name	Local Address	Remote Address	Add
HUB2-HUB1-VPN1	10.10.200.0/255.255.2	10.10.100.0/255.255.2	<input type="checkbox"/>

This screenshot shows the configuration for the second IPsec tunnel, HUB2-HUB1-VPN2, on HUB2. The configuration includes:

- Name:** HUB2-HUB1-VPN2
- Comments:** VPN: HUB2-HUB1-VPN2 (Created by VPN wizard) 43/255
- Network:** Remote Gateway: Static IP Address (200.2.100.1), Interface: port2
- Authentication:** Authentication Method: Pre-shared Key, IKE Version: 1, Mode: Main (ID protection)
- Phase 1 Proposal:** Algorithms: AES128-SHA256, AES256-SHA256, AES128-SHA1, AES256-SHA1; Diffie-Hellman Groups: 14, 5
- XAUTH:** Type: Disabled
- Phase 2 Selectors:**

Name	Local Address	Remote Address	Add
HUB2-HUB1-VPN2	10.10.200.0/255.255.2	10.10.100.0/255.255.2	<input type="checkbox"/>

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

Tunnel	Interface Binding	Status	Connections
HUB1-HUB2-VPN1	port1	Up	4
HUB1-HUB2-VPN2	port2	Up	4
VPN1	port1	2 dialup connection(s)	2
VPN2	port2	2 dialup connection(s)	2
VPN3	port1	2 dialup connection(s)	2
VPN4	port2	2 dialup connection(s)	2

Tunnel	Interface Binding	Status	Connections
HUB2-HUB1-VPN1	port1	Up	4
HUB2-HUB1-VPN2	port2	Up	4
VPN5	port1	2 dialup connection(s)	2
VPN6	port2	2 dialup connection(s)	2
VPN7	port1	2 dialup connection(s)	2
VPN8	port2	2 dialup connection(s)	2

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

Name	From	To	Source	Destination	Schedule	Service	Action	NAT
BRANCHES-FMG	DIA	port10	all	PRI-FMG SEC-FMG	always	ALL	ACCEPT	Disabled
BRANCHES-TO-HQ	overlay	port3	all	all	always	ALL	ACCEPT	Disabled
HQ-TO-BRANCHES	port3	overlay	all	all	always	ALL	ACCEPT	Disabled
BRANCH-TO-BRANCH	overlay	overlay	all	all	always	ALL	ACCEPT	Disabled
BRANCHES-TO-LOOPBACK	overlay	HUB_LOOP	all	HUB1-LOOPBACK	always	ALL_ICMP	ACCEPT	Disabled
LAN-HUB2	port3	port5	all	Data-Networks	always	ALL	ACCEPT	Disabled
HUB2-LAN	port5	port3	Data-Networks	all	always	ALL	ACCEPT	Disabled
DIA	port3	DIA	all	all	always	ALL	ACCEPT	Enabled
vpn_HUB1-HUB2-VPN1_local_0	port3	HUB1-HUB2-VPN1	HUB1-HUB2-VPN1_local	HUB1-HUB2-VPN1_remote	always	ALL	ACCEPT	Disabled
vpn_HUB1-HUB2-VPN1_remote_0	HUB1-HUB2-VPN1	port3	HUB1-HUB2-VPN1_remote	HUB1-HUB2-VPN1_local	always	ALL	ACCEPT	Disabled
vpn_HUB1-HUB2-VPN2_local_0	port3	HUB1-HUB2-VPN2	HUB1-HUB2-VPN2_local	HUB1-HUB2-VPN2_remote	always	ALL	ACCEPT	Disabled
vpn_HUB1-HUB2-VPN2_remote_0	HUB1-HUB2-VPN2	port3	HUB1-HUB2-VPN2_remote	HUB1-HUB2-VPN2_local	always	ALL	ACCEPT	Disabled
Implicit Deny	any	any	all	all	always	ALL	DENY	

Name	From	To	Source	Destination	Schedule	Service	Action	NAT
BRANCHES-TO-HQ	overlay	port3	all	all	always	ALL	ACCEPT	Disabled
HQ-TO-BRANCHES	port3	overlay	all	all	always	ALL	ACCEPT	Disabled
BRANCH-TO-BRANCH	overlay	overlay	all	all	always	ALL	ACCEPT	Disabled
BRANCHES-TO-LOOPBACK	overlay	HUB_LOOP	all	HUB2-LOOPBACK	always	ALL_ICMP	ACCEPT	Disabled
LAN-HUB1	port3	port5	all	Data-Networks	always	ALL	ACCEPT	Disabled
HUB1-LAN	port5	port3	Data-Networks	all	always	ALL	ACCEPT	Disabled
DIA	port3	DIA	all	all	always	ALL	ACCEPT	Enabled
vpn_HUB2-HUB1-VPN1_local_0	port3	HUB2-HUB1-VPN1	HUB2-HUB1-VPN1_local	HUB2-HUB1-VPN1_remote	always	ALL	ACCEPT	Disabled
vpn_HUB2-HUB1-VPN1_remote_0	HUB2-HUB1-VPN1	port3	HUB2-HUB1-VPN1_remote	HUB2-HUB1-VPN1_local	always	ALL	ACCEPT	Disabled
vpn_HUB2-HUB1-VPN2_local_0	port3	HUB2-HUB1-VPN2	HUB2-HUB1-VPN2_local	HUB2-HUB1-VPN2_remote	always	ALL	ACCEPT	Disabled
vpn_HUB2-HUB1-VPN2_remote_0	HUB2-HUB1-VPN2	port3	HUB2-HUB1-VPN2_remote	HUB2-HUB1-VPN2_local	always	ALL	ACCEPT	Disabled
Implicit Deny	any	any	all	all	always	ALL	DENY	

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

HUB1			
+ Create New Edit > Edit in CLI Clone Delete <input type="text" value="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			
+ Create New Edit > Edit in CLI Clone Delete <input type="text" value="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

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.

Device Manager - Install Wizard

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

Name: Single-HUB

Description:

Select New Topology:

- Single HUB
- Dual HUB (Primary & Secondary)
- Dual HUB (Primary & Primary)

Advanced

Loopback IP Address: 10.100.100.0/255.255.255.0

Overlay Network: 10.50.0.0/255.255.0.0

BGP-AS Number: 65001

Auto-Discovery VPN:

Next > Cancel

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

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

Name: Single-HUB

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

HUB

Standalone HUB: HUB1

Branch

Device Group Assignment: Branches

< Back Next > Cancel

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 them once the wizard is complete.

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

Name: Single-HUB

HUB

Standalone HUB: HUB1

Underlay	#	Private Link	Override IP	Action
WAN Underlay 1		<input checked="" type="checkbox"/> port1	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
WAN Underlay 2		<input checked="" type="checkbox"/> port2	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>

Network Advertisement: Connected Static

#	Interface	Action
Interface 1	port3	<input type="checkbox"/> <input type="checkbox"/>

Advanced >

Branch Route Maps

Route map in: Route-Map-In

Route map out:

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

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

Branch

Branch Device Group: Branches

Underlay	#	Private Link	Action
WAN Underlay 1		<input checked="" type="checkbox"/> port1	<input type="checkbox"/> <input type="checkbox"/>
WAN Underlay 2		<input checked="" type="checkbox"/> port2	<input type="checkbox"/> <input type="checkbox"/>

Network Advertisement: Connected Static

#	Interface	Action
Interface 1	port3	<input type="checkbox"/> <input type="checkbox"/>

Advanced ▾

Apply to All / Specify: Apply to All Specify

HUB 1 Overlay 1

Route map in:

Route map out: OUT-OF-SLA

Route map out preferable: IN-SLA-TUN1

HUB 1 Overlay 2

Route map in:

Route map out: OUT-OF-SLA

Route map out preferable: IN-SLA-TUN2

< Back Next > Cancel

Finally add overlay interfaces and zones.

Create New SD-WAN Overlay Template - SD-WAN Template Options (4/5) ✕

Add Overlay Objects to SD-WAN Template	<input checked="" type="checkbox"/>
Add Overlay Interfaces and Zones	<input checked="" type="checkbox"/>
Add Health Check Servers for Each HUB as Performance SLA	<input type="checkbox"/>

The summary of the settings done is displayed below:

Create New SD-WAN Overlay Template - Summary (5/5) ✕

Please review the summary of SD-WAN Overlay configurations

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

Template Name	Single-HUB
Topology	Single HUB

Region Network Settings ▾

Loopback Allocated	10.100.100.0/255.255.255.0
Overlay Network	10.50.0.0/255.255.0.0
BGP AS Number	65001
Auto-Discovery VPN	<input checked="" type="checkbox"/>

Device Assignment ▾

Standalone HUB	HUB1 (192.168.11.201, Platform: FortiGate-VM64-KVM)
Branch	☰ Branches

Underlay Assignment ▾

Standalone HUB Underlays	Underlay 1: port1 Underlay 2: port2
Branch Underlays	Underlay 1: port1 Underlay 2: port2

Network Advertisement ▾

Standalone HUB	Connected Interface 1: port3
Branch	Connected Interface 1: port3

SD-WAN Template Options ▾

Add Overlay Objects to SD-WAN Template	<input type="checkbox"/>
Add Overlay Interfaces and Zones	<input checked="" type="checkbox"/>
Add Health Check Servers for Each HUB as Performance SLA	<input type="checkbox"/>

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

#	Template Name	Topology	Assign to Device/Group	Loopback IP Address	Overlay Network	BGP AS Number
1	Single-HUB	Single HUB	HUB1 Branches	10.100.100.0/255.255.255...	10.50.0.0/255.255.0.0	65001

Name	Assigned to Device/Group
HUB_IPsec_Recommended	0 Devices in Total
BRANCH_IPsec_Recommended	0 Devices in Total
IPsec_Fortinet_Recommended	0 Devices in Total
Single-HUB_BRANCH_IPsec	0 Devices in Total
Single-HUB_HUB1_IPsec	0 Devices in Total

Name	Assigned to Device/Group
HUB_BGP_Recommended	0 Devices in Total
BRANCH_BGP_Recommended	0 Devices in Total
Single-HUB_BRANCH_BGP	0 Devices in Total
Single-HUB_HUB1_BGP	0 Devices in Total

It also creates Template Groups and CLI templates:

Template Group Name	Provisioning Templates	Assigned to Device/Group
Single-HUB_HUB1	Single-HUB_HUB1_IPsec Single-HUB_HUB1_BGP Single-HUB_HUB1_CLIGRP	1 Device in Total View Details > HUB1 [root]
Single-HUB_BRANCH	Single-HUB_BRANCH_IPsec Single-HUB_BRANCH_BGP Single-HUB_BRANCH_CLIGRP	2 Devices in Total View Details > Branches (2)

Name	Type	Assigned to Device/Group	Variables	Description	Members
Pre-Run CLI Template (3)					
provision_interfaces_on_vm	Jinja	0 Devices in Total	vm_interface_number	predefined script fo...	
split_hardware_switch_ports_40...	CLI	0 Devices in Total		predefined script fo...	
split_hardware_switch_ports_60...	CLI	0 Devices in Total		predefined script fo...	
CLI Template (2)					
Single-HUB_BRANCH_CLI	Jinja	0 Devices in Total	branch_id		
Single-HUB_HUB1_CLI	CLI	0 Devices in Total			
CLI Template Group (2)					
Single-HUB_BRANCH_CLIGRP	CLI/Jinja	0 Devices in Total			Single-HUB_BRANCH_CLI
Single-HUB_HUB1_CLIGRP	CLI/Jinja	0 Devices in Total			Single-HUB_HUB1_CLI

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

Name	Default Value	Description	Created Time
branch_id			admin / 2024-11-05 06:01:37
vm_interface_number	1	predefined variable for set fg	admin / 2024-10-13 19:05:26

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

Edit IPsec Tunnel - HUB1-VPN1

Tunnel Name: HUB1-VPN1

Network: [Empty]

Routing: **Manual** Automatic

Remote Device: **IP Address** Dynamic DNS Dynamic

Remote Gateway (IP Address): 100.1.100.1

Outgoing Interface: port1

Local ID: Branch\$(branch_id)

Network Overlay:

Network ID: 1

Proposal: aes256-sha256

FEC Health Check: [Empty]

Authentication Method: **Pre-shared Key** Signature

Pre-shared Key: [Redacted]

Tunnel Interface Setup

IP: [Empty]

Remote IP: [Empty]

Phase2 Interface

Name	Keep Alive
HUB1-VPN1	<input checked="" type="checkbox"/>

Edit IPsec Tunnel - HUB1-VPN2

Tunnel Name: HUB1-VPN2

Network: [Empty]

Routing: **Manual** Automatic

Remote Device: **IP Address** Dynamic DNS Dynamic

Remote Gateway (IP Address): 200.2.100.1

Outgoing Interface: port2

Local ID: Branch\$(branch_id)

Network Overlay:

Network ID: 2

Proposal: aes256-sha256

FEC Health Check: [Empty]

Authentication Method: **Pre-shared Key** Signature

Pre-shared Key: [Redacted]

Tunnel Interface Setup

IP: [Empty]

Remote IP: [Empty]

Phase2 Interface

Name	Keep Alive
HUB1-VPN2	<input checked="" type="checkbox"/>

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

Edit IPsec Tunnel - VPN1

Tunnel Name: VPN1

Network

Routing: **Manual** Automatic

Remote Device: IP Address Dynamic DNS **Dynamic**

Outgoing Interface: port1

Local ID: [Search]

Network Overlay: [Off]

Network ID: 1

IPv4 Start IP: 10.50.0.1

IPv4 End IP: 10.50.63.252

IPv4 Netmask: 255.255.192.0

Proposal: aes256-sha256

FEC Health Check: [Search]

Authentication

Authentication Method: **Pre-shared Key** Signature

Pre-shared Key: [Redacted]

Tunnel Interface Setup

IP: 10.50.63.253/255.255.255.255

Remote IP: 10.50.63.254/255.255.192.0

Edit IPsec Tunnel - VPN1

Tunnel Name: VPN1

Network

Routing: **Manual** Automatic

Remote Device: IP Address Dynamic DNS **Dynamic**

Outgoing Interface: port1

Local ID: [Search]

Network Overlay: [Off]

Network ID: 1

IPv4 Start IP: 10.50.1.10

IPv4 End IP: 10.50.1.250

IPv4 Netmask: 255.255.255.0

Proposal: aes256-sha256

FEC Health Check: [Search]

Authentication

Authentication Method: **Pre-shared Key** Signature

Pre-shared Key: [Redacted]

Tunnel Interface Setup

IP: 10.50.1.1/255.255.255.255

Remote IP: 10.50.1.253/255.255.255.0

Edit IPsec Tunnel - VPN2

Tunnel Name: VPN2

Network

Routing: Manual Automatic

Remote Device: IP Address Dynamic DNS Dynamic

Outgoing Interface: port2

Local ID: [Search]

Network Overlay: [On]

Network ID: 2

IPv4 Start IP: 10.50.64.1

IPv4 End IP: 10.50.127.252

IPv4 Netmask: 255.255.192.0

Proposal: aes256-sha256

FEC Health Check: [Search]

Authentication Method: Pre-shared Key Signature

Pre-shared Key: [Masked]

Tunnel Interface Setup

IP: 10.50.127.253/255.255.255.255

Remote IP: 10.50.127.254/255.255.192.0

Edit IPsec Tunnel - VPN2

Tunnel Name: VPN2

Network

Routing: Manual Automatic

Remote Device: IP Address Dynamic DNS Dynamic

Outgoing Interface: port2

Local ID: [Search]

Network Overlay: [On]

Network ID: 2

IPv4 Start IP: 10.50.2.10

IPv4 End IP: 10.50.2.250

IPv4 Netmask: 255.255.255.0

Proposal: aes256-sha256

FEC Health Check: [Search]

Authentication Method: Pre-shared Key Signature

Pre-shared Key: [Masked]

Tunnel Interface Setup

IP: 10.50.2.1/255.255.255.255

Remote IP: 10.50.2.253/255.255.255.0

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.

Edit BGP Template

Name: Single-HUB_BRANCH_BGP

Description: [Empty]

Local As: 65001

Router ID: 10.100.100.\$(branch_id)

Neighbors

IP	Remote AS
10.50.127.253	65001
10.50.63.253	65001

IPv4 Redistribute

Connected: All Filter port3_only

RIP: [On]

OSPF: [On]

Static: [On]

ISIS: [On]

Edit BGP Template

Name: Single-HUB_BRANCH_BGP

Description: [Empty]

Local As: 65001

Router ID: 10.100.100.\$(branch_id)

Neighbors

IP	Remote AS
10.50.2.1	65001
10.50.1.1	65001

Networks

IP/Netmask	Action
10.10.\$(branch_id).0 255.255.255.0	[X] [Add]

IPv6 Networks

[Add]

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

Edit Route Map Rule		Edit Route Map Rule		Edit Route Map Rule	
ID	1	ID	1	ID	1
Action	Deny Permit	Action	Deny Permit	Action	Deny Permit
IP Address Rule Variables		IP Address Rule Variables		IP Address Rule Variables	
Match interface	<input type="checkbox"/>	Match interface	<input type="checkbox"/>	Match interface	<input type="checkbox"/>
Match IP address	<input type="checkbox"/>	Match IP address	<input type="checkbox"/>	Match IP address	<input type="checkbox"/>
Match IPv6 address	<input type="checkbox"/>	Match IPv6 address	<input type="checkbox"/>	Match IPv6 address	<input type="checkbox"/>
Match next hop router IP address	<input type="checkbox"/>	Match next hop router IP address	<input type="checkbox"/>	Match next hop router IP address	<input type="checkbox"/>
Match next hop router IPv6 address	<input type="checkbox"/>	Match next hop router IPv6 address	<input type="checkbox"/>	Match next hop router IPv6 address	<input type="checkbox"/>
Set next hop router IP address	<input type="text"/>	Set next hop router IP address	<input type="text"/>	Set next hop router IP address	<input type="text"/>
Set next hop router local IPv6 address	<input type="text"/>	Set next hop router local IPv6 address	<input type="text"/>	Set next hop router local IPv6 address	<input type="text"/>
AS Path Rule Variables		AS Path Rule Variables		AS Path Rule Variables	
Match AS path list	<input type="checkbox"/>	Match AS path list	<input type="checkbox"/>	Match AS path list	<input type="checkbox"/>
Set AS path	<input type="text"/>	Set AS path	<input type="text"/>	Set AS path	<input type="text"/>
Community Rule Variables		Community Rule Variables		Community Rule Variables	
Match community	<input type="checkbox"/>	Match community	<input type="checkbox"/>	Match community	<input type="checkbox"/>
Set community delete	<input type="checkbox"/>	Set community delete	<input type="checkbox"/>	Set community delete	<input type="checkbox"/>
Set Community <i>i</i>	<input checked="" type="checkbox"/> 65001:1	Set Community <i>i</i>	<input checked="" type="checkbox"/> 65001:2	Set Community <i>i</i>	<input checked="" type="checkbox"/> 65001:11
Set Community Additive	<input type="checkbox"/>	Set Community Additive	<input type="checkbox"/>	Set Community Additive	<input type="checkbox"/>

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

The image shows three screenshots of the Fortinet configuration interface, illustrating the customization of BGP templates and neighbor ranges at the HUB.

- Top Left Screenshot:** Shows the "Edit BGP Template" window. The "Name" is "Single-HUB_HUB1_BGP". The "Local As" is "65001". The "Router ID" is "10.100.100.253".
- Top Right Screenshot:** Shows the "Edit BGP Template" window. The "Name" is "Single-HUB_HUB1_BGP". The "Local As" is "65001". The "Router ID" is "10.100.100.\${branch_id}".
- Bottom Left Screenshot:** Shows the "Neighbor Ranges" table with two entries:

Prefix	Neighbor Group	Maximum Neighbor Number
10.50.0.0/255.255.192.0	VPN1	0
10.50.64.0/255.255.192.0	VPN2	0

 Below the table, the "Networks" section shows two entries:

IP/Netmask	Action
10.50.0.0/255.255.192.0	+
10.50.64.0/255.255.192.0	+
- Bottom Right Screenshot:** Shows the "Neighbor Ranges" table with two entries:

Prefix	Neighbor Group	Maximum Neighbor Number
10.50.1.0/255.255.255.0	VPN1	0
10.50.2.0/255.255.255.0	VPN2	0

 Below the table, the "Networks" section shows one entry:

IP/Netmask	Action
10.10.\${branch_id}.0/255.255.255.0	+

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

The image shows three screenshots of the Fortinet configuration interface, illustrating the configuration of route map rules at the HUB.

- Left Screenshot:** Shows the "Edit Route Map Rule" window. The "Match community" is "65001:1". The "Match community exact" is "Disable". The "Set route tag" is "1".
- Middle Screenshot:** Shows the "Edit Route Map Rule" window. The "Match community" is "65001:2". The "Match community exact" is "Disable". The "Set route tag" is "2".
- Right Screenshot:** Shows the "Edit Route Map Rule" window. The "Match community" is "65001:11". The "Match community exact" is "Disable". The "Set route tag" is "11".

The Metadata Variables are configured as shown below:

Edit Metadata Variables - branch_id

Name:

Description:

Default Value:

Per-Device Mapping

Mapped Device	Value
<input type="checkbox"/> BR1 [root]	1
<input type="checkbox"/> BR2 [root]	2
<input type="checkbox"/> HUB1 [root]	100

Edit Metadata Variables - Br_ISPA_gw

Name:

Description:

Default Value:

Per-Device Mapping

Mapped Device	Value
<input type="checkbox"/> BR1 [root]	100.1.1.2
<input type="checkbox"/> BR2 [root]	100.1.2.2

Edit Metadata Variables - Br_ISPB_gw

Name:

Description:

Default Value:

Per-Device Mapping

Mapped Device	Value
<input type="checkbox"/> BR1 [root]	200.2.1.2
<input type="checkbox"/> BR2 [root]	200.2.2.2

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

Edit SD-WAN Template CLI Configurations ✕

Name:

Description:

SD-WAN Status:

SD-WAN Zones

ID	Interface	Gateway	Cost	Priority	Status
<input type="checkbox"/> DIA					
<input type="checkbox"/> 1	port1	100.1.100.2	0	1	Enable
<input type="checkbox"/> 2	port2	200.2.100.2	0	1	Enable
<input type="checkbox"/> overlay					
<input type="checkbox"/> 3	VPN1	0.0.0.0	0	1	Enable
<input type="checkbox"/> 4	VPN2	0.0.0.0	0	1	Enable

96% 7

SD-WAN Rules

ID	Name	Source	Destination	Criteria	Members	Status	Performance SLA	Port	Prot
<input type="checkbox"/> 1	TUN1	all	Route Tag: 1		VPN1	Enable			any
<input type="checkbox"/> 2	TUN2	all	Route Tag: 2		VPN2	Enable			any
<input type="checkbox"/> 3	OUT-OF-SLA	ALL	Route Tag: 11		VPN1 VPN2	Enable			any
<input type="checkbox"/>	sd-wan	ALL	ALL	Source IP	ALL				any

4

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.

Edit SD-WAN Template
CLI Configurations ✕

Name:

Description:

SD-WAN Status:

SD-WAN Zones

+ Create New
Edit
Delete
Where Used
Search...

<input type="checkbox"/>	ID	Interface	Gateway	Cost	Priority	Status	
<input type="checkbox"/>	DIA						
<input type="checkbox"/>	1	port1	\$(Br_ISPA_gw)	0	1	Enable	
<input type="checkbox"/>	2	port2	\$(Br_ISPB_gw)	0	1	Enable	
<input type="checkbox"/>	overlay						
<input type="checkbox"/>	3	HUB1-VPN1	0.0.0.0	0	1	Enable	
<input type="checkbox"/>	4	HUB1-VPN2	0.0.0.0	0	1	Enable	

96% 7

SD-WAN Rules

+ Create New
Edit
Delete
More
Search...

<input type="checkbox"/>	ID	Name	Source	Destination	Criteria	Members	Status	Performance SLA	Port	Protocol	
<input type="checkbox"/>		sd-wan	ALL	ALL	Source IP	ALL				any	

1

Performance SLA

+ Create New
Edit
Delete
Where Used
Search...

<input type="checkbox"/>	Name	Detect Server	Packet Loss	Latency	Jitter	Failure Threshold	Recovery	
<input type="checkbox"/>	Default_DNS	(System DNS)	5	250	50	5	10	
<input type="checkbox"/>	Default_FortiGuard	fortiguard.com	5	250	50	5	10	
<input type="checkbox"/>	Default_Gmail	gmail.com	2	250	50	5	10	
<input type="checkbox"/>	Default_Google Search	www.google.com	5	250	50	5	10	
<input type="checkbox"/>	Default_Office_365	www.office.com	5	250	50	5	10	
<input type="checkbox"/>	HUB_HC	10.100.100.100	5	20	20	5	5	

98% 7

Neighbor

+ Create New
Edit
Delete
Search...

<input type="checkbox"/>	Neighbor	Role	Interface Member	Performance SLA	SLA	
<input type="checkbox"/>	10.50.2.1	Standalone	HUB1-VPN1	HUB_HC	1	
<input type="checkbox"/>	10.50.1.1	Standalone	HUB1-VPN2	HUB_HC	1	

2

Branch Configuration Preview

```
1 config vpn ipsec phase1-interface
2   edit "HUB1-VPN1"
3     set interface "port1"
4     set ike-version 2
5     set comments "VPN: HUB1-VPN1 [Created by IPSEC Template]"
6     set proposal aes256-sha256
7     set peertype any
8     set mode-cfg enable
9     set localid "Branch1"
10    set remote-gw 100.1.100.1
11    set idle-timeout enable
12    set net-device enable
13    set add-route disable
14    set auto-discovery-receiver enable
15    set psksecret ENC Z8Zpc/bwU2j1HxCFWz0/XkKwz1iP/WK4qAGVHE9oazICt
16    set network-overlay enable
17    set network-id 1
18    set auto-discovery-shortcuts dependent
19  next
20 end
21 config system interface
22   edit "HUB1-VPN1"
23     set vdom "root"
24     set type tunnel
25     set snmp-index 116
26     set interface "port1"
27   next
28 end
29 config vpn ipsec phase1-interface
30   edit "HUB1-VPN2"
31     set interface "port2"
32     set ike-version 2
33     set comments "VPN: HUB1-VPN2 [Created by IPSEC Template]"
34     set proposal aes256-sha256
35     set peertype any
36     set mode-cfg enable
37     set localid "Branch1"
38     set remote-gw 200.2.100.1
39     set idle-timeout enable
40     set net-device enable
41     set add-route disable
42     set auto-discovery-receiver enable
43     set psksecret ENC Z8Zpc/bwU2j1HxCFWz0/XkKwz1iP/WK4qAGVHE9oazICt
44     set network-overlay enable
45     set network-id 2
46     set auto-discovery-shortcuts dependent
47   next
48 end
49
50 config system interface
51   edit "HUB1-VPN2"
52     set vdom "root"
53     set allowaccess ping
54     set type tunnel
55     set snmp-index 117
56     set interface "port2"
57   next
58   edit "BR-Lo"
59     set vdom "root"
60     set ip 10.100.100.1 255.255.255.255
61     set allowaccess ping
62     set type loopback
63     set snmp-index 118
64   next
65 end
66 config vpn ipsec phase2-interface
67   edit "HUB1-VPN1"
68     set phase1name "HUB1-VPN1"
69     set proposal aes256-sha256
70     set auto-negotiate enable
71     set comments "VPN: HUB1-VPN1 [Created by IPSEC Template]"
72   next
73   edit "HUB1-VPN2"
74     set phase1name "HUB1-VPN2"
75     set proposal aes256-sha256
76     set auto-negotiate enable
77     set comments "VPN: HUB1-VPN2 [Created by IPSEC Template]"
78   next
79 end
80 config system sdwan
81   set status enable
82   config zone
83     edit "DIA"
84     next
85     edit "overlay"
86     next
87   end
88   config members
89     edit 1
90       set interface "port1"
91       set zone "DIA"
92       set gateway 100.1.1.2
93     next
94     edit 2
95       set interface "port2"
96       set zone "DIA"
97       set gateway 200.2.1.2
98   end
99
100 next
101 edit 3
102   set interface "HUB1-VPN1"
103   set zone "overlay"
104 next
105 edit 4
106   set interface "HUB1-VPN2"
107   set zone "overlay"
108 next
109 end
110 config health-check
111   edit "HUB_HC"
112     set server 10.100.100.100
113     set update-static-route disable
114     set members 0
115   config sla
116     edit 1
117       set latency-threshold 20
118       set jitter-threshold 20
119       set packetloss-threshold 5
120     next
121   end
122 next
123 config router route-map
124   edit "OUT-OF-SLA"
125     config rule
126       edit 1
127         set set-community "65001:1"
128       next
129     end
130   next
131   edit "IN-SLA-TUN1"
132     config rule
133       edit 1
134         set set-community "65001:1"
135       next
136     end
137   next
138 end
139 config router bgp
140   config neighbor
141     edit "10.50.1.1"
142       set advertisement-interval 1
143       set capability-graceful-restart enable
144       set link-down-failover enable
145       set soft-reconfiguration enable
146     end
```

```

145     set description "HUB1-VPN1"
146     set interface "HUB1-VPN1"
147     set remote-as 65001
148     set route-map-out "OUT-OF-SLA"
149     set connect-timer 10
150     set additional-path receive
151     set route-map-out-preferable "IN-SLA-TUN1"
152     next
153     end
154 end
155 config system swan
156 config neighbor
157     edit "10.50.1.1"
158     set health-check "HUB_HC"
159     set sla-id 1
160     set member 4
161     next
162     end
163 end
164 config router route-map
165     edit "IN-SLA-TUN2"
166     config rule
167         edit 1
168             set set-community "65001:2"
169         next
170     end
171     next
172 end
173 config router bgp
174     config neighbor
175         edit "10.50.2.1"
176             set advertisement-interval 1
177             set capability-graceful-restart enable
178             set link-down-failover enable
179             set soft-reconfiguration enable
180             set description "HUB1-VPN2"
181             set interface "HUB1-VPN2"
182             set remote-as 65001
183             set route-map-out "OUT-OF-SLA"
184             set connect-timer 10
185             set additional-path receive
186             set route-map-out-preferable "IN-SLA-TUN2"
187         next
188     end
189 end
190 config system swan
191 config neighbor
192     edit "10.50.2.1"

```

```

193     set health-check "HUB_HC"
194     set sla-id 1
195     set member 3
196     next
197     end
198 end
199 config router prefix-list
200     edit "all_prefixes"
201     config rule
202         edit 1
203             set prefix any
204             unset ge
205             unset le
206         next
207     end
208     next
209 end
210 config router route-map
211     edit "port3_only"
212     config rule
213         edit 1
214             set match-interface "port3"
215         next
216         edit 2
217             set action deny
218             set match-ip-address "all_prefixes"
219         next
220     end
221     next
222 end
223 config router bgp
224     set as 65001
225     set router-id 10.100.100.1
226     set ibgp-multipath enable
227     set additional-path enable
228     set graceful-restart enable
229     set additional-path-select 255
230     config network
231         edit 1
232             set prefix 10.10.1.0 255.255.255.0
233         next
234     end
235     set recursive-next-hop enable
236     config redistribute "connected"
237         set route-map "port3_only"
238     end
239 end

```

HUB Configuration Preview

```

1  config vpn ipsec phase1-interface
2      edit "VPN1"
3          set type dynamic
4          set interface "port1"
5          set ike-version 2
6          set dpd on-idle
7          set comments "VPN: VPN1 [Created by IPSEC Template]"
8          set proposal aes256-sha256
9          set peertype any
10         set mode-cfg enable
11         set dpd-retryinterval 60
12         set net-device disable
13         set add-route disable
14         set auto-discovery-sender enable
15         set ipv4-start-ip 10.50.1.10
16         set ipv4-end-ip 10.50.1.250
17         set ipv4-netmask 255.255.255.0
18         set psksecret ENC UN0Ue1vt1Gf18sCvME5Nhp6LVgclszcHC8MEeCa
19         set network-overlay enable
20         set network-id 1
21     next
22 end
23 config system interface
24     edit "VPN1"
25         set vdom "root"
26         set ip 10.50.1.1 255.255.255.255
27         set type tunnel
28         set remote-ip 10.50.1.253 255.255.255.0
29         set snmp-index 116
30         set interface "port1"
31     next
32 end
33 config vpn ipsec phase1-interface
34     edit "VPN2"
35         set type dynamic
36         set interface "port2"
37         set ike-version 2
38         set dpd on-idle
39         set comments "VPN: VPN2 [Created by IPSEC Template]"
40         set proposal aes256-sha256
41         set peertype any
42         set mode-cfg enable
43         set dpd-retryinterval 60
44         set net-device disable
45         set add-route disable
46         set auto-discovery-sender enable
47         set ipv4-start-ip 10.50.2.10
48         set ipv4-end-ip 10.50.2.250

```

```

49         set ipv4-netmask 255.255.255.0
50         set psksecret ENC UN0Ue1vt1Gf18sCvME5Nhp6LVgclszcHC8MEeCa
51         set network-overlay enable
52         set network-id 2
53     next
54 end
55 config system interface
56     edit "VPN2"
57         set vdom "root"
58         set ip 10.50.2.1 255.255.255.255
59         set type tunnel
60         set remote-ip 10.50.2.253 255.255.255.0
61         set snmp-index 117
62         set interface "port2"
63     next
64     edit "HUB1-Lo"
65         set vdom "root"
66         set ip 10.100.100.100 255.255.255.255
67         set allowaccess ping
68         set type loopback
69         set snmp-index 118
70     next
71 end
72 config vpn ipsec phase2-interface
73     edit "VPN1"
74         set phase-name "VPN1"
75         set proposal aes256-sha256
76         set comments "VPN: VPN1 [Created by IPSEC Template]"
77     next
78     edit "VPN2"
79         set phase-name "VPN2"
80         set proposal aes256-sha256
81         set comments "VPN: VPN2 [Created by IPSEC Template]"
82     next
83 end
84 config system sdwan
85     config zone
86         edit "overlay"
87         next
88     end
89     config members
90         edit 3
91             set interface "VPN1"
92             set zone "overlay"
93         next
94         edit 4
95             set interface "VPN2"
96             set zone "overlay"

```

```

97     next
98 end
99 config service
100     edit 1
101         set name "TUN1"
102         set route-tag 1
103         set src "all"
104         set priority-members 3
105     next
106     edit 2
107         set name "TUN2"
108         set route-tag 2
109         set src "all"
110         set priority-members 4
111     next
112     edit 3
113         set name "OUT-OF-SLA"
114         set route-tag 11
115         set priority-members 3 4
116     next
117 end
118 end
119 config router prefix-list
120     edit "all_prefixes"
121     config rule
122         edit 1
123             set prefix any
124             unset ge
125             unset le
126     next
127 end
128 next
129 end
130 config router community-list
131     edit "65001:1"
132     config rule
133         edit 1
134             set action permit
135             set match "65001:1"
136     next
137 end
138 next
139     edit "65001:11"
140     config rule
141         edit 1
142             set action permit
143             set match "65001:11"
144     next

```

```

145     end
146     next
147     edit "65001:2"
148         config rule
149             edit 1
150                 set action permit
151                 set match "65001:2"
152             next
153         end
154     next
155 end
156 config router route-map
157     edit "Route-Map-In"
158         config rule
159             edit 1
160                 set match-community "65001:1"
161                 set set-route-tag "1"
162             next
163             edit 2
164                 set match-community "65001:2"
165                 set set-route-tag "2"
166             next
167             edit 3
168                 set match-community "65001:11"
169                 set set-route-tag "11"
170             next
171         end
172     next
173     edit "port3_only"
174         config rule
175             edit 1
176                 set match-interface "port3"
177             next
178             edit 2
179                 set action deny
180                 set match-ip-address "all_prefixes"
181             next
182         end
183     next
184 end
185 config router bgp
186     set as 65001
187     set router-id 10.100.100.100
188     set ibgp-multipath enable
189     set additional-path enable
190     set graceful-restart enable
191     set additional-path-select 255
192     config neighbor-group

```

```

193     edit "VPN1"
194         set capability-graceful-restart enable
195         set link-down-failover enable
196         set next-hop-self enable
197         set soft-reconfiguration enable
198         set remote-as 65001
199         set route-map-in "Route-Map-In"
200         set additional-path send
201         set route-reflector-client enable
202     next
203     edit "VPN2"
204         set capability-graceful-restart enable
205         set link-down-failover enable
206         set next-hop-self enable
207         set soft-reconfiguration enable
208         set remote-as 65001
209         set route-map-in "Route-Map-In"
210         set additional-path send
211         set route-reflector-client enable
212     next
213 end
214 config neighbor-range
215     edit 1
216         set prefix 10.50.1.0 255.255.255.0
217         set neighbor-group "VPN1"
218     next
219     edit 2
220         set prefix 10.50.2.0 255.255.255.0
221         set neighbor-group "VPN2"
222     next
223 end
224 config network
225     edit 1
226         set prefix 10.10.100.0 255.255.255.0
227     next
228 end
229 config redistribute "connected"
230     set route-map "port3_only"
231 end
232 end

```

IPsec Verification

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

The image displays three screenshots of the Fortinet IPsec Monitor interface, showing the status of established VPN tunnels. Each screenshot includes a table with columns for Name, Remote Gateway, Peer ID, Incoming Data, Outgoing Data, Phase 1, and Phase 2 Selectors.

HUB1 IPsec Monitor

Name	Remote Gateway	Peer ID	Incoming Data	Outgoing Data	Phase 1	Phase 2 Selectors
VPN1_0	100.1.2.1	Branch2	20.48 kB	14.21 kB	VPN1_0	VPN1
VPN1_1	100.1.1.1	Branch1	20.44 kB	14.21 kB	VPN1_1	VPN1
VPN2_0	200.2.2.1	Branch2	20.48 kB	14.21 kB	VPN2_0	VPN2
VPN2_1	200.2.1.1	Branch1	20.44 kB	14.21 kB	VPN2_1	VPN2

DR1 IPsec Monitor

Name	Remote Gateway	Peer ID	Incoming Data	Outgoing Data	Phase 1	Phase 2 Selectors
HUB1-VPN1	100.1.100.1	100.1.100.1	10.48 kB	16.72 kB	HUB1-VPN1	HUB1-VPN1
HUB1-VPN2	200.2.100.1	200.2.100.1	10.48 kB	16.72 kB	HUB1-VPN2	HUB1-VPN2

BR2 IPsec Monitor

Name	Remote Gateway	Peer ID	Incoming Data	Outgoing Data	Phase 1	Phase 2 Selectors
HUB1-VPN1	100.1.100.1	100.1.100.1	11.40 kB	17.68 kB	HUB1-VPN1	HUB1-VPN1
HUB1-VPN2	200.2.100.1	200.2.100.1	11.40 kB	17.68 kB	HUB1-VPN2	HUB1-VPN2

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         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.

The screenshot shows the 'Create New SD-WAN Overlay Template - Region Settings (1/5)' window. The left sidebar lists various provisioning templates, with 'SD-WAN Overlay Templ...' selected. The main area contains the following fields and options:

- Name:** DUAL-HUB
- Description:** (empty text area)
- Select New Topology:** Three options are shown: 'Single HUB', 'Dual HUB (Primary & Secondary)' (selected), and 'Dual HUB (Primary & Primary)'.
- Advanced:** A dropdown menu is expanded to show:
 - Loopback IP Address:** 10.100.100.0/255.255.255.0
 - Overlay Network:** 10.50.0.0/255.255.0.0
 - BGP-AS Number:** 65001
 - Auto-Discovery VPN:** A toggle switch is turned on.

Specify the HUBs (primary/secondary) and branches.

The screenshot shows the 'Create New SD-WAN Overlay Template - Role Assignment (2/5)' window. The configuration is as follows:

- Name:** DUAL-HUB
- Topology:** Three buttons are shown: 'Single HUB', 'Dual HUB (Primary & Secondary)' (selected), and 'Dual HUB (Primary & Primary)'.
- HUB:** A section with two rows:
 - Primary HUB:** A dropdown menu containing 'HUB1'.
 - Secondary HUB:** A dropdown menu containing 'HUB2'.
- Branch:** A section with one row:
 - Device Group Assignment:** A dropdown menu containing 'Branches'.

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

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

Name: DUAL-HUB

HUB

Primary HUB: HUB1

Underlay	#	Private Link	Override IP	Action
WAN Underlay 1		<input checked="" type="checkbox"/> port1	<input checked="" type="checkbox"/>	<input type="button" value="x"/> <input type="button" value="+"/>
WAN Underlay 2		<input checked="" type="checkbox"/> port2	<input checked="" type="checkbox"/>	<input type="button" value="x"/> <input type="button" value="+"/>

Network Advertisement: Connected Static

#	Interface	Action
Interface 1	port3	<input type="button" value="x"/> <input type="button" value="+"/>

Advanced >

Secondary HUB: HUB2

Underlay	#	Private Link	Override IP	Action
WAN Underlay 1		<input checked="" type="checkbox"/> port1	<input checked="" type="checkbox"/>	<input type="button" value="x"/> <input type="button" value="+"/>
WAN Underlay 2		<input checked="" type="checkbox"/> port2	<input checked="" type="checkbox"/>	<input type="button" value="x"/> <input type="button" value="+"/>

Network Advertisement: Connected Static

#	Interface	Action
Interface 1	port3	<input type="button" value="x"/> <input type="button" value="+"/>

Advanced >

Branch Route Maps

Route map in: **Route-Map-In**

Route map out:

Create New SD-WAN Overlay Template - SD-WAN Template Options (4/5)

- Add Overlay Objects to SD-WAN Template
- Add Overlay Interfaces and Zones
- Add Health Check Servers for Each HUB as Performance SLA

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

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

Branch

Branch Device Group: Branches

Underlay

#	Private Link	Action
WAN Underlay 1	port1	[X] [+]
WAN Underlay 2	port2	[X] [+]

Network Advertisement

Connected Static

#	Interface	Action
Interface 1	port3	[X] [+]

Advanced

Apply to All / Specify: Apply to All Specify

HUB 1 Overlay 1

- Route map in: [OFF]
- Route map out: [ON] OUT-OF-SLA
- Route map out preferable: [ON] IN-SLA-TUN1

HUB 1 Overlay 2

- Route map in: [OFF]
- Route map out: [ON] OUT-OF-SLA
- Route map out preferable: [ON] IN-SLA-TUN2

HUB 2 Overlay 1

- Route map in: [OFF]
- Route map out: [ON] OUT-OF-SLA
- Route map out preferable: [ON] IN-SLA-TUN5

HUB 2 Overlay 2

- Route map in: [OFF]
- Route map out: [ON] OUT-OF-SLA
- Route map out preferable: [ON] IN-SLA-TUN6

< Back Next > Cancel

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

Create New SD-WAN Overlay Template - Summary (5/5) ✕

Please review the summary of SD-WAN Overlay configurations


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

Template Name	DUAL-HUB
Topology	Dual HUB (Primary & Secondary)

Region Network Settings ▾

Loopback Allocated	10.100.100.0/255.255.255.0
Overlay Network	10.50.0.0/255.255.0.0
BGP AS Number	65001
Auto-Discovery VPN	<input checked="" type="checkbox"/>

Device Assignment ▾

Primary HUB	HUB1 (192.168.11.201, Platform: FortiGate-VM64-KVM)
Secondary HUB	HUB2 (192.168.11.202, Platform: FortiGate-VM64-KVM)
Branch	 Branches

Underlay Assignment ▾

Primary HUB Underlays	Underlay 1: port1 Underlay 2: port2
Secondary HUB Underlays	Underlay 1: port1 Underlay 2: port2
Branch Underlays	Underlay 1: port1 Underlay 2: port2

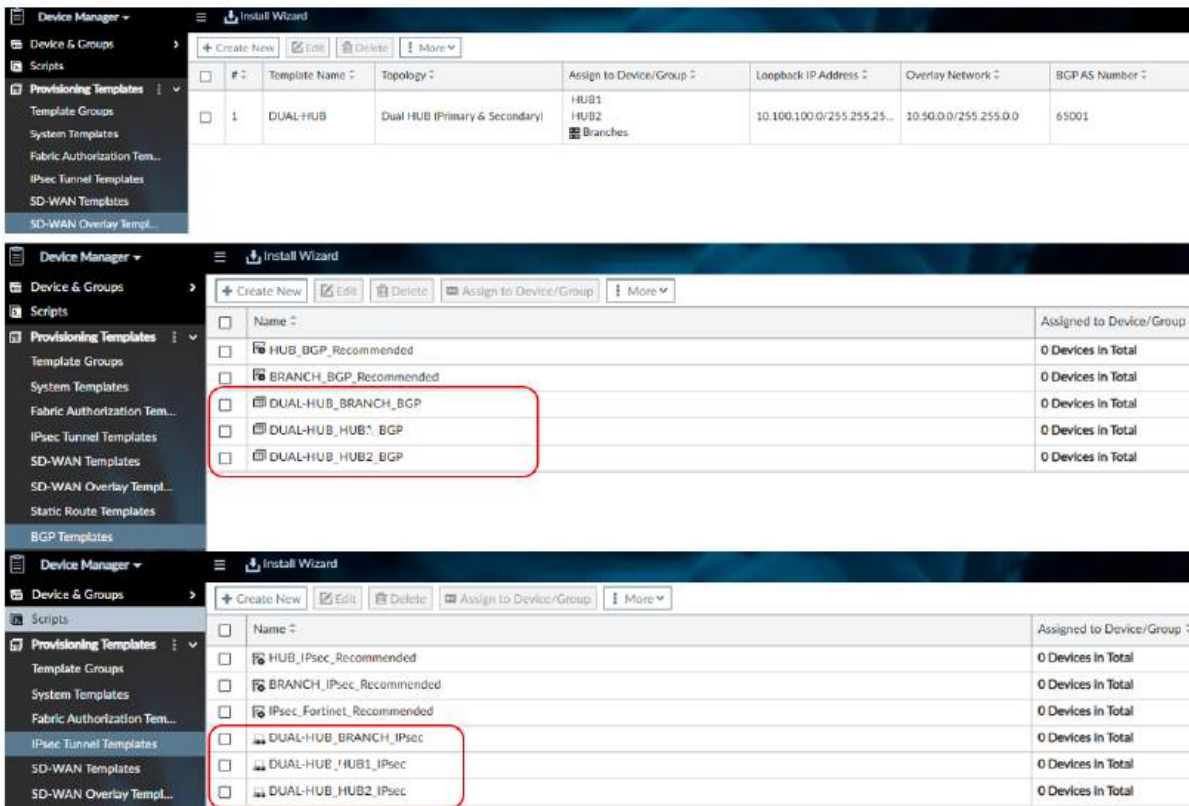
Network Advertisement ▾

Primary HUB	Connected Interface 1: port3
Secondary HUB	Connected Interface 1: port3
Branch	Connected Interface 1: port3

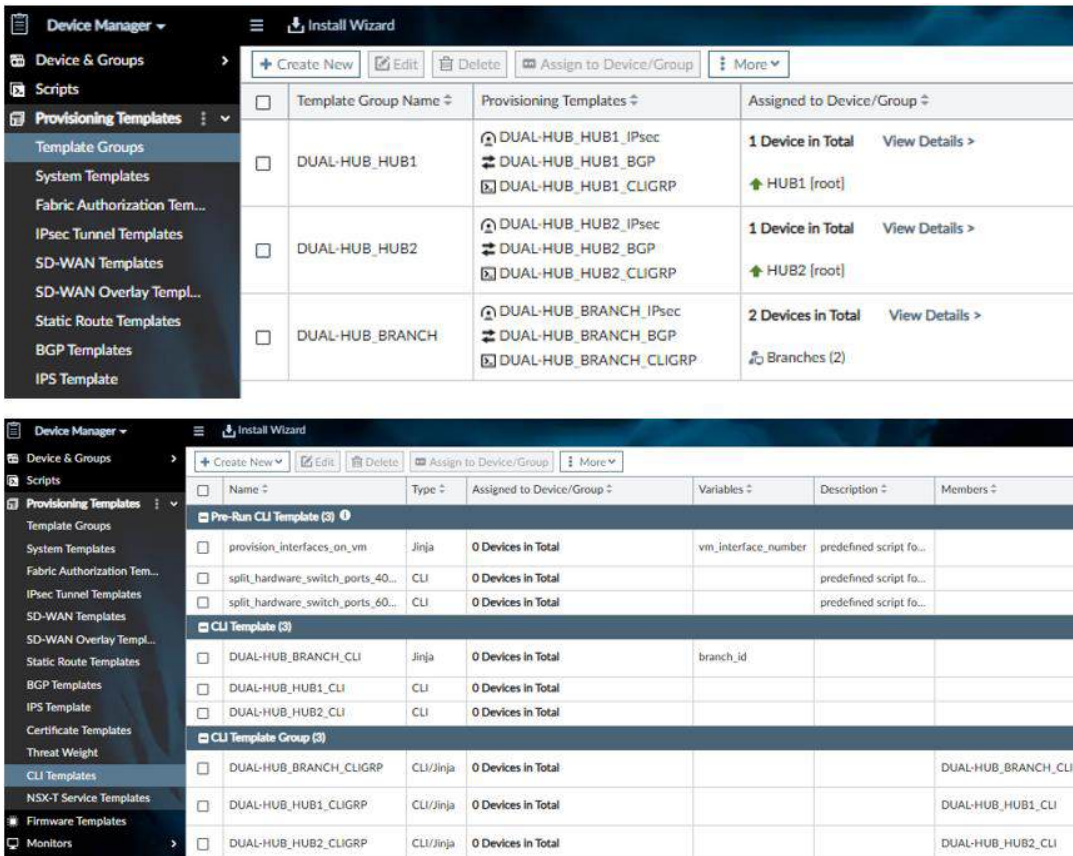
SD-WAN Template Options ▾

Add Overlay Objects to SD-WAN Template	<input checked="" type="checkbox"/>
Add Overlay Interfaces and Zones	<input checked="" type="checkbox"/>
Add Health Check Servers for Each HUB as Performance SLA	<input checked="" type="checkbox"/>

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.

The screenshot shows the 'Policy & Objects' interface. On the left, a sidebar lists 'Object Configurations' with sub-items: 'Normalized Interface', 'Firewall Objects', 'Security Profiles', and 'Fabric Connectors'. The main area displays a table of metadata variables:

<input type="checkbox"/>	Name	Default Value	Description	Created Time
<input type="checkbox"/>	branch_id			admin / 2024-11-05 09:45:13
<input type="checkbox"/>	vm_interface_number	1	predefined variable for set fg	admin / 2024-10-13 19:05:26

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

The image displays three screenshots of IPsec Template configuration windows. The first window, 'IPsec Template - DUAL-HUB_BRANCH_IPsec', shows a table with four entries:

<input type="checkbox"/>	Name	Type	Outgoing Interface	Local Interface
<input type="checkbox"/>	HUB1-VPN1	Static	port1	
<input type="checkbox"/>	HUB1-VPN2	Static	port2	
<input type="checkbox"/>	HUB2-VPN1	Static	port1	
<input type="checkbox"/>	HUB2-VPN2	Static	port2	

The second window, 'IPsec Template - DUAL-HUB_HUB1_IPsec', shows a table with two entries:

<input type="checkbox"/>	Name	Type	Outgoing Interface	Local Interface
<input type="checkbox"/>	VPN1	Dynamic	port1	
<input type="checkbox"/>	VPN2	Dynamic	port2	

The third window, 'IPsec Template - DUAL-HUB_HUB2_IPsec', shows a table with two entries. A red box highlights the table's header and the first two rows:

<input type="checkbox"/>	Name	Type	Outgoing Interface	Local Interface
<input type="checkbox"/>	VPN5	Dynamic	port1	
<input type="checkbox"/>	VPN6	Dynamic	port2	

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.

Edit Metadata Variables - branch_id

Name: branch_id

Description:

Default Value:

Per-Device Mapping

Mapped Device	Value
BR1 [root]	1
BR2 [root]	2
HUB1 [root]	100
HUB2 [root]	200

Edit Metadata Variables - Br_ISPA_gw

Name: Br_ISPA_gw

Description:

Default Value:

Per-Device Mapping

Mapped Device	Value
BR1 [root]	100.1.1.2
BR2 [root]	100.1.2.2

Edit Metadata Variables - Br_ISPB_gw

Name: Br_ISPB_gw

Description:

Default Value:

Per-Device Mapping

Mapped Device	Value
BR1 [root]	200.2.1.2
BR2 [root]	200.2.2.2

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

The image shows two instances of the 'Edit BGP Template' window. Both windows have the following fields: Name (DUAL-HUB_BRANCH_BGP), Description, Local As (65001), and Router ID (10.100.100.\$(branch_id)).

Left Window Neighbors Table:

IP	Remote AS
<input type="checkbox"/> 10.50.159.253	65001
<input type="checkbox"/> 10.50.191.253	65001
<input type="checkbox"/> 10.50.31.253	65001
<input type="checkbox"/> 10.50.63.253	65001

Right Window Neighbors Table:

IP	Remote AS
<input type="checkbox"/> 10.50.1.1	65001
<input type="checkbox"/> 10.50.2.1	65001
<input type="checkbox"/> 10.50.5.1	65001
<input type="checkbox"/> 10.50.6.1	65001

IPv4 Redistribute: Connected: All Filter port3_only. RIP, OSPF, Static, and ISIS are all checked.

Networks: IP/Netmask: 10.10.\$(branch_id).0 255.255.255.0. IPv6 Networks: IP/Netmask: +

The default template neighbor settings are shown below:

The image shows four instances of the 'Edit Neighbor' window, each with a different configuration:

- Instance 1:** IP: 10.50.31.253, Remote AS: 65001, Password: ***** (masked), Interface: HUB1-VPN1, Update Source: , Graceful Restart Time: 0, Advertisement Interval: 1, Route Map Out Preferable: IN-SLA-TUN1, Additional Path: Receive, Connect Timer: 10, Bidirectional Forwarding Detection: , Link Down Fallover: , Shutdown Enable: , IPv4 Filtering: , Filter List In: , Filter List Out: , Distribute List In: , Distribute List Out: , Prefix List In: , Prefix List Out: , Route Map In: , Route Map Out: OUT-OF-SLA.
- Instance 2:** IP: 10.50.63.253, Remote AS: 65001, Password: ***** (masked), Interface: HUB1-VPN2, Update Source: , Graceful Restart Time: 0, Advertisement Interval: 1, Route Map Out Preferable: IN-SLA-TUN2, Additional Path: Receive, Connect Timer: 10, Bidirectional Forwarding Detection: , Link Down Fallover: , Shutdown Enable: , IPv4 Filtering: , Filter List In: , Filter List Out: , Distribute List In: , Distribute List Out: , Prefix List In: , Prefix List Out: , Route Map In: , Route Map Out: OUT-OF-SLA.
- Instance 3:** IP: 10.50.159.253, Remote AS: 65001, Password: ***** (masked), Interface: HUB2-VPN1, Update Source: , Graceful Restart Time: 0, Advertisement Interval: 1, Route Map Out Preferable: IN-SLA-TUN3, Additional Path: Receive, Connect Timer: 10, Bidirectional Forwarding Detection: , Link Down Fallover: , Shutdown Enable: , IPv4 Filtering: , Filter List In: , Filter List Out: , Distribute List In: , Distribute List Out: , Prefix List In: , Prefix List Out: , Route Map In: , Route Map Out: OUT-OF-SLA.
- Instance 4:** IP: 10.50.191.253, Remote AS: 65001, Password: ***** (masked), Interface: HUB2-VPN2, Update Source: , Graceful Restart Time: 0, Advertisement Interval: 1, Route Map Out Preferable: IN-SLA-TUN4, Additional Path: Receive, Connect Timer: 10, Bidirectional Forwarding Detection: , Link Down Fallover: , Shutdown Enable: , IPv4 Filtering: , Filter List In: , Filter List Out: , Distribute List In: , Distribute List Out: , Prefix List In: , Prefix List Out: , Route Map In: , Route Map Out: OUT-OF-SLA.

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

The image displays four screenshots of the 'Edit Neighbor' configuration page, arranged in a 2x2 grid. Each screenshot shows the configuration for a different hub:

- Top Left (HUB1):** IP: 10.50.1.1, Remote AS: 65001, Interface: HUB1-VPN1, Route Map Out Preferable: IN-SLA-TUN1.
- Top Right (HUB2):** IP: 10.50.2.1, Remote AS: 65001, Interface: HUB1-VPN2, Route Map Out Preferable: IN-SLA-TUN2.
- Bottom Left (HUB3):** IP: 10.50.5.1, Remote AS: 65001, Interface: HUB2-VPN1, Route Map Out Preferable: IN-SLA-TUN5.
- Bottom Right (HUB4):** IP: 10.50.6.1, Remote AS: 65001, Interface: HUB2-VPN2, Route Map Out Preferable: IN-SLA-TUN6.

All configurations include a Password field (masked with asterisks), Update Source (radio buttons), Graceful Restart Time (0), Advertisement Interval (1), Connect Timer (10), Bidirectional Forwarding Detection (radio buttons), Link Down Failover (radio buttons), Shutdown Enable (radio buttons), and IPv4 Filtering (radio buttons). The IPv4 Filtering section includes Filter List In/Out, Distribute List In/Out, Prefix List In/Out, and Route Map In/Out, with 'OUT-OF-SLA' selected for Route Map Out in all cases.

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

The image displays two screenshots of the 'Edit BGP Template' configuration page, arranged side-by-side. Both templates are named 'DUAL-HUB_HUB1_BGP' and 'DUAL-HUB_HUB2_BGP' respectively.

- Local As:** 65001
- Router ID:** 10.100.100.253
- Neighbors:** No record found.
- Neighbor Group:**
 - VPN1: Remote AS 65001
 - VPN2: Remote AS 65001
- Neighbor Ranges:**
 - Prefix: 10.50.0.0/255.255.224.0, Neighbor Group: VPN1, Maximum Neighbor Number: 0
 - Prefix: 10.50.32.0/255.255.224.0, Neighbor Group: VPN2, Maximum Neighbor Number: 0
- Networks:**
 - IP/Netmask: 10.50.0.0/255.255.224.0
 - IP/Netmask: 10.50.32.0/255.255.224.0

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

Edit BGP Template

Name: DUAL-HUB_HUB1_BGP

Description:

Local As: 65001

Router ID: 10.100.100.100

Neighbors

+ Create New | Edit | Delete | More

No record found.

Neighbor Group

+ Create New | Edit | Delete | More

Name	Remote AS
VPN1	65001
VPN2	65001

Neighbor Ranges

+ Create New | Edit | Delete | More

Prefix	Neighbor Group	Maximum Neighbor Number
10.50.1.0/255.255.255.0	VPN1	0
10.50.2.0/255.255.255.0	VPN2	0

Networks

IP/Netmask	Action
10.10.\$(branch_id).0/255.255.255.0	[X] [+] []

Edit BGP Template

Name: DUAL-HUB_HUB2_BGP

Description:

Local As: 65001

Router ID: 10.100.100.200

Neighbors

+ Create New | Edit | Delete | More

No record found.

Neighbor Group

+ Create New | Edit | Delete | More

Name	Remote AS
VPN5	65001
VPN6	65001

Neighbor Ranges

+ Create New | Edit | Delete | More

Prefix	Neighbor Group	Maximum Neighbor Number
10.50.5.0/255.255.255.0	VPN5	0
10.50.6.0/255.255.255.0	VPN6	0

Networks

IP/Netmask	Action
10.10.\$(branch_id).0/255.255.255.0	[X] [+] []

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

Edit Route Map

Name:

Comments:

Rules

ID	Action
<input type="checkbox"/> 1	<input checked="" type="checkbox"/> Permit
<input type="checkbox"/> 2	<input checked="" type="checkbox"/> Permit
<input type="checkbox"/> 3	<input checked="" type="checkbox"/> Permit
<input type="checkbox"/> 4	<input checked="" type="checkbox"/> Permit
<input type="checkbox"/> 11	<input checked="" type="checkbox"/> Permit

Edit Route Map Rule

Match community: 65001:1

Match community exact:

Set community delete:

Set Community:

Set extended community target:

Set extended community site-of-origin:

Other Rule Variables

Match metric:

Match origin:

Set origin:

Match route type:

Set metric type:

Set metric:

Set tag value:

Set route tag:

Edit Route Map Rule

Match community: 65001:2

Match community exact:

Set community delete:

Set Community:

Set extended community target:

Set extended community site-of-origin:

Other Rule Variables

Match metric:

Match origin:

Set origin:

Match route type:

Set metric type:

Set metric:

Set tag value:

Set route tag:

Edit Route Map Rule

Match community: 65001:5

Match community exact:

Set community delete:

Set Community:

Set extended community target:

Set extended community site-of-origin:

Other Rule Variables

Match metric:

Match origin:

Set origin:

Match route type:

Set metric type:

Set metric:

Set tag value:

Set route tag:

Edit Route Map Rule

Match community: 65001:6

Match community exact:

Set community delete:

Set Community:

Set extended community target:

Set extended community site-of-origin:

Other Rule Variables

Match metric:

Match origin:

Set origin:

Match route type:

Set metric type:

Set metric:

Set tag value:

Set route tag:

Edit Route Map Rule

Match community: 65001:11

Match community exact:

Set community delete:

Set Community:

Set extended community target:

Set extended community site-of-origin:

Other Rule Variables

Match metric:

Match origin:

Set origin:

Match route type:

Set metric type:

Set metric:

Set tag value:

Finally, the CLI templates are customized accordingly.

The image displays a 3x3 grid of screenshots showing the customization of CLI templates in Fortinet SD-WAN. Each cell shows an 'Edit CLI Template' window with fields for Template Name, Type, Description, and Script Details. Blue dashed arrows point from the original templates to their customized versions.

Top Row (Original Templates):

- Left:** Template Name: DUAL-HUB_HUB1_CLI, Type: CLI Script. Script Details:

```
1 config system interface
2 edit "HUB1-Lo"
3 set vdom "root"
4 set ip 10.100.100.253 255.255.255.255
5 set allowaccess ping
6 set type loopback
7 next
8 end
```
- Middle:** Template Name: DUAL-HUB_HUB2_CLI, Type: CLI Script. Script Details:

```
1 config system interface
2 edit "HUB2-Lo"
3 set vdom "root"
4 set ip 10.100.100.253 255.255.255.255
5 set allowaccess ping
6 set type loopback
7 next
8 end
```
- Right:** Template Name: DUAL-HUB_BRANCH_CLI, Type: Jinja Script. Script Details:

```
1 config system interface
2 edit "branch([branch_id|int])-Lo"
3 set vdom "root"
4 set ip [{"10.100.100.0"|ipmath(branch_id|int)}] 255.255.255.255
5 set allowaccess ping
6 set type loopback
7 next
8 end
9 config router bgp
10 set router-id [{"10.100.100.0"|ipmath(branch_id|int)}]
11 end
```

Middle Row (Customized Templates):

- Left:** Template Name: DUAL-HUB_HUB1_CLI, Type: CLI Script. Script Details:

```
1 config system interface
2 edit "HUB1-Lo"
3 set vdom "root"
4 set ip 10.100.100.100 255.255.255.255
5 set allowaccess ping
6 set type loopback
7 next
8 end
```
- Middle:** Template Name: DUAL-HUB_HUB2_CLI, Type: CLI Script. Script Details:

```
1 config system interface
2 edit "HUB2-Lo"
3 set vdom "root"
4 set ip 10.100.100.200 255.255.255.255
5 set allowaccess ping
6 set type loopback
7 next
8 end
```
- Right:** Template Name: DUAL-HUB_BRANCH_CLI_2, Type: CLI Script. Script Details:

```
1 config system interface
2 edit "BR-Lo"
3 set vdom "root"
4 set ip 10.100.100.5(branch_id) 255.255.255.255
5 set allowaccess ping
6 set type loopback
7 next
8 end
```

Bottom Row (Empty Templates):

- Left:** Template Name: DUAL-HUB_HUB1_CLI, Type: CLI Script. Script Details: (Empty)
- Middle:** Template Name: DUAL-HUB_HUB2_CLI, Type: CLI Script. Script Details: (Empty)
- Right:** Template Name: DUAL-HUB_BRANCH_CLI_2, Type: CLI Script. Script Details: (Empty)

Branch 2 Configuration Preview

```
1 config vpn ipsec phase1-interface
2   edit "HUB1-VPN1"
3     set interface "port1"
4     set ike-version 2
5     set comments "VPN: HUB1-VPN1 [Created by IPSEC Template]"
6     set proposal aes256-sha256
7     set peertype any
8     set mode-cfg enable
9     set localid "Branch2"
10    set remote-gw 100.1.100.1
11    set idle-timeout enable
12    set net-device enable
13    set add-route disable
14    set auto-discovery-receiver enable
15    set psksecret ENC RjZ5BztK7ZzFo5D3gv36uNuoyqfRa0x6RZv1Qh5XQa
16    set network-overlay enable
17    set network-id 1
18    set auto-discovery-shortcuts dependent
19  next
20 end
21 config system interface
22   edit "HUB1-VPN1"
23     set vdom "root"
24     set type tunnel
25     set snmp-index 116
26     set interface "port1"
27  next
28 end
29 config vpn ipsec phase1-interface
30   edit "HUB1-VPN2"
31     set interface "port2"
32     set ike-version 2
33     set comments "VPN: HUB1-VPN2 [Created by IPSEC Template]"
34     set proposal aes256-sha256
35     set peertype any
36     set mode-cfg enable
37     set localid "Branch2"
38     set remote-gw 200.2.100.1
39     set idle-timeout enable
40     set net-device enable
41     set add-route disable
42     set auto-discovery-receiver enable
43     set psksecret ENC RjZ5BztK7ZzFo5D3gv36uNuoyqfRa0x6RZv1Qh5XQa
44     set network-overlay enable
45     set network-id 2
46     set auto-discovery-shortcuts dependent
47  next
48 end
```

```
49 config system interface
50   edit "HUB1-VPN2"
51     set vdom "root"
52     set type tunnel
53     set snmp-index 117
54     set interface "port2"
55  next
56 end
57 config vpn ipsec phase1-interface
58   edit "HUB2-VPN1"
59     set interface "port1"
60     set ike-version 2
61     set comments "VPN: HUB2-VPN1 [Created by IPSEC Template]"
62     set proposal aes256-sha256
63     set peertype any
64     set mode-cfg enable
65     set localid "Branch2"
66     set remote-gw 100.1.200.1
67     set idle-timeout enable
68     set net-device enable
69     set add-route disable
70     set auto-discovery-receiver enable
71     set psksecret ENC RjZ5BztK7ZzFo5D3gv36uNuoyqfRa0x6RZv1Qh5XQa
72     set network-overlay enable
73     set network-id 5
74     set auto-discovery-shortcuts dependent
75  next
76 end
77 config system interface
78   edit "HUB2-VPN1"
79     set vdom "root"
80     set type tunnel
81     set snmp-index 118
82     set interface "port1"
83  next
84 end
85 config vpn ipsec phase1-interface
86   edit "HUB2-VPN2"
87     set interface "port2"
88     set ike-version 2
89     set comments "VPN: HUB2-VPN2 [Created by IPSEC Template]"
90     set proposal aes256-sha256
91     set peertype any
92     set mode-cfg enable
93     set localid "Branch2"
94     set remote-gw 200.2.200.1
95     set idle-timeout enable
96     set net-device enable
```

```
97   set add-route disable
98   set auto-discovery-receiver enable
99   set psksecret ENC RjZ5BztK7ZzFo5D3gv36uNuoyqfRa0x6RZv1Qh5XQa
100  set network-overlay enable
101  set network-id 6
102  set auto-discovery-shortcuts dependent
103 next
104 end
105 config system interface
106   edit "HUB2-VPN2"
107     set vdom "root"
108     set type tunnel
109     set snmp-index 119
110     set interface "port2"
111  next
112   edit "BR-Lo"
113     set vdom "root"
114     set ip 10.100.100.2 255.255.255.255
115     set allowaccess ping
116     set type loopback
117     set snmp-index 120
118  next
119 end
120 config vpn ipsec phase2-interface
121   edit "HUB1-VPN1"
122     set phaseName "HUB1-VPN1"
123     set proposal aes256-sha256
124     set auto-negotiate enable
125     set comments "VPN: HUB1-VPN1 [Created by IPSEC Template]"
126  next
127   edit "HUB1-VPN2"
128     set phaseName "HUB1-VPN2"
129     set proposal aes256-sha256
130     set auto-negotiate enable
131     set comments "VPN: HUB1-VPN2 [Created by IPSEC Template]"
132  next
133   edit "HUB2-VPN1"
134     set phaseName "HUB2-VPN1"
135     set proposal aes256-sha256
136     set auto-negotiate enable
137     set comments "VPN: HUB2-VPN1 [Created by IPSEC Template]"
138  next
139   edit "HUB2-VPN2"
140     set phaseName "HUB2-VPN2"
141     set proposal aes256-sha256
142     set auto-negotiate enable
143     set comments "VPN: HUB2-VPN2 [Created by IPSEC Template]"
144  next
```

```

145 end
146 config system sdwan
147 set status enable
148 config zone
149 edit "overlay"
150 next
151 edit "overlay2"
152 next
153 edit "DIA"
154 next
155 end
156 config members
157 edit 1
158 set interface "port1"
159 set zone "DIA"
160 set gateway 100.1.2.2
161 next
162 edit 2
163 set interface "port2"
164 set zone "DIA"
165 set gateway 200.2.2.2
166 next
167 edit 3
168 set interface "HUB1-VPN1"
169 set zone "overlay"
170 next
171 edit 4
172 set interface "HUB1-VPN2"
173 set zone "overlay"
174 next
175 edit 5
176 set interface "HUB2-VPN1"
177 set zone "overlay2"
178 next
179 edit 6
180 set interface "HUB2-VPN2"
181 set zone "overlay2"
182 next
183 end
184 config health-check
185 edit "HUB1_HC"
186 set server 10.100.100.100
187 set update-static-route disable
188 set members 4 3
189 config sla
190 edit 1
191 set latency-threshold 20
192 set jitter-threshold 20

```

```

193 set packetloss-threshold 5
194 next
195 end
196 next
197 edit "HUB2_HC"
198 set server 10.100.100.200
199 set update-static-route disable
200 set members 6 5
201 config sla
202 edit 1
203 set latency-threshold 20
204 set jitter-threshold 20
205 set packetloss-threshold 5
206 next
207 end
208 next
209 end
210 end
211 config router route-map
212 edit "OUT-OF-SLA"
213 config rule
214 edit 1
215 set set-community "65001:11"
216 next
217 end
218 next
219 edit "IN-SLA-TUN1"
220 config rule
221 edit 1
222 set set-community "65001:1"
223 next
224 end
225 next
226 end
227 config router bgp
228 config neighbor
229 edit "10.50.1.1"
230 set advertisement-interval 1
231 set capability-graceful-restart enable
232 set link-down-failover enable
233 set soft-reconfiguration enable
234 set description "HUB1-VPN1"
235 set interface "HUB1-VPN1"
236 set remote-as 65001
237 set route-map-out "OUT-OF-SLA"
238 set connect-timer 10
239 set additional-path receive
240 set route-map-out-preferable "IN-SLA-TUN1"

```

```

241 next
242 end
243 end
244 config system sdwan
245 config neighbor
246 edit "10.50.1.1"
247 set health-check "HUB1_HC"
248 set sla-id 1
249 set member 3
250 next
251 end
252 end
253 config router route-map
254 edit "IN-SLA-TUN2"
255 config rule
256 edit 1
257 set set-community "65001:2"
258 next
259 end
260 next
261 end
262 config router bgp
263 config neighbor
264 edit "10.50.2.1"
265 set advertisement-interval 1
266 set capability-graceful-restart enable
267 set link-down-failover enable
268 set soft-reconfiguration enable
269 set description "HUB1-VPN2"
270 set interface "HUB1-VPN2"
271 set remote-as 65001
272 set route-map-out "OUT-OF-SLA"
273 set connect-timer 10
274 set additional-path receive
275 set route-map-out-preferable "IN-SLA-TUN2"
276 next
277 end
278 end
279 config system sdwan
280 config neighbor
281 edit "10.50.2.1"
282 set health-check "HUB1_HC"
283 set sla-id 1
284 set member 4
285 next
286 end
287 end
288 config router route-map

```

```

289 edit "IN-SLA-TUN5"
290 config rule
291 edit 1
292 set set-community "65001:5"
293 next
294 end
295 next
296 end
297 config router bgp
298 config neighbor
299 edit "10.50.5.1"
300 set advertisement-interval 1
301 set capability-graceful-restart enable
302 set link-down-fallover enable
303 set soft-reconfiguration enable
304 set description "HUB2-VPN1"
305 set interface "HUB2-VPN1"
306 set remote-as 65001
307 set route-map-out "OUT-OF-SLA"
308 set connect-timer 10
309 set additional-path receive
310 set route-map-out-preferable "IN-SLA-TUN5"
311 next
312 end
313 end
314 config system sdwan
315 config neighbor
316 edit "10.50.5.1"
317 set health-check "HUB2_HC"
318 set sla-id 1
319 set member 5
320 next
321 end
322 end
323 config router route-map
324 edit "IN-SLA-TUN6"
325 config rule
326 edit 1
327 set set-community "65001:6"
328 next
329 end
330 next
331 end
332 config router bgp
333 config neighbor
334 edit "10.50.6.1"
335 set advertisement-interval 1
336 set capability-graceful-restart enable

```

```

337 set link-down-fallover enable
338 set soft-reconfiguration enable
339 set description "HUB2-VPN2"
340 set interface "HUB2-VPN2"
341 set remote-as 65001
342 set route-map-out "OUT-OF-SLA"
343 set connect-timer 10
344 set additional-path receive
345 set route-map-out-preferable "IN-SLA-TUN6"
346 next
347 end
348 end
349 config system sdwan
350 config neighbor
351 edit "10.50.6.1"
352 set health-check "HUB2_HC"
353 set sla-id 1
354 set member 6
355 next
356 end
357 end
358 config router prefix-list
359 edit "all_prefixes"
360 config rule
361 edit 1
362 set prefix any
363 unset ge
364 unset le
365 next
366 end
367 next
368 end
369 config router route-map
370 edit "port3_only"
371 config rule
372 edit 1
373 set match-interface "port3"
374 next
375 edit 2
376 set action deny
377 set match-ip-address "all_prefixes"
378 next
379 end
380 next
381 end
382 config router bgp
383 set as 65001
384 set router-id 10.100.100.2

```

```

385 set ibgp-multipath enable
386 set additional-path enable
387 set graceful-restart enable
388 set additional-path-select 255
389 config network
390 edit 1
391 set prefix 10.10.2.0 255.255.255.0
392 next
393 end
394 set recursive-next-hop enable
395 config redistribute "connected"
396 set route-map "port3_only"
397 end
398 end
399

```

HUB2 Configuration Preview

```
1 config vpn ipsec phase1-interface
2   edit "VPNS"
3     set type dynamic
4     set interface "port1"
5     set ike-version 2
6     set dpd on-idle
7     set comments "VPN: VPNS [Created by IPSEC Template]"
8     set proposal aes256-sha256
9     set peertype any
10    set mode-cfg enable
11    set dpd-retryinterval 60
12    set net-device disable
13    set add-route disable
14    set auto-discovery-sender enable
15    set ipv4-start-ip 10.50.5.10
16    set ipv4-end-ip 10.50.5.250
17    set ipv4-netmask 255.255.255.0
18    set psksecret ENC Z8Zpc/bwU2j1HxCFWz0/XkH211P/WK4qAGVHE9oazI
19    set network-overlay enable
20    set network-id 5
21  next
22 end
23 config system interface
24   edit "VPNS"
25     set vdom "root"
26     set ip 10.50.5.1 255.255.255.255
27     set type tunnel
28     set remote-ip 10.50.5.253 255.255.255.0
29     set snmp-index 116
30     set interface "port1"
31  next
32 end
33 config vpn ipsec phase1-interface
34   edit "VPN6"
35     set type dynamic
36     set interface "port2"
37     set ike-version 2
38     set dpd on-idle
39     set comments "VPN: VPN6 [Created by IPSEC Template]"
40     set proposal aes256-sha256
41     set peertype any
42     set mode-cfg enable
43     set dpd-retryinterval 60
44     set net-device disable
45     set add-route disable
46     set auto-discovery-sender enable
47     set ipv4-start-ip 10.50.6.10
48     set ipv4-end-ip 10.50.6.250
49     set ipv4-netmask 255.255.255.0
50     set psksecret ENC Z8Zpc/bwU2j1HxCFWz0/XkH211P/WK4qAGVHE9oazI
51     set network-overlay enable
52     set network-id 6
53  next
54 end
55 config system interface
56   edit "VPN6"
57     set vdom "root"
58     set ip 10.50.6.1 255.255.255.255
59     set type tunnel
60     set remote-ip 10.50.6.253 255.255.255.0
61     set snmp-index 117
62     set interface "port2"
63  next
64   edit "HUB2-Lo"
65     set vdom "root"
66     set ip 10.100.100.200 255.255.255.255
67     set allowaccess ping
68     set type loopback
69     set snmp-index 118
70  next
71 end
72 config firewall address
73   edit "all"
74     set uuid f4beb08c-897c-51ef-3958-70fd1739153f
75  next
76 end
77 config vpn ipsec phase2-interface
78   edit "VPNS"
79     set phase1name "VPNS"
80     set proposal aes256-sha256
81     set comments "VPN: VPNS [Created by IPSEC Template]"
82  next
83   edit "VPN6"
84     set phase1name "VPN6"
85     set proposal aes256-sha256
86     set comments "VPN: VPN6 [Created by IPSEC Template]"
87  next
88 end
89 config system sdwan
90   set status enable
91 config zone
92   edit "DIA"
93  next
94   edit "overlay"
95  next
96 end
97 config members
98   edit 1
99     set interface "port1"
100    set zone "DIA"
101    set gateway 100.1.200.2
102  next
103   edit 2
104     set interface "port2"
105     set zone "DIA"
106     set gateway 200.2.200.2
107  next
108   edit 3
109     set interface "VPNS"
110     set zone "overlay"
111  next
112   edit 4
113     set interface "VPN6"
114     set zone "overlay"
115  next
116 end
117 config service
118   edit 1
119     set name "TUN5"
120     set route-tag 5
121     set src "all"
122     set priority-members 3
123  next
124   edit 2
125     set name "TUN6"
126     set route-tag 6
127     set src "all"
128     set priority-members 4
129  next
130   edit 3
131     set name "OUT-OF-SLA"
132     set route-tag 11
133     set priority-members 3 4
134  next
135 end
136 end
137 config router prefix-list
138   edit "all_prefixes"
139   config rule
140     edit 1
141     set prefix any
142     unset ge
143     unset le
144  next
```

```

145     end
146     next
147 end
148 config router community-list
149     edit "65001:1"
150     config rule
151         edit 1
152             set action permit
153             set match "65001:1"
154         next
155     end
156     next
157     edit "65001:11"
158     config rule
159         edit 1
160             set action permit
161             set match "65001:11"
162         next
163     end
164     next
165     edit "65001:2"
166     config rule
167         edit 1
168             set action permit
169             set match "65001:2"
170         next
171     end
172     next
173     edit "65001:5"
174     config rule
175         edit 1
176             set action permit
177             set match "65001:5"
178         next
179     end
180     next
181     edit "65001:6"
182     config rule
183         edit 1
184             set action permit
185             set match "65001:6"
186         next
187     end
188     next
189 end
190 config router route-map
191     edit "Route-Map-In"
192     config rule
193         edit 1
194             set match-community "65001:1"
195             set set-route-tag "1"
196         next
197         edit 2
198             set match-community "65001:2"
199             set set-route-tag "2"
200         next
201         edit 5
202             set match-community "65001:5"
203             set set-route-tag "5"
204         next
205         edit 6
206             set match-community "65001:6"
207             set set-route-tag "6"
208         next
209         edit 11
210             set match-community "65001:11"
211             set set-route-tag "11"
212         next
213     end
214     next
215     edit "port3_only"
216     config rule
217         edit 1
218             set match-interface "port3"
219         next
220         edit 2
221             set action deny
222             set match-ip-address "all_prefixes"
223         next
224     end
225     next
226 end
227 config router bgp
228     set as 65001
229     set router-id 10.100.100.200
230     set igmp-multipath enable
231     set additional-path enable
232     set graceful-restart enable
233     set additional-path-select 255
234     config neighbor-group
235         edit "VPNS"
236             set capability-graceful-restart enable
237             set link-down-fallover enable
238             set next-hop-self enable
239             set soft-reconfiguration enable
240             set remote-as 65001
241         set route-map-in "Route-Map-In"
242         set additional-path send
243         set route-reflector-client enable
244     next
245     edit "VPNS"
246         set capability-graceful-restart enable
247         set link-down-fallover enable
248         set next-hop-self enable
249         set soft-reconfiguration enable
250         set remote-as 65001
251         set route-map-in "Route-Map-In"
252         set additional-path send
253         set route-reflector-client enable
254     next
255 end
256 config neighbor-range
257     edit 1
258         set prefix 10.50.5.0 255.255.255.0
259         set neighbor-group "VPNS"
260     next
261     edit 2
262         set prefix 10.50.6.0 255.255.255.0
263         set neighbor-group "VPNS"
264     next
265 end
266 config network
267     edit 1
268         set prefix 10.10.200.0 255.255.255.0
269     next
270 end
271 config redistribute "connected"
272     set route-map "port3_only"
273 end
274 end
275

```

Branches IPsec Verification

The image shows two screenshots of the Fortinet IPsec Monitor interface. The top screenshot is for BR1 and the bottom is for BR2. Both show a table of active IPsec tunnels with columns for Name, Remote Gateway, Peer ID, Incoming Data, Outgoing Data, Phase 1, and Phase 2 Selectors.

Name	Remote Gateway	Peer ID	Incoming Data	Outgoing Data	Phase 1	Phase 2 Selectors
HUB2-VPN2	200.2.200.1	200.2.200.1	2.97 kB	77.61 kB	HUB2-VPN2	HUB2-VPN2
HUB2-VPN1	100.1.200.1	100.1.200.1	3.04 kB	77.67 kB	HUB2-VPN1	HUB2-VPN1
HUB1-VPN1	100.1.100.1	100.1.100.1	3.14 kB	77.74 kB	HUB1-VPN1	HUB1-VPN1
HUB1-VPN2	200.2.100.1	200.2.100.1	3.19 kB	77.79 kB	HUB1-VPN2	HUB1-VPN2

Name	Remote Gateway	Peer ID	Incoming Data	Outgoing Data	Phase 1	Phase 2 Selectors
HUB1-VPN1	100.1.100.1	100.1.100.1	3.12 kB	78.86 kB	HUB1-VPN1	HUB1-VPN1
HUB1-VPN2	200.2.100.1	200.2.100.1	3.12 kB	78.86 kB	HUB1-VPN2	HUB1-VPN2
HUB2-VPN1	100.1.200.1	100.1.200.1	3.07 kB	78.79 kB	HUB2-VPN1	HUB2-VPN1
HUB2-VPN2	200.2.200.1	200.2.200.1	3.14 kB	78.84 kB	HUB2-VPN2	HUB2-VPN2

HUBs IPsec Verification

HUB1

IPsec

Reset Statistics Bring Up Bring Down Locate on VPN Map Show Matching Logs

Name	Remote Gateway	Peer ID	Incoming Data	Outgoing Data	Phase 1	Phase 2 Selectors
VPN1_0	100.1.2.1	Branch2	84.18 kB	3.25 kB	VPN1_0	VPN1
VPN1_1	100.1.1.1	Branch1	84.13 kB	3.32 kB	VPN1_1	VPN1
VPN2_0	200.2.2.1	Branch2	84.23 kB	3.32 kB	VPN2_0	VPN2
VPN2_1	200.2.1.1	Branch1	84.11 kB	3.32 kB	VPN2_1	VPN2

HUB2

IPsec

Reset Statistics Bring Up Bring Down Locate on VPN Map Show Matching Logs

Name	Remote Gateway	Peer ID	Incoming Data	Outgoing Data	Phase 1	Phase 2 Selectors
VPN5_0	100.1.2.1	Branch2	86.23 kB	3.32 kB	VPN5_0	VPN5
VPN5_1	100.1.1.1	Branch1	86.11 kB	3.28 kB	VPN5_1	VPN5
VPN6_0	200.2.2.1	Branch2	86.28 kB	3.39 kB	VPN6_0	VPN6
VPN6_1	200.2.1.1	Branch1	86.06 kB	3.21 kB	VPN6_1	VPN6

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.

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
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