



Architecture for MSSPs

SD-WAN/SD-Branch 7.6



DEFINE / DESIGN / DEPLOY / DEMO



Table of Contents

Change Log	5
Introduction	6
Intended audience	6
Secure SD-WAN/SD-Branch Solution	7
Basic SD-WAN configuration	7
Practical basic SD-WAN examples	9
Overlay network	10
Practical overlay examples	11
Provisioning and management	13
Deployment workflow	15
Zero-touch and low-touch provisioning	16
Analytics and reporting	19
Secure SD-Branch	20
SD-Branch components	21
SD-Branch provisioning	21
Monitoring and incidence response	22
FortiMonitor	22
FortiAIOps	24
Summary	25
MSSP deployment blueprints	26
Spokes as Edge CPEs	26
Hubs	27
Blueprint 1: MSSP premises, multitenant	27
Blueprint 2: MSSP premises, no multitenancy	28
Blueprint 3: customer premises	28
Management	29
Blueprint 1: MSSP premises / public cloud, multitenant with direct customer access	29
Blueprint 2: MSSP premises / public cloud, multitenant with FortiPortal	30
Blueprint 3: MSSP premises / public cloud, no multitenancy	30

Monitoring and incidence response	31
External orchestration	31
Overlay network designs	33
Basic SD-WAN/ADVPN design	34
IPsec overlay	35
BGP routing	36
ADVPN 2.0	37
Transport groups	38
Health measurement	38
Shortcut management	39
SD-WAN node reachability	41
Routing design	42
BGP on loopback	42
IKE extension: exchange-ip-addrv4	43
ADVPN support with dynamic BGP (RR-less)	44
ADVPN support with BGP RR	45
Multi-regional deployment	48
Choosing ADVPN support method	49
Mixed deployment	51
Planning guidelines	53
Segmentation over single overlay	53
Routing design	55
CE and PE VRFs	56
Planning guidelines	56
Providing Internet access	57
Additional topics	59
Topologies with multiple Internet links	59
Hub connection: many-to-one	61
Hub connection: full-mesh	62
Hub-to-Edge SD-WAN	62
Hub-to-Spoke sessions	63
Advertising preferred Hub	64
Summary	67
SD-WAN strategy and dynamic routing	68
Practical example	68
Deep-dive: route-tags in SD-WAN rules	69
Reference configurations	72
Dynamic BGP on loopback (dual-hub region)	72
Spoke	72
Hub	76
Optimizing failover	78
ADVPN 2.0	78
Spoke	79
Hub	80
Segmentation over single overlay	80
Spoke	80
Hub	83
Providing Internet access	85

Hub-to-edge SD-WAN	86
Spoke	86
Hub	87
Product models and firmware versions	89
External resources	90

Change Log

Date	Change Description
2025-08-11	Initial release.

Introduction

This document describes the reference architecture of Fortinet Secure SD-WAN/SD-Branch solution. It is mainly written for Managed Service Providers, although it may benefit any type of customer looking for a better understanding of our solution, its components, planning guidelines, and best-practice designs.

- In [Secure SD-WAN/SD-Branch Solution on page 7](#), we describe the Secure SD-WAN functionality available on any FortiGate device. We also explain how the functionality can grow into a full SD-WAN (or, optionally, SD-Branch) solution, with fully functional FortiGate devices that control the local wired and wireless connectivity deployed on every site and centrally managed by FortiManager and FortiAnalyzer.
- Next, in [MSSP deployment blueprints on page 26](#), we cover the main deployment models recommended for those willing to offer our solution as a Managed Service, that is, for the MSSPs.
- The next chapter, [Overlay network designs on page 33](#), provides a deeper technical description of the routing designs used in our solution and the overlay network topologies used to interconnect the SD-WAN sites. We also discuss in depth the topic of multi-VRF segmentation across the SD-WAN network.
- Finally, in [Additional topics on page 59](#) we cover other important use cases widely seen in practice.

In all these chapters our aim is to help you design a highly scalable, redundant, and secure SD-WAN/SD-Branch solution, either for your organization or for your customers.

This document is complemented by the [SD-WAN Deployment for MSSPs Guide](#), which describes our recommended approach to configuring the designs described in this document.

Intended audience

This document has been primarily created for a technical audience, including system architects and design engineers. We assume that the reader is familiar with the main concepts of network design, dynamic routing, and security. Working knowledge of Fortinet products is not required (albeit beneficial).

This document does not discuss the benefits of SD-WAN/SD-Branch solutions compared to the traditional WAN/LAN networks; neither does it list competitive advantages offered by Fortinet. We focus on the technical aspects of the Fortinet Secure SD-WAN/SD-Branch Solution design.

Secure SD-WAN/SD-Branch Solution

It is important to distinguish between *Secure SD-WAN functionality* and *Secure SD-WAN Solution*. The Secure SD-WAN functionality can be configured on any FortiGate device, without requiring a separate license or additional products and components. In other words, any FortiGate device can provide this functionality in a completely autonomous manner, including intelligent traffic steering towards multiple WAN links, health monitoring, and of course security.

This chapter will explain how to transform a group of autonomous devices, each providing local Secure SD-WAN functionality, into a comprehensive Secure SD-WAN Solution. FortiGate devices will act as intelligent SD-WAN nodes that are interconnected by an overlay network to provide secure connectivity across all sites, cloud services, and public Internet and to always select an optimal path for each application. Next, we will extend the intelligence into the LAN side, ending up with a complete Secure SD-Branch Solution.

However, before we discuss the design of the Secure SD-WAN/SD-Branch Solution, we must spend some time describing the SD-WAN functionality itself.

- [Basic SD-WAN configuration on page 7](#)
- [Overlay network on page 10](#)
- [Provisioning and management on page 13](#)
- [Analytics and reporting on page 19](#)
- [Secure SD-Branch on page 20](#)
- [Monitoring and incidence response on page 22](#)
- [Summary on page 25](#)

Basic SD-WAN configuration

Still focusing on a single FortiGate device, the SD-WAN configuration includes the following main steps:

1. **SD-WAN Interface Members** define your *SD-WAN bundle*. They are the interfaces controlled by SD-WAN and where the outgoing traffic can be potentially steered. Almost any interface supported by a FortiGate device can become an SD-WAN interface Member, including physical ports, VLAN interfaces, LAGs, IPsec/GRE/IPIP tunnels, and even FortiExtender interfaces.

For convenience, the SD-WAN Members are grouped into *SD-WAN Zones*.

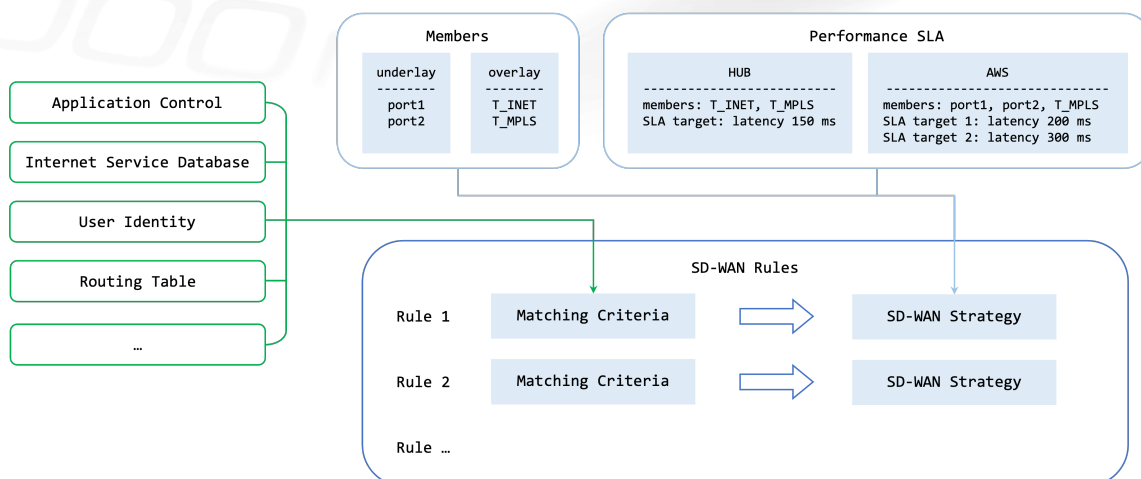


In a complete SD-WAN Solution, all your WAN-facing underlays and overlays are usually configured as SD-WAN Members (although this is not mandatory!). We will return to this topic in the next chapters.

2. **Performance SLA** are the health-check probes used by the FortiGate devices to actively measure the health of each available path. You can define what server to probe and what protocol to use (including Ping, HTTP, TCP/UDP Echo, TWAMP, or DNS). Each probe will measure latency, jitter, and packet loss percentage over a configured subset of the SD-WAN Members.

In addition, you can configure multiple *SLA Targets* for each probe. These metrics together allow SD-WAN to compare the health of different available paths and determine which paths are acceptable for a particular application and which are not. Unacceptable paths are sometimes called *out of SLA*.

3. **SD-WAN Rules** finally combine all the elements together. They are the set of business rules which can steer a particular application to a particular SD-WAN Member, considering its current health and SLA status.



Each SD-WAN rule has the following logical parts:

- *Matching Criteria* defines what applications or what kind of traffic will match this rule. We can match based on a large variety of inputs, including signature-based L7 Application detection (Application Control Database), dynamic feeds (Internet Service Database or ISDB), multiple User Identity providers, information received from dynamic routing (using Route-Tags), DSCP/ToS fields... and of course, also based on simple L3/L4 criteria!
- *SD-WAN Strategy* defines the logic used to select one of the SD-WAN Members to steer this traffic. The following strategies can be configured:
 - *Best Quality*: select an SD-WAN Member with the best measured quality.
 - *Lowest Cost (SLA)*: select the cheapest SD-WAN Member that meets a given SLA target.
 - *Manual*: manually specify an SD-WAN Member to select.
 - *Load-Balance*: load-balance across a list of SD-WAN Members or only the members that meet a given SLA target.



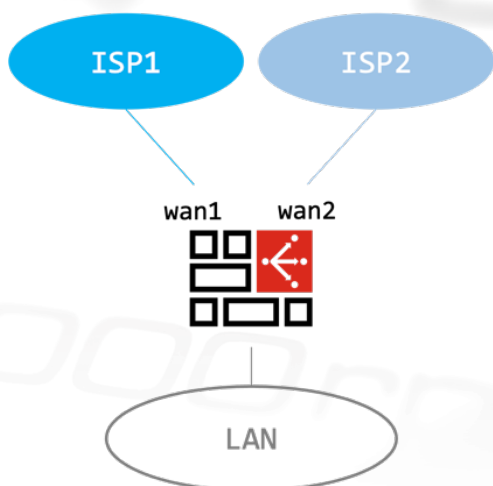
The SD-WAN rules probably remind you of Firewall Rules. Indeed, many of the same matching criteria are used, and the SD-WAN rules are also evaluated in the order of their configuration—just like the Firewall Rules. But they serve the following complementary goals:

- Firewall rules define how to secure a particular application, should a particular path be selected.
- SD-WAN rules define how to select a particular path for a particular application.

Needless to say, every outgoing session must be permitted by a Firewall Rule, after matching an SD-WAN rule!

Having both rule sets rely on the same inputs (such as Application Control Database, Internet Service Database (ISDB), same User Identity providers, and so on) greatly improves integration between traffic steering and security, making the overall solution consistent.

Once we complete this configuration, our FortiGate device (deployed at the site perimeter) becomes an SD-WAN node and is ready for intelligent traffic steering.



Any new session arriving at the FortiGate from the LAN side is evaluated against the configured SD-WAN rules and steered accordingly. Furthermore, existing sessions can switchover to a different path, should the network health conditions change.

We will soon see how this functionality applies to both underlays and overlays. But before talking about it, see [Practical basic SD-WAN examples on page 9](#) about a single FortiGate device connected to the public Internet, without any overlay network involved.

Practical basic SD-WAN examples

Let's consider a site with two Internet connections plugged into the interfaces "wan1" and "wan2" respectively (as depicted on the diagram in [Basic SD-WAN configuration on page 7](#)). What kind of SD-WAN strategies could we configure in this setup?

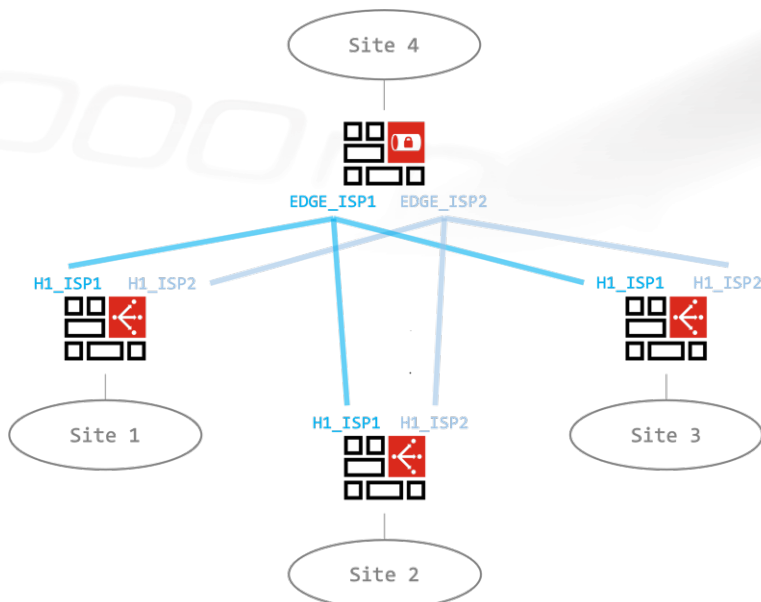
- **Best Quality.** For services particularly sensitive to high latency (such as video conferencing), we can configure an SD-WAN rule that selects a WAN link with the best measured latency among the two SD-WAN Members "wan1" and "wan2". The health of both links is being constantly probed. If the latency of the selected link degrades during the video call, the session will automatically switchover to a better SD-WAN Member, ensuring the best possible call experience.
- **Lowest Cost (SLA).** It is quite common to treat the available links as a primary and a backup connection. For example, imagine that "wan2" is connected to a mobile router, providing a costly 4G/5G Internet connection with imposed traffic limits. We would like to use this connection only as a backup. At the same time, a typical requirement is to guarantee minimal SLA for business-critical applications. Hence, if "wan1" cannot meet the SLA target, we must use "wan2".

We can configure an SD-WAN rule matching our business-critical applications and preferring "wan1" over "wan2", but only as long as it meets the required SLA target (for example, 200 ms latency). If the health of "wan1" degrades, and it can no longer meet the target, the sessions will be steered to "wan2". All the existing sessions will (by default) switchover. Still, the health of "wan1" will be constantly probed. Once the Internet connection "wan1" recovers, the application sessions will switchover back to "wan1", freeing up the backup link "wan2" and always meeting their SLA targets.

- **Load-Balance.** Another typical requirement is to guarantee minimal SLA for business-critical applications, while striving to utilize all available bandwidth (as opposed to the primary/backup model described in the previous example). We can configure an SD-WAN rule matching these applications and load-balancing them across both SD-WAN Members ("wan1" and "wan2"), but only as long as they meet a certain SLA target (for example, 200 ms). If the health of one of the Internet connections (for example, "wan1") degrades, and it can no longer meet the target, the sessions will not be steered there, and all the existing sessions will (by default) switchover to the remaining healthy Internet connection ("wan2"). Still, the health of both links will be constantly probed. Once the Internet connection "wan1" recovers, the application sessions will be load-balanced again, always meeting their SLA targets.

Overlay network

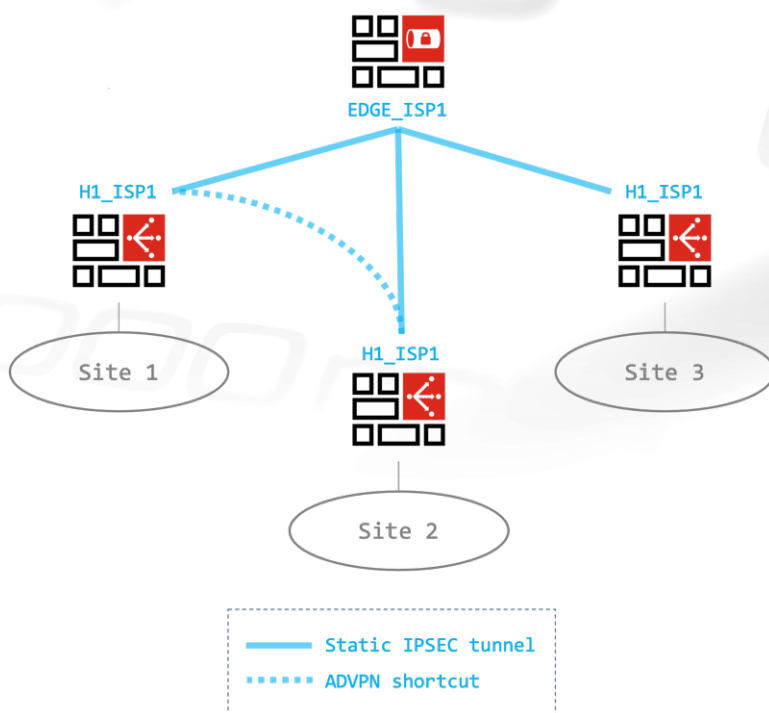
The first step in transforming a group of autonomous FortiGate devices into an SD-WAN Solution is to build an **overlay network** that interconnects all sites. FortiGates deployed at the perimeter of each site will become the SD-WAN nodes in our solution, and each FortiGate is *responsible* for the site behind it.



We will use **IPsec tunnels** to build the overlay network over all available underlay transports, forming a Hub-and-Spoke topology. This way we can secure corporate (site-to-site) traffic and provide confidentiality, integrity, and mutual site authentication, as expected from an industry-standard IPsec suite.

Hub-and-Spoke topologies are highly scalable, and they have an important Zero-Touch property: when adding or removing a Spoke, configuration of all other devices remains untouched. Hub-and-Spoke topologies can also be enhanced with redundancy options (such as Dual-Hub), and they can be extended to multiple **regions** (which are multiple Hub-and-Spoke topologies interconnected together) for large-scale deployments. We will discuss these options in more detail in [Overlay network designs on page 33](#).

Auto-Discovery VPN (ADVPN), which is our dynamic tunneling technology, is enabled in our Hub-and-Spoke topology. ADVPN dynamically builds direct Spoke-to-Spoke tunnels (also known as *shortcuts*) when they are needed, and tears down the tunnels when no longer in use. It maintains a Zero-Touch property of a Hub-and-Spoke and helps avoid the complexity of a Full-Mesh, while preserving the benefits of direct site-to-site communication without bottlenecks.



An SD-WAN/ADVPN Hub-and-Spoke topology is the most fundamental building block of our SD-WAN Solution. All the overlay tunnels are configured as the SD-WAN Members, in addition to the underlay interfaces. SD-WAN rules can then be used to steer corporate traffic, applying the same strategies we have briefly discussed in the previous section.



Overlays are *optional* in our SD-WAN Solution. They are the most common and secure way of interconnecting your sites. But what if you provide a **private transport** circuit fully under your control (both from routing and security perspective)? These cases are rare, but they happen.

As you have already learned, the SD-WAN functionality on the FortiGates can steer the traffic both to the overlays and directly to the underlays. Hence, you can opt to skip building the overlay network over (some of) your transports, as long as the transport network can route the traffic properly and guarantee the required levels of security.

Similarly, ADVPN is *optional* too. The SD-WAN functionality can be applied to a classic Hub-and-Spoke topology (or in any desired topology for that matter).

That being said, in this document we will stick to the overlay network built over all available underlay transports, with ADVPN enabled, as this is the most common and recommended approach.

Practical overlay examples

Let's continue an example from the previous section, where a site has two Internet connections, plugged into the interfaces "wan1" and "wan2" respectively. We will configure two overlay tunnels to the Hub, one over each Internet transport, named H1_ISP1 and H1_ISP2 respectively. Our list of SD-WAN Members now includes both the underlays ("wan1" "wan2") and the overlays (H1_ISP1, H1_ISP2).



Here "H1" stands for "Hub 1", just to have our naming ready for a Dual-Hub topology, which is very popular for redundancy reasons. We will discuss this in more detail in [Overlay network designs on page 33](#).

What kind of SD-WAN configuration could we apply, in addition to the examples from the previous section?

- Corporate traffic (site-to-site) on page 12
- Internet access on page 12

Corporate traffic (site-to-site)

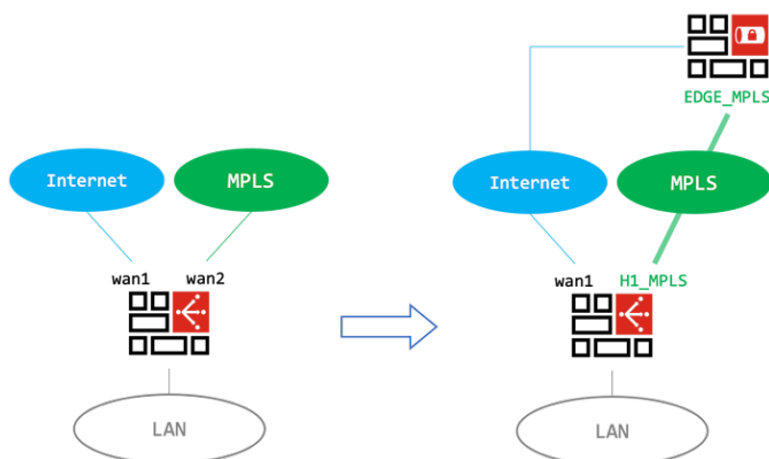
- **Corporate traffic - Lowest Cost (SLA)** It is common to treat the overlays as a Primary and a Backup, similar to the underlays. Therefore, if we prefer using the ISP1 transport, we will configure an SD-WAN rule matching our corporate (site-to-site) traffic and preferring H1_ISP1 over H1_ISP2, but only as long as it meets the required SLA target (for example, 100 ms). If the health of ISP1 path degrades, so that H1_ISP1 overlay can no longer meet the target, the sessions will switchover to H1_ISP2. Once the H1_ISP1 overlay can meet the SLA target again, the sessions will switchover back.
- **Corporate traffic - Best Quality** If there is a need to make a VoIP call between two Branch offices, a typical requirement is to select an overlay that provides the best quality at the moment. We can configure an SD-WAN rule to select the best out of the overlays H1_ISP1 and H1_ISP2. If the health of the overlays changes during the call, the audio stream will switchover, ensuring the best possible call experience.

Internet access

The same SD-WAN rule is often used to select between underlays and overlays. To see why, let's talk about different ways of Internet access in the SD-WAN Solution.

- In the examples presented in the previous section, our applications were steered to the public Internet directly through the underlays ("wan1" and "wan2"). This is known as **Direct Internet Access (DIA)**, because the traffic leaves the corporate boundaries directly at the site edge.
- An alternative way is known as **Remote Internet Access (RIA)**, in which the traffic is backhauled through a central location (normally through the Hub).

Let's look at another SD-WAN site, where we have one Internet connection (plugged into "wan1") and one MPLS connection (plugged into "wan2"). We will configure two overlay tunnels to the Hub, with one over each transport, but only one of them is relevant for this example: namely, the one over the MPLS transport, named H1_MPLS. The following diagram depicts both the underlay connectivity and the relevant part of the overlay network:



Here is a common SD-WAN strategy for this case:

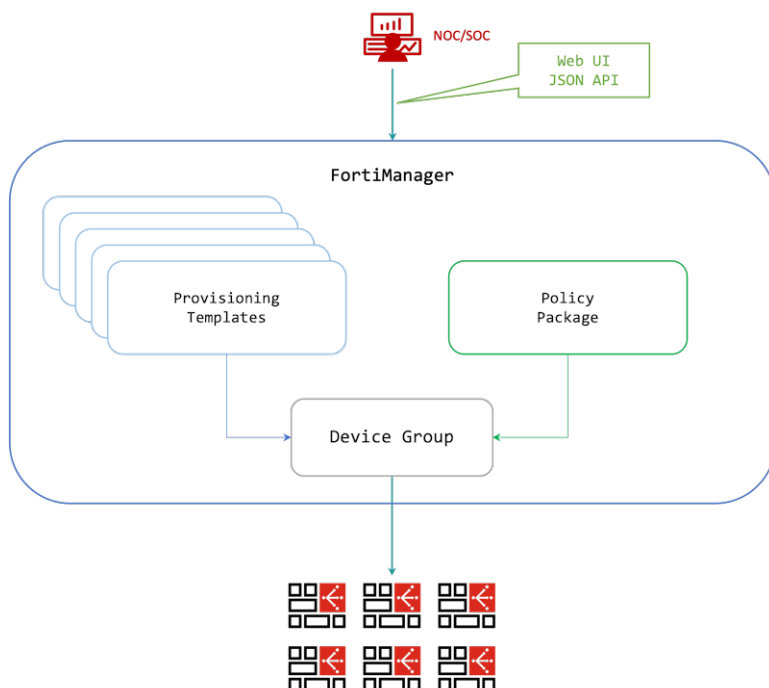
- **Internet access - Lowest Cost (SLA)** Although an MPLS connection may generally provide premium levels of service, the advantage is not straightforward when the destination is on the public Internet. It is often more efficient to breakout to the Internet directly at the site edge (DIA). Furthermore, DIA offloads the central Hub and, last but not least, it is often cheaper (reducing the bandwidth required from the MPLS network). But at the same time, we must guarantee the required SLA targets for business-critical applications.

Therefore, we may configure an SD-WAN rule to match these applications and prefer "wan1" over H1_MPLS, but only as long as it meets the configured SLA target. If the local Internet connection cannot meet the SLA target, the traffic will be backhauled through the H1_MPLS overlay (RIA). As you can see, this is yet another example of the Primary/Backup model, just in this case the selection is between an underlay and an overlay.

Provisioning and management

Since all our SD-WAN nodes are fully-functional FortiGate devices, there are plenty of ways to configure and manage them. But an SD-WAN Solution cannot be complete without defining a central tool handling site provisioning, configuration management, network monitoring, and troubleshooting.

FortiManager is the natural choice for these tasks.



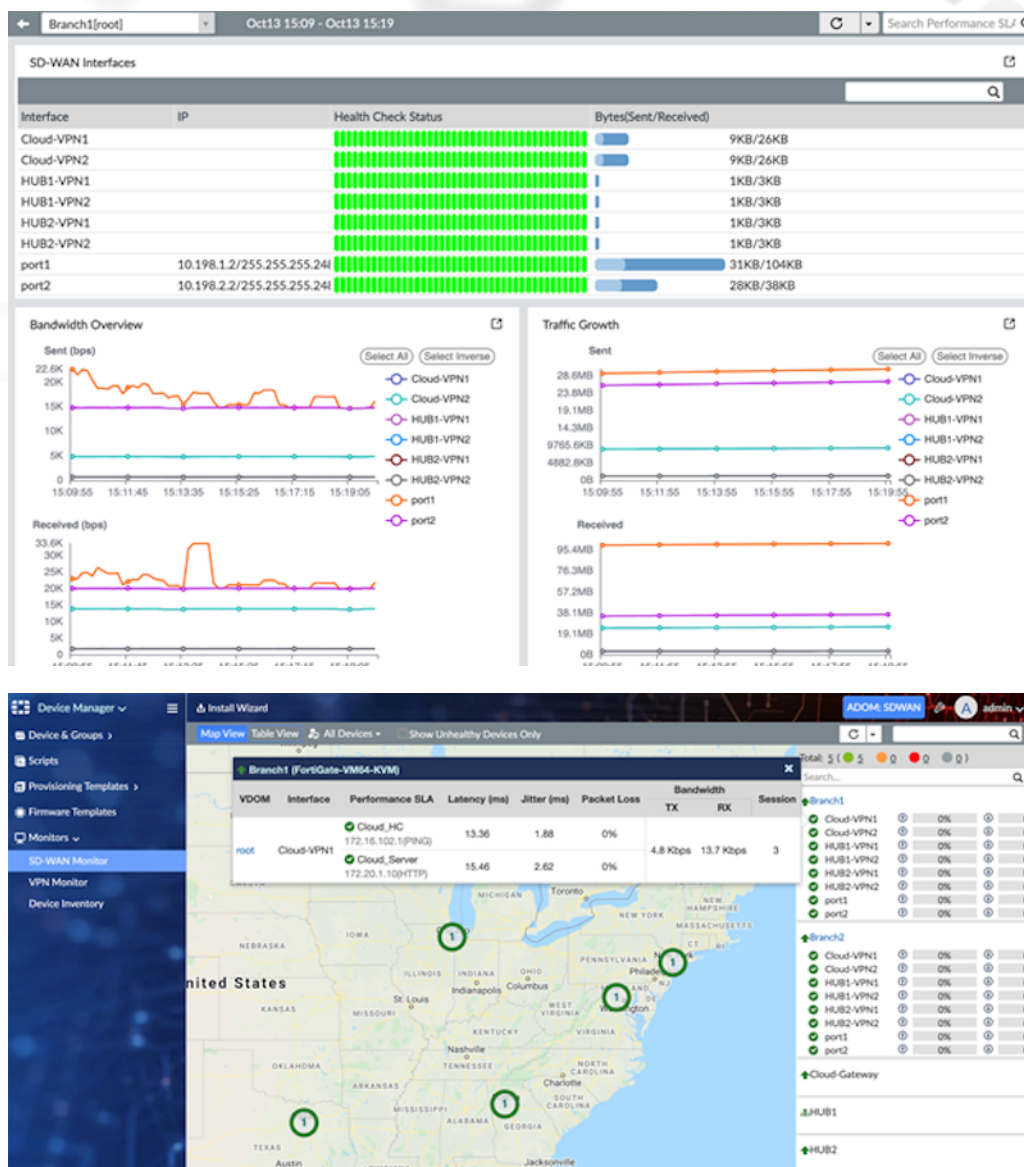
The main two entities used to describe your FortiGate device configuration on the FortiManager are:

- A set of **Provisioning Templates** that cover different parts of the configuration, including the overlay network (IPsec/BGP), the SD-WAN configuration, and so on.
- A **Policy Package** defines the Security Policy (Firewall Rules).

All your FortiGate devices are organized into **Device Groups**, with the right set of the Provisioning Templates and the right Policy Package assigned to each group.

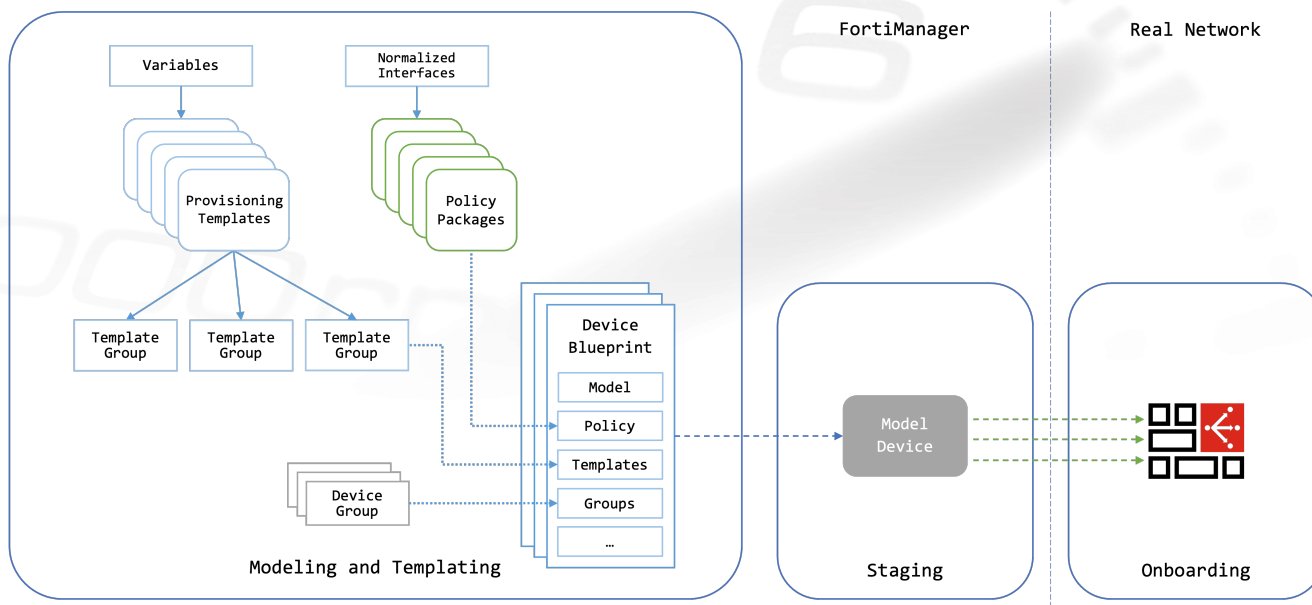
You can interact with FortiManager using the GUI or using the comprehensive industry-standard JSON API. The latter allows you to automate the entire deployment workflow and opens a wide range of third-party integration possibilities (such as, external Orchestrators, OSS/BSS/ITSM systems, and so on).

When it comes to Monitoring and Troubleshooting, FortiManager offers several tools. Most notably, the **SD-WAN Monitor** is a single console overlooking the entire SD-WAN network, with the ability to detect health degradation and dig into specific sites for further investigation.



Finally, FortiManager is a multi-tenant platform. **Administrative Domains (ADOMs)** are used to create independent environments for different tenants, each with its own sets of Provisioning Templates, Policy Packages, and so on.

Deployment workflow



The recommended deployment workflow for our SD-WAN Solution logically consists of the following stages:

- 1. Modeling and Templatng** In this stage, create the Provisioning Templates and the Policy Packages for all types of sites in your SD-WAN network. Several tools are available to help you organize your work: different Provisioning Templates are grouped into Template Groups, sites of the same type are grouped into Device Groups, and so on. Ultimately, all the elements are collected into **Device Blueprints**, which are essentially the recipes for device creation: the goal is to apply the entire configuration to any SD-WAN node by simply selecting the right blueprint for it.
- 2. Staging a Site** At this point, a new SD-WAN node is created, using the right Device Blueprint. As a result, it is added to the right Device Group, with the right set of Provisioning Templates applied to it, and with the right Policy Package assigned. The node is created as a **Model Device**—a special object in FortiManager that represents a device that is not yet online (not yet connected to FortiManager). The entire FortiGate configuration can be applied to the Model Device, as if it were a real device. Remember that you can safely modify or delete the model device without any impact to the real network.
- 3. Onboarding a Site** At this stage, a real FortiGate device connects to FortiManager, and FortiManager maps the device to its corresponding Model Device and pushes the entire configuration to the device in a process called **Auto-Link**. This is the *only* stage that has impact on the real network. Once the Auto-Link process is complete, the Model Device is converted into a real device and is fully managed by FortiManager.

As we have explained, the first two stages happen inside FortiManager and have no impact on the real network. You can perform the first two stages interactively using the GUI or in a fully automated, programmatic way using JSON API. We discuss this in more detail in our [SD-WAN Deployment for MSSPs Guide](#).

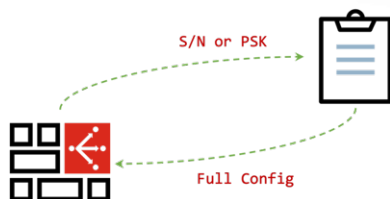
The onboarding stage can be done several ways. Each way requires different levels of human intervention that range from **Zero-Touch Provisioning (ZTP)** to a human-assisted **Low-Touch Provisioning**. The choice is mainly determined by the circumstances specific to a given environment (such as, availability of Internet access, a DHCP server, external reachability, and so on). These circumstances can differ from one SD-WAN site to another. Consequently, it is common that different onboarding methods are used for different SD-WAN sites, even within the same project.

Let us describe some of the most popular onboarding options in the [Zero-touch and low-touch provisioning on page 16](#) section.

Zero-touch and low-touch provisioning

Recall from the previous section that, once the real FortiGate device connects to the FortiManager, a process called **Auto-Link** takes place. During this process:

1. The device establishes a secure management tunnel to the FortiManager and authenticates itself, using either its Serial Number (S/N) or a Pre-Shared Key (PSK).
2. This allows the FortiManager to link the real device to its corresponding Model Device, created on the FortiManager at an earlier stage.
3. Optionally, the FortiManager instructs the real device to upgrade its firmware to guarantee compatibility with the target configuration. If this option is used, FortiManager sends the firmware image to the real device.
4. The Model Device is then converted into a real device, and the entire configuration is pushed to the real device through the management tunnel.



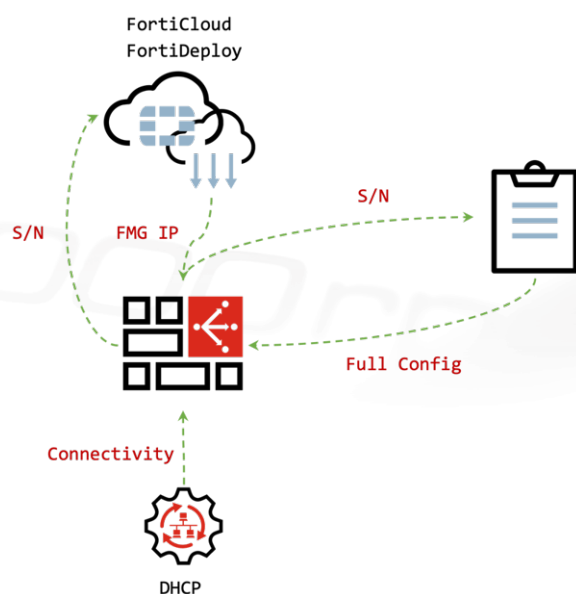
The remaining question is, therefore, how a real FortiGate device can discover the FortiManager to which it must connect. The answer to this question depends on the chosen onboarding method:

- [ZTP using FortiCloud and FortiDeploy on page 17](#)
- [ZTP using DHCP option on page 17](#)
- [ZTP using USB boot on page 18](#)
- [LTP on page 19](#)



It is important to understand that the Auto-Link process remains the same in all the onboarding methods described in this section. In other words, once the Auto-Link process starts, it no longer matters *how* the device has found its FortiManager. This allows us to mix different onboarding methods within the same project, without altering the overall deployment workflow.

ZTP using FortiCloud and FortiDeploy

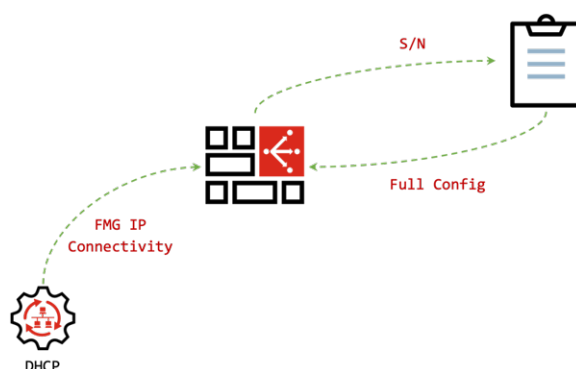


Using FortiCloud and FortiDeploy is the classic *call-home* onboarding method:

- A new FortiGate device with a factory-default configuration is plugged in and powered on.
- The FortiGate boots up and obtains Internet connectivity from a DHCP server.
- The FortiGate automatically contacts FortiCloud (in other words, the FortiGate *calls home*), authenticates using its own serial number, and obtains the location of FortiManager (based on the device registration data in FortiCloud).
- Finally, the FortiGate connects to FortiManager, authenticates using its own serial number, and triggers the Auto-Link process.

As can be seen, this method requires neither initial configuration nor any human intervention, apart from connecting the device to a power socket and to Internet access equipment (for example, an ISP modem).

ZTP using DHCP option



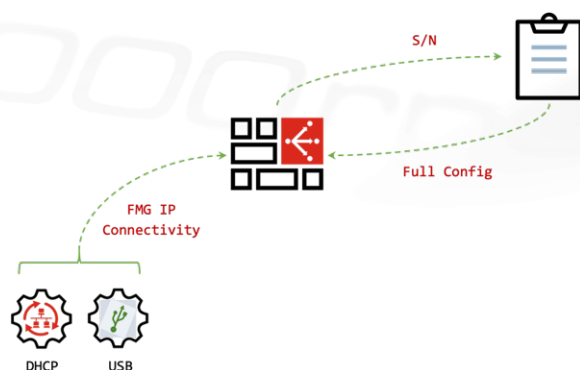
With the DHCP option, FortiCloud connectivity is not required:

- A new FortiGate device with a factory-default configuration is plugged in and powered on.
- FortiGate boots up and obtains its WAN connectivity from a DHCP server.
- The same DHCP server also provides the location of FortiManager, using DHCP Option 240/241.
- Finally, the FortiGate device connects to FortiManager, authenticates using its own serial number, and triggers the Auto-Link process.

This method is well-suited for corporate environments, where on-site network infrastructure is under your control. Therefore, it is possible to apply custom configuration to the local DHCP server, adding the FortiManager location.

As can be seen, this method requires neither initial configuration nor any additional human intervention, once the on-site infrastructure is properly configured.

ZTP using USB boot



With the USB boot method, we rely on the ability of the FortiGate to obtain its initial configuration from a USB stick:

- A new FortiGate device with a factory-default configuration is plugged in and powered on, with a USB stick that contains the necessary initial configuration plugged into it as well.
- Optionally, the USB stick can also contain the target firmware image. The device will then automatically use this image to boot up.
- Once it boots up, FortiGate loads the initial configuration from the USB stick. This configuration provides the FortiManager location, as well as any configuration required to reach it.
- Finally, the FortiGate device connects to FortiManager, authenticates using its own serial number, and triggers the Auto-Link process.

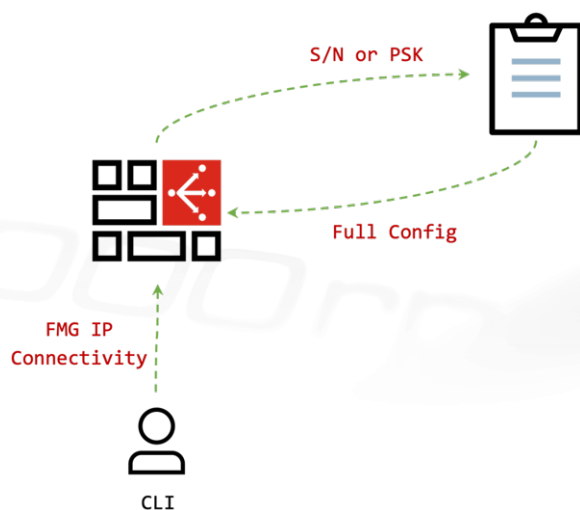
As can be seen, this method requires a USB stick to be shipped to every new site prior to the deployment. However, it proves useful in cases, where other onboarding methods are not feasible.

One additional advantage of this method is that the target firmware image can be obtained locally (from the USB stick), instead of being downloaded from FortiManager during the Auto-Link process. This is usually faster, especially on remote locations with relatively slow network access.



The USB stick should NOT contain a complete configuration of an SD-WAN node! The requirement is only to *reach* FortiManager and trigger the Auto-Link process. The complete SD-WAN configuration must be maintained by the FortiManager, as described earlier.

LTP



Finally, if none of the other methods are acceptable, the following LTP method can be used:

- A new FortiGate device with a factory-default configuration is plugged in and powered on.
- A human operator connects to the FortiGate device, manually provides the FortiManager location, optionally applies any initial configuration necessary to reach FortiManager, and then triggers the FortiManager connection.
- The FortiGate device connects to FortiManager, authenticates using either its own serial number or a PSK provided by the operator, and triggers the Auto-Link process.

Note that this is the only method that supports authenticating the device using a PSK. Therefore, it is particularly useful when the authentication based on the serial number is not feasible (for example, when the serial number is not known in advance or when the Model Devices are created on FortiManager).

Analytics and reporting

While FortiManager provides the tools necessary for real-time network troubleshooting, a centralized Analytics and Reporting tool is another crucial element that we need. Having a central point collect the logs, consolidate historical data from all the SD-WAN sites, analyze this data, and generate reports based on this analysis, those are important tasks in any network.

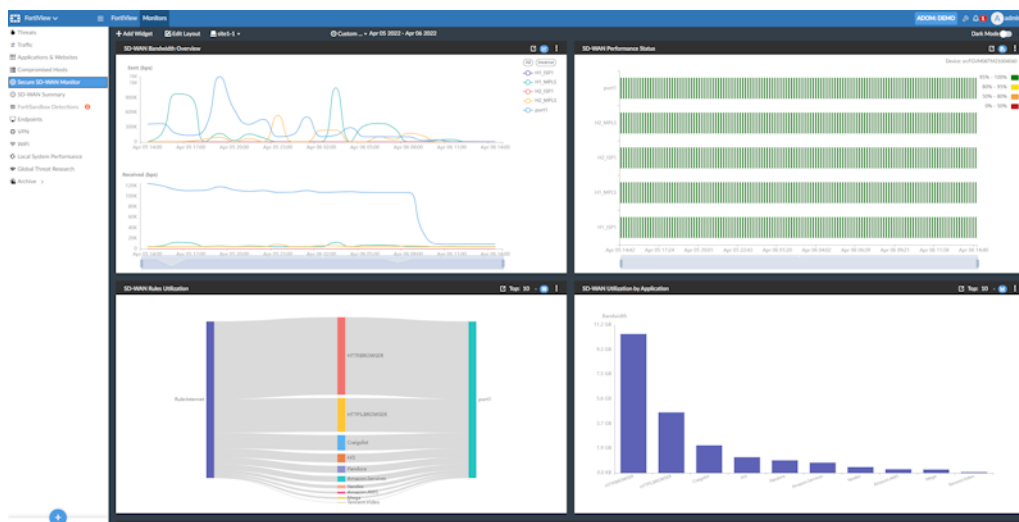
FortiAnalyzer is an integral part of our solution and is responsible for these tasks. It can be natively integrated with FortiManager, resulting in a single pane of glass for your Network and Security Operations (NOC/SOC).

The two products compliment each other: one communicates with the SD-WAN nodes using REST API and the other mainly provides log analysis. The following table provides a high-level overview of their duties:

Duty	FortiManager (API)	FortiAnalyzer (Logging)
Device uptime and availability	X	
System health monitoring	X	
Bandwidth overview	X	
FortiGate route table monitoring	X	

Duty	FortiManager (API)	FortiAnalyzer (Logging)
IPsec tunnel monitoring	X	
ADVPN monitoring	X	
SD-WAN rule selection and steering monitoring	X	
Performance SLA monitoring	X	
Application performance	X	X
Per-application bandwidth consumption		X
Single page network summary		X
SD-WAN rule and interface utilization		X
Advanced SD-WAN analytics (per device and network)		X
Pre-built or custom reporting		X
Custom log dashboard		X
Long term log storage		X

The logs collected by FortiAnalyzer are automatically analyzed and used to populate different widgets on the **FortiView SD-WAN Summary** and **FortiView SD-WAN Monitor** pages. The former provides a global view of the SD-WAN network, while the latter provides deeper historical data on specific sites.



Additionally, comprehensive customizable reports can be generated based on the analysis of all the gathered data.

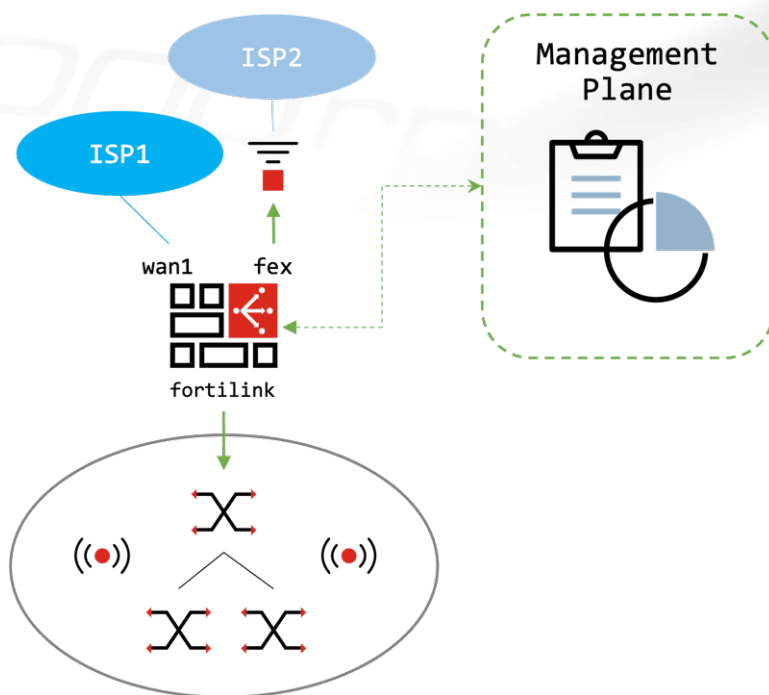
Finally, similar to FortiManager, FortiAnalyzer is a multi-tenant platform. **Administrative Domains (ADOMs)** are used to create independent environments for different tenants, each with its own disk utilization quota, data retention policies, and more.

Secure SD-Branch

It is no coincidence that we are describing the Secure SD-Branch *after* describing the management plane of our SD-WAN Solution in the previous sections. Indeed, our Secure SD-Branch can be seen as an evolution of a given SD-WAN

site, without influencing the rest of the solution components.

For any given SD-WAN site, its serving SD-WAN node (which is a fully-functional FortiGate device, as you will recall) becomes the local **Controller** of all the SD-Branch components deployed on that site. All these components run in a *managed mode*, with their configuration fully controlled by the FortiGate device, their ports seen as its logical interfaces, and their monitoring done through the same FortiGate device as well. Consequently, the entire management plane (including FortiManager and FortiAnalyzer) communicates, as before, solely with the FortiGate devices, which are then provisioning and monitoring their local SD-Branch components.



This is how an SD-WAN site can easily become an *SD-Branch site*, interconnected with other sites by the same overlay network and managed by the same management plane described in the previous sections.

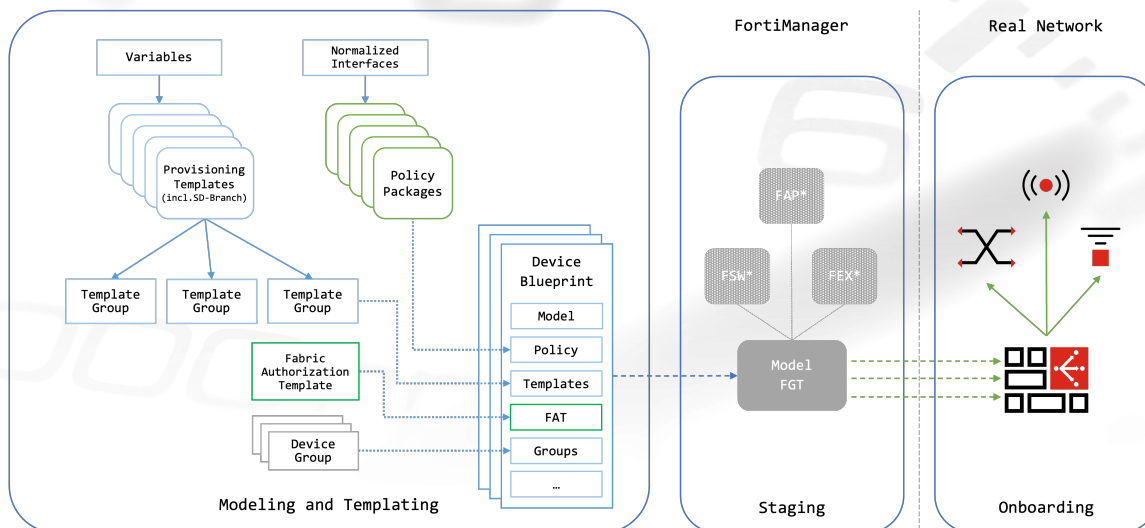
SD-Branch components

What are the main SD-Branch components comprising our solution?

- On the LAN side, the **FortiSwitch (FSW)** and the **FortiAP (FAP)** provide wired and wireless connectivity respectively. Similar to the FortiGates, a wide range of hardware models exists, allowing you to address the requirements of each particular site. Multiple devices can be interconnected in different multi-VLAN and multi-SSID Layer2 topologies, coupled with their managing FortiGate device using a proprietary **FortiLink** protocol. In this way, a wide variety of security features are extended into the LAN, including microsegmentation, 802.1X authentication policies, built-in NAC functionality, and more.
- On the WAN side, the **FortiExtender (FEX)** supplements the FortiGate WAN connectivity by adding 3G/4G/5G connectivity options. For example, it is often used to provide a backup WAN connection (with the logical FEX interface configured as an SD-WAN Member on the managing FortiGate).

SD-Branch provisioning

The recommended deployment workflow is easily extended to cover the SD-Branch solution:



As can be seen, the same stages (described in the [Deployment workflow on page 15](#) section) remain. The following changes are added:

- During the **Modeling and Templating** stage, additional Provisioning Templates are created for the SD-Branch components (FortiSwitch, FortiAP, and FortiExtender). Recall that each SD-WAN node (FortiGate) acts as a local Controller for its SD-Branch components. Therefore, from the FortiManager perspective, the configuration of the SD-Branch components is applied on the SD-WAN node itself.
- A special template is added, called Fabric Authorization Template. It generates on the SD-WAN node (FGT) a specified number of *wildcards* for the SD-Branch components (FSW, FAP and FEX).
- Finally, during the **Onboarding**, a real FortiGate device connects to the FortiManager and receives its entire configuration (including the added Provisioning Templates and the generated wildcards). This allows the FortiGate to automatically authorize and configure all of its local SD-Branch components, once they contact it.



No direct communication between FortiManager and the SD-Branch components is needed.

All the previously described onboarding options remain valid and range from a complete **Zero-Touch Provisioning (ZTP)** (of the entire SD-Branch) to a human-assisted **Low-Touch Provisioning**. At the end of this process, the entire SD-Branch is fully operational.

Monitoring and incidence response

The following products are part of our solution stack:

- [FortiMonitor on page 22](#)
- [FortiAIOps on page 24](#)

Below we summarize the parts of their functionality which are directly relevant to the SD-WAN/SD-Branch solution.

FortiMonitor

FortiMonitor is a SaaS platform that provides a full-stack performance monitoring and incident response solution. For the SD-WAN network monitoring, the following features are of particular interest:

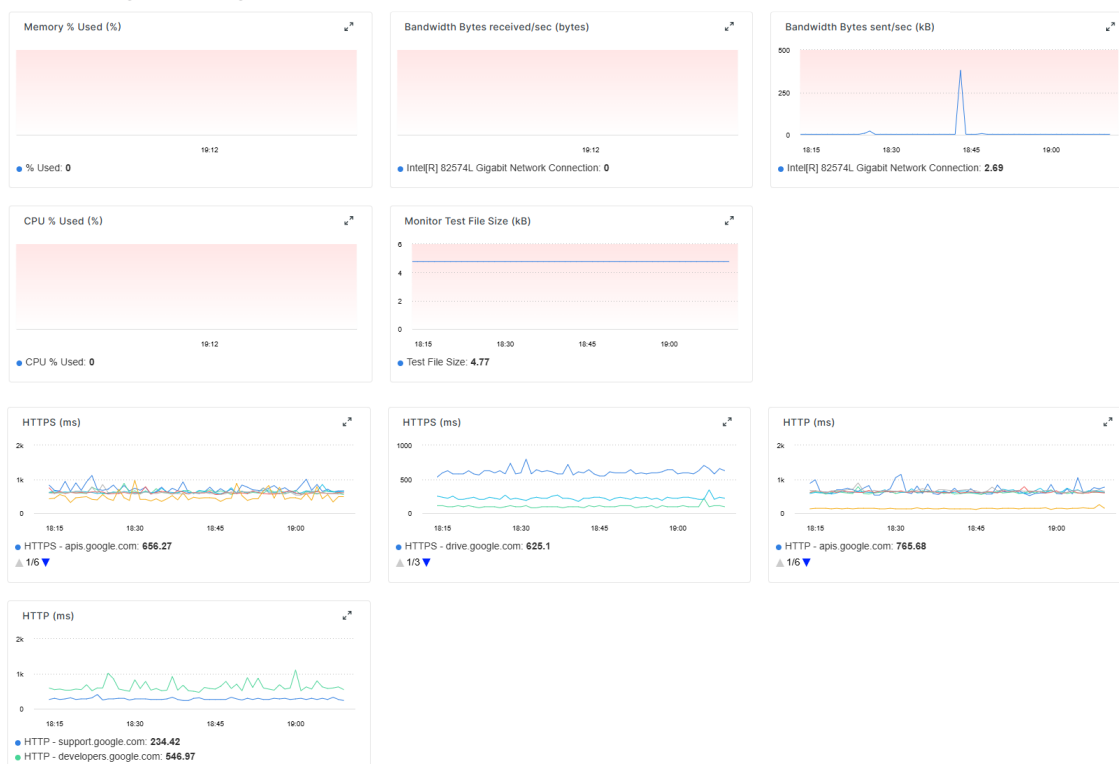
1. **Digital Experience Monitoring.** An Endpoint Agent (DEM Agent) is a local monitoring utility that is deployed directly on a Windows or a MacOS workstation.

The Agent allows monitoring of the machine's health, performance and DEM (Digital Experience Monitoring) metrics to ensure that end-user experience is optimized.

The Endpoint DEM provides assurance that the clients have good network availability, and more importantly, that the users, no matter where they connect to the network, can reach the key applications that they rely on.

Besides the standard OS health metrics, the following list of the network-related metrics is collected by the DEM agent:

- Network quality checks(Jitter, Latency, MOS, Packet loss)
- Synthetic checks for specific applications (HTTP/HTTPS transactions, traceroute)
- Download speed
- WiFi Rx (Reception) Rate
- WiFi Tx (Transmission) Rate
- WiFi Signal Strength



Additionally, there is a capability to define **custom metrics** to be collected.

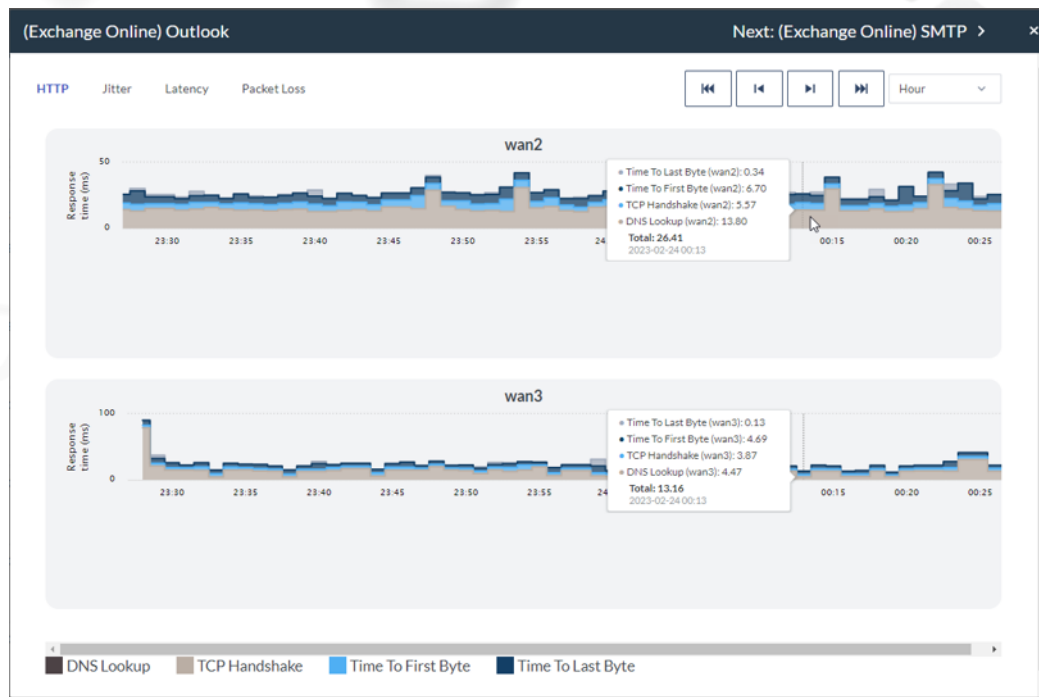
Besides the incident alerts and notifications, FortiMonitor provides a possibility to create different kinds of countermeasures, if a certain metric crosses the predefined thresholds. These countermeasures are automated responses, and they can be highly customizable (using Python and PowerShell) and comprehensive.

2. **SD-WAN Application Monitoring.** SD-WAN synthetic checks simulate a connection to an application to provide real-time metric collection. Synthetic monitoring is done per-application over SD-WAN members.

This feature is using the OnSight Collector as the originator of the application-specific probes for the individual SD-WAN members.

FortiMonitor adds visibility of the following metrics to help detecting server response issues:

- TCP Handshake
- SSL Handshake
- Time to First Byte
- Time to Last Byte
- DNS Lookup



The SLA information collected by FortiMonitor is sent back to the FortiGates as the monitored interface's SLA information. This statistics can then be used as a Performance SLA inside the SD-WAN rules on the FortiGate.

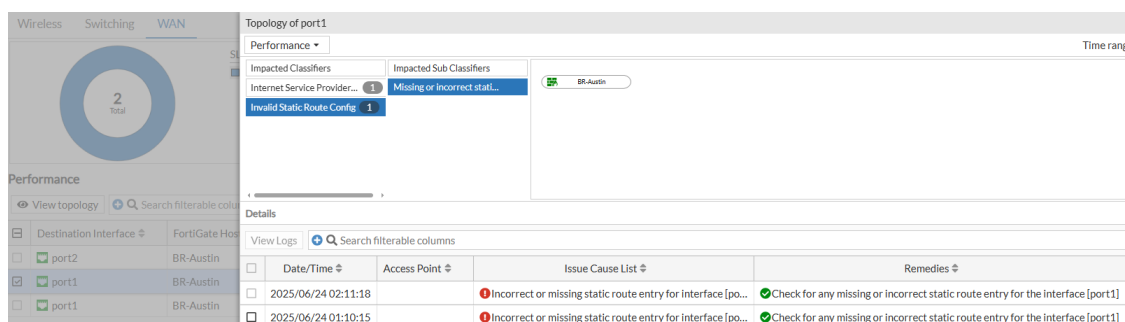
It is worth noting that apart from the functionality described above, FortiMonitor supports a range of monitoring capabilities, not limited to Fortinet devices: it can also monitor 3rd-party network devices, servers, virtualization platforms, cloud providers and so on.

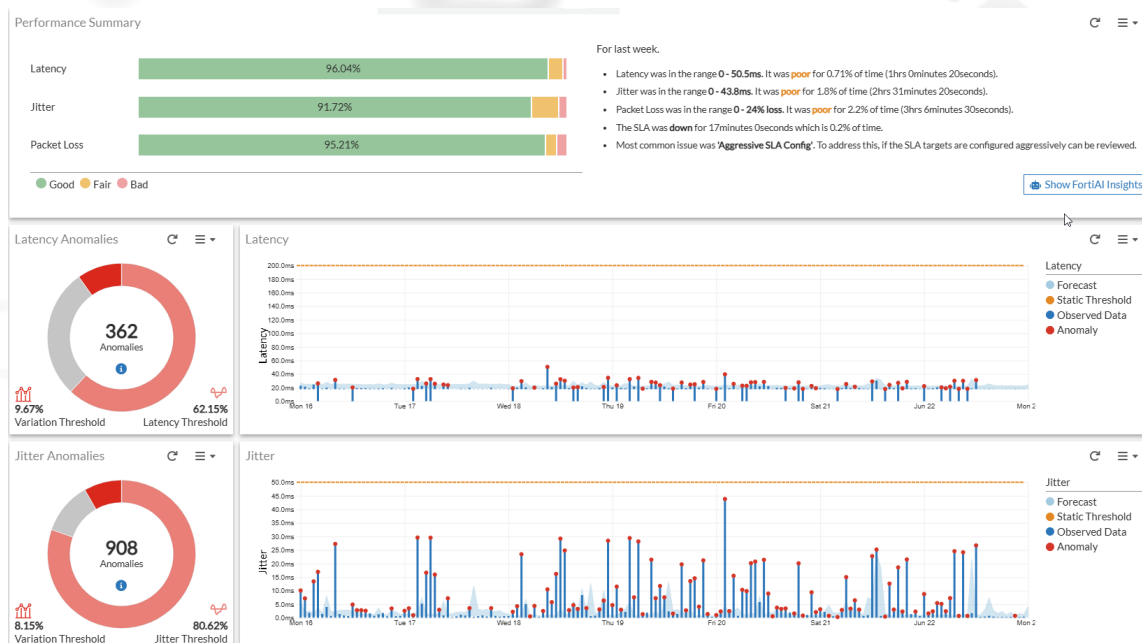
FortiAIops

FortiAIops leverages artificial intelligence with machine learning (AI/ML) to enhance network monitoring and anomaly detection

across WAN and LAN (SD-WAN/SD-Branch):

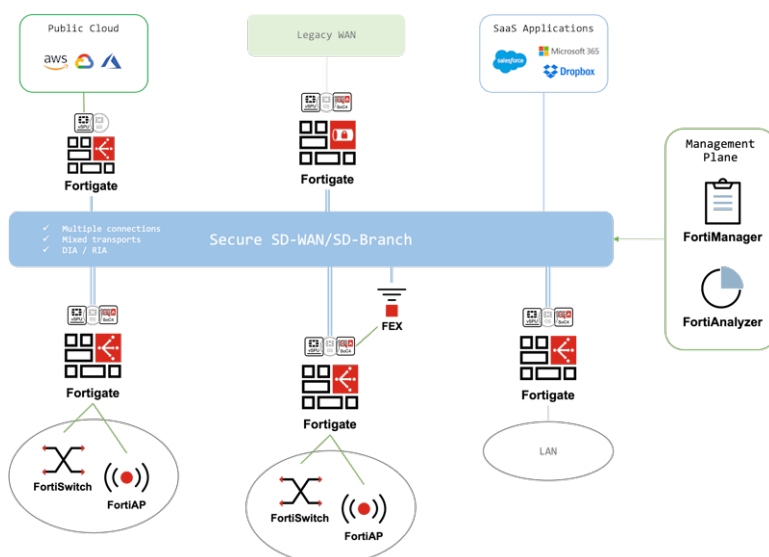
- The AI/ML is analyzing monitoring data from Fortinet network devices (FortiGates, FortiAPs, FortiSwitches, FortiExtenders).
- It collects logs either directly from the FortiGates or from the FortiAnalyzer, together with the data collected using the API calls.
- It then correlates it and analyzes the results with Machine Learning.
- After performing cross-correlation of the events and data collected from the device, it creates AI insights for the number of wireless, switching and SD-WAN SLAs.
- The AI Insights show detected anomalies and recommend fixes.
- For the SD-WAN, there is a forecast based on the historical data collected.





Summary

In this chapter, we have gradually built our Secure SD-WAN/SD-Branch Solution. The following diagram finally summarizes it as a whole:



In the next chapters, we will turn to the aspects specific to MSSPs, and then we will deep-dive into some of the technical topics briefly mentioned so far.

MSSP deployment blueprints

Now that you are familiar with the main products involved in our Secure SD-WAN/SD-Branch Solution and the roles they play, let's discuss where they can be deployed when this solution is offered as a **Managed Service**.

What should be deployed on the Customer premises? What can be deployed on the MSSP premises? Our product portfolio offers a lot of flexibility in this regard. All the mentioned products exist in different form factors, both physical and virtual, and can be deployed either in a multitenant setup or dedicated to a specific Customer.

In this chapter we discuss the blueprints most widely used by the MSSPs.

- [Spokes as Edge CPEs on page 26](#)
- [Hubs on page 27](#)
- [Management on page 29](#)
- [Monitoring and incidence response on page 31](#)
- [External orchestration on page 31](#)

Spokes as Edge CPEs

There is no doubt where to deploy the Spokes in our solution. As we have explained in the [Secure SD-WAN/SD-Branch Solution on page 7](#) chapter, the Spokes are fully functional FortiGate devices, and they are always deployed at the perimeter of each SD-WAN site. Therefore, they will be shipped to the Customer premises, becoming the **Edge CPEs** of the Managed Service.



This is why we often use the term *Edge* instead of a *Spoke*, when talking in the MSSP context.

A choice of an Edge CPE model depends on the size of each particular site and its requirements:

- How many users will it serve?
- How many WAN links are required and of what type (SFP, copper)?
- Does the site host clients, servers, or both?
- What kind of security inspection is required for the traffic entering and leaving this site?

Most Customers will require a mix of different FortiGate models to satisfy the needs of all types of sites.

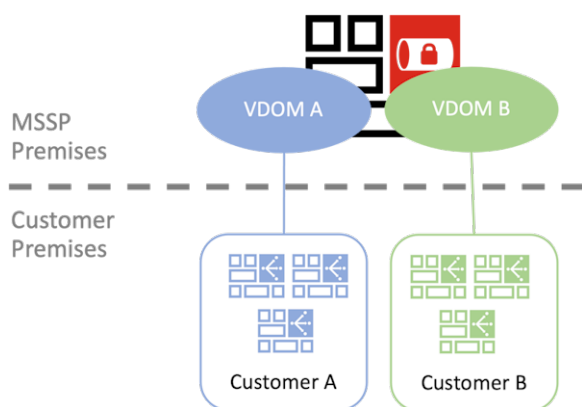
In case of an *SD-Branch site*, similar considerations will be applied to the SD-Branch components: FortiSwitches, FortiAPs, and FortiExtenders. All the components will be shipped to the Customer premises, deployed in the managed mode, and controlled by the local Edge CPE.

Hubs

Just like the Spokes, the Hubs are also fully functional FortiGate devices. But unlike the Spokes, they can be deployed either on the Customer premises or on the MSSP premises. Let's describe the following most common options:

- [Blueprint 1: MSSP premises, multitenant on page 27](#)
- [Blueprint 2: MSSP premises, no multitenancy on page 28](#)
- [Blueprint 3: customer premises on page 28](#)

Blueprint 1: MSSP premises, multitenant



The Hubs are deployed on the MSSP premises and offered as a service to the Customers. Multitenancy is configured using Virtual Domains (VDOMs), with one VDOM per Customer.

In this blueprint, the Hubs mainly perform control plane functions in the SD-WAN network. They facilitate route exchange and ADVPN operation.

All types of Customer sites are Spokes, including branch offices, datacenters, and so on. As a result, nearly all the Customer traffic is either Spoke-to-Spoke (flowing through ADVPN shortcuts) or Direct Internet Access (DIA) on the Spokes. There are often no workloads hosted behind the Hubs on MSSP premises, which reduces traffic flow through the Hubs to the minimum.

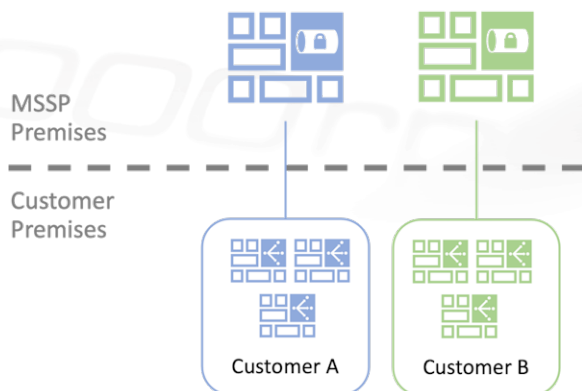
However, it would be incorrect to state that the Hubs were not part of the data plane in this blueprint. They must forward traffic in the following cases:

- Building ADVPN shortcuts is not an option. For example, when the two Spokes are not connected to any common underlay transport. Think of one Spoke connected only to the Internet and another Spoke connected only to an MPLS transport. The traffic between them will have to flow through the Hub(s).
- An MSSP offers services, such as Remote Internet Access (RIA), Cloud on-Ramp, or similar. The role of the Hubs is then to connect the Customer sites to the world outside of the SD-WAN network.
- Similar to the previous case, there may be legacy sites belonging to the Customer that have not yet migrated to the SD-WAN network. The role of the Hubs is then to provide connectivity between the SD-WAN network and those legacy sites.
- Finally, an MSSP can offer certain services hosted on its premises. From the Spokes perspective, these services will be also reachable through the Hub(s).

For these reasons, it is important to select the Hubs carefully and with the appropriate sizing.

The Hubs deployed on the MSSP premises are often called **SD-WAN Gateways**, since they play the role of interconnecting the Customer SD-WAN sites and connecting them to the world outside of the SD-WAN network.

Blueprint 2: MSSP premises, no multitenancy

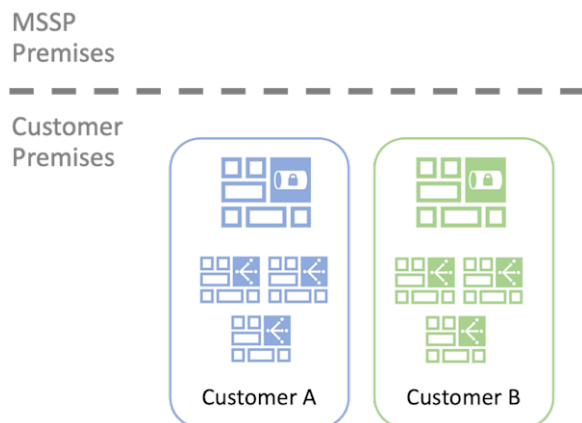


With the growing adoption of virtualized Telco Cloud environments in the recent years, another blueprint became popular. It is similar to [Blueprint 1: MSSP premises, multitenant on page 27](#), but has one difference: the Hubs are not multitenant, but rather dedicated to a specific Customer. Indeed, when an MSSP has a Telco Cloud infrastructure on its premises, it becomes reasonable to delegate the multitenancy duties to this infrastructure.

Consequently, the Hubs in this blueprint are virtual (FortiGate VM).

The rest of the points discussed in [Blueprint 1: MSSP premises, multitenant on page 27](#) apply here too.

Blueprint 3: customer premises



Another popular choice is simply to deploy the Hub(s) on Customer premises, such as at a Customer's central office or a datacenter.

The traffic flow in this blueprint is quite different from [Blueprint 1: MSSP premises, multitenant on page 27](#) and [Blueprint 2: MSSP premises, no multitenancy on page 28](#). Majority of the traffic is likely to be either Spoke-to-Hub (branch sites accessing workloads hosted in the datacenter, RIA through centralized breakout owned by the Customer, and so on) or DIA on the Spokes. Spoke-to-Spoke traffic is only occasional (for example, a voice call between two branches) or even (nearly) non-existent.

As can be seen, MSSP is not part of the overlay network in this case. (Both control plane and data plane are located at Customer premises.) It can still provide the management plane as a service, as we will see in the [Management on page 29](#) section.

This blueprint is sometimes called an *Enterprise flavor*.

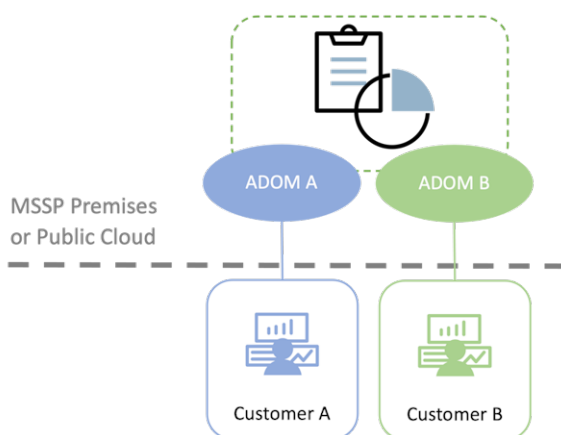
Management

Taking away the management maintenance burden from the Customer is an important benefit of the Managed Service. No wonder that the management plane of our SD-WAN Solution (including at least FortiManager and FortiAnalyzer) is almost never deployed on Customer premises.

Following are some of the commonly used options:

- [Blueprint 1: MSSP premises / public cloud, multitenant with direct customer access on page 29](#)
- [Blueprint 2: MSSP premises / public cloud, multitenant with FortiPortal on page 30](#)
- [Blueprint 3: MSSP premises / public cloud, no multitenancy on page 30](#)

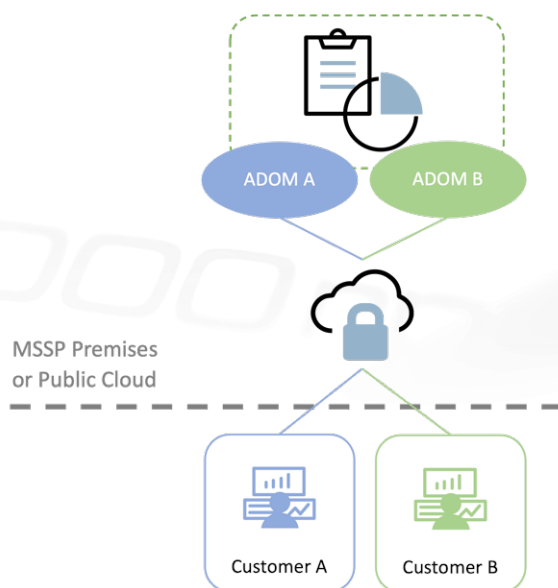
Blueprint 1: MSSP premises / public cloud, multitenant with direct customer access



Both FortiManager and FortiAnalyzer are deployed either on MSSP premises or in the public cloud (such as AWS, Azure, GCP, and others). Multitenancy is configured using Administrative Domains (ADOM), with one ADOM per Customer.

The Customer has direct access to its ADOMs to perform tasks, such as checking network status using FortiAnalyzer widgets, generating a report, and so on. Access to the device configuration and real-time troubleshooting tools (such as the SD-WAN Monitor in FortiManager) can also be provided, depending on the nature of the Managed Service (that is, depending on whether this duty is fully taken by the MSSP or not).

Blueprint 2: MSSP premises / public cloud, multitenant with FortiPortal

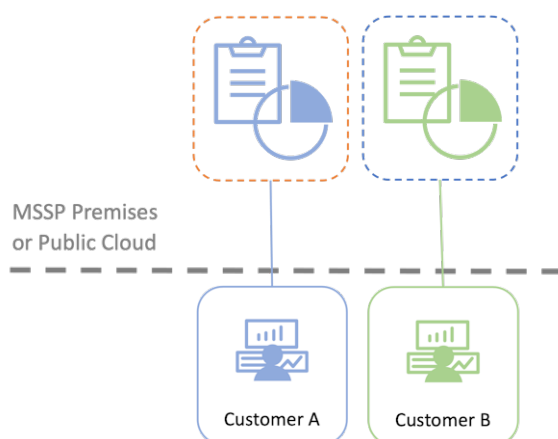


Similar to [Blueprint 1: MSSP premises / public cloud, multitenant with direct customer access on page 29](#), but Customers have no direct access to FortiManager and FortiAnalyzer. A dedicated product is used instead—**FortiPortal**.

FortiPortal is a Customer portal designed for MSSP deployments. It communicates to both FortiManager and FortiAnalyzer (using JSON API) and exposes the necessary information to the Customers. The look-and-feel can be customized by the MSSP, to make the Managed Service unique.

The Customer can use FortiPortal both for monitoring and for modifying the configuration (for example, to edit Firewall Policies, if this is allowed by the Managed Service).

Blueprint 3: MSSP premises / public cloud, no multitenancy



Finally, either FortiManager or FortiAnalyzer or both can be deployed without multitenancy. The rationale is similar to the one described for the dedicated Hubs in one of the Hubs blueprints: it may be reasonable to delegate multitenancy duties to the underlying infrastructure. In this case, it can be a Telco Cloud on MSSP premises or simply a public cloud.

Customers can have direct access to their dedicated management products, depending on the nature of the Managed Service.

Monitoring and incidence response

- **FortiMonitor** is a SaaS platform, which means that there will be no local deployment of the FortiMonitor core engine.
It supports multi-tenancy, which means that the MSSPs own a Master Tenant account, which allows them to maintain a single pane of glass view across all their customers (subtenants).
The customers (subtenants) can be allowed to access and manage their own isolated accounts.
- **FortiAIOps** instances need to be provisioned per customer. They can be installed on a variety of platforms: VMs (ESXi, KVM, Hyper-V, Nutanix), Public Cloud platforms (GCP, Azure, AWS, OCI) or hardware appliances (500G).

External orchestration

We have seen in [Blueprint 2: MSSP premises / public cloud, multitenant with FortiPortal on page 30](#) how an external product (in that case, FortiPortal) can be used to access the information from both FortiManager and FortiAnalyzer and expose it to the Customer, rather than letting the Customer access the management products directly.

When designing the Managed Service infrastructure, this idea can be taken further. The SD-WAN Solution can be fully managed using an external Orchestration system. Here we will present only a brief introduction to this topic.

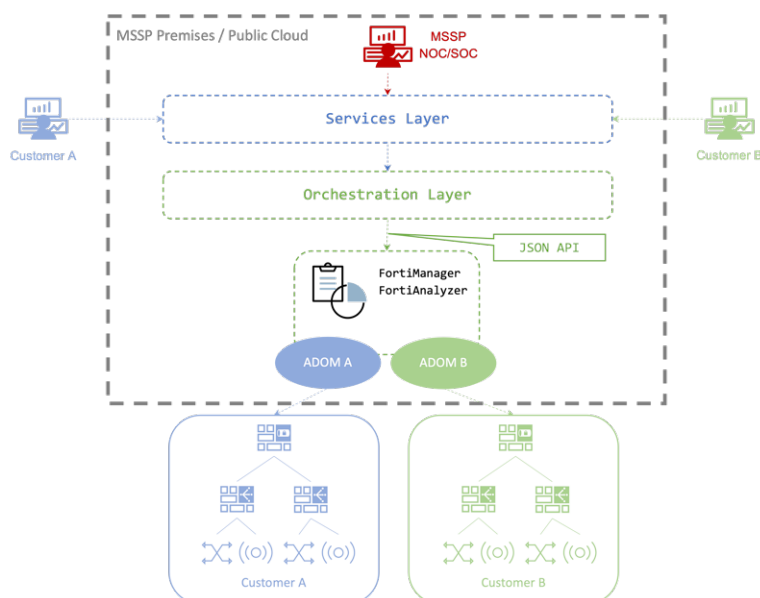
Multiple vendor-agnostic third-party Orchestrators exist on the market, and their description is outside the scope of this document. Typically they are not deployed off-the-shelf, but rather tailored to the designed Managed Service. What is important to mention is that a successful deployment of an external Orchestrator depends heavily on the following items:

1. An ability to interact with the management plane using industry-standard **API**.
Both FortiManager and FortiAnalyzer satisfy this requirement in full: nearly any action that can be done interactively can also be done through an API call.
2. An ability to abstract vendor-specific implementation details away from an external Orchestrator.

This allows to simplify and generalize the *conversations* between the management plane and the external Orchestrator, which greatly reduces the time needed to tailor the solution, to maintain it, and to extend it in the future.

The methods described in the [SD-WAN Deployment for MSSPs Guide](#) that complements this document proved to be very efficient in this regard.

The following blueprint illustrates the integration with external Orchestration systems:



Here we depict not one, but two external layers. Let's briefly describe their roles:

- **Services Layer** provides a Customer Portal for the entire Managed Service. Customer operators, as well as MSSP's own operators, will interact with the service using this portal. It can be developed in-house by the MSSP, or a customized third-party product can be used.

This layer can perform a wide variety of business tasks and interacts with different systems, such as Procurement, Billing, Logistics, and so on. For simplicity, the above diagram depicts only a single interaction vector relevant for our discussion: passing operator commands down to the Orchestration Layer. These commands will be very high-level and in line with the Managed Service definition. For example: *Add Site*, *Enable RIA*, *Get Site Health*, and so on.

- **Orchestration Layer** interacts directly with the management plane of our SD-WAN Solution using JSON API. It translates high-level commands received from the Service Layer into a series of API calls sent to FortiManager and FortiAnalyzer.

For example, an *Add Site* command could be translated into a series of API calls such as: "Create Model Device", "Set Variables", "Generate Certificate", "Install Policy", and so on.

A collection of Ansible playbooks or Python automation scripts written in-house by an MSSP is a good example of the Orchestration Layer. But also here, comprehensive vendor-agnostic third-party products exist on the market.



In this blueprint, external systems are not required to maintain your site inventory. Neither are they required to communicate to the Edge devices deployed on the Customer premises, facilitate Zero-Touch Provisioning, and so on. All these duties remain the responsibility of FortiManager and FortiAnalyzer.

Overlay network designs

In this chapter we describe several reference design flavors for the SD-WAN/ADVPN overlay network.

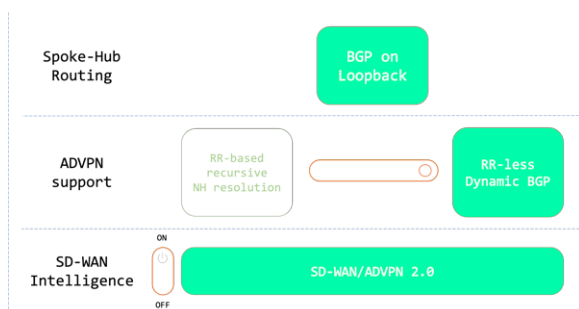
- Starting from release 7.4, we use only one foundational overlay routing design as is preferred by some of the largest service providers: [BGP on loopback on page 42](#).
- The *BGP on Loopback* design is further enhanced by the new mechanism known as *Dynamic BGP*, which greatly increases ADVPN scalability and configuration simplicity by lifting the requirement for BGP Route Reflection. While optional, *Dynamic BGP* is recommended as a default choice for most new deployments.



The combination of *BGP on Loopback* and *Dynamic BGP* is often referred to as **Dynamic BGP on Loopback**.

- The overall SD-WAN intelligence is greatly improved, thanks to the **ADVPN 2.0** framework—the new generation of ADVPN, designed for and natively integrated with SD-WAN.

The following scheme summarizes the elements. (The combination highlighted in green is recommended as the default choice for most of the new deployments.):



Finally, the choice must be made between a simple single-VRF deployment and a more advanced *multi-VRF* deployment. The latter allows you to implement end-to-end **segmentation** across the entire SD-WAN network. It is often referred to as a *Segmentation over Single Overlay*, because the same overlay network is forwarding traffic belonging to multiple segments (VRFs), while preserving their segregation across all the SD-WAN sites. All the elements mentioned in this topic support this functionality.

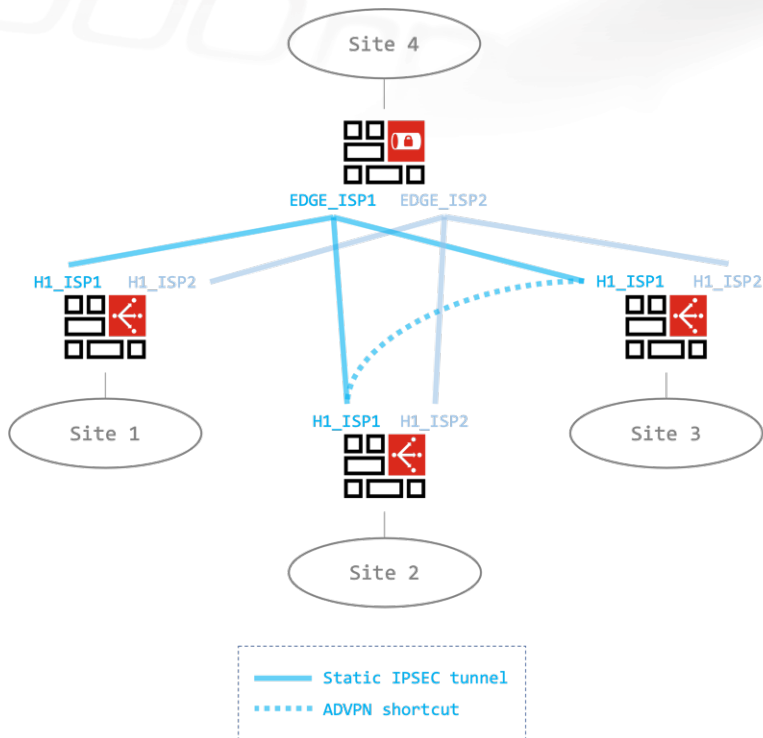
In the remaining of this chapter, we discuss all these elements in more detail.

- [Basic SD-WAN/ADVPN design on page 34](#)
- [ADVPN 2.0 on page 37](#)

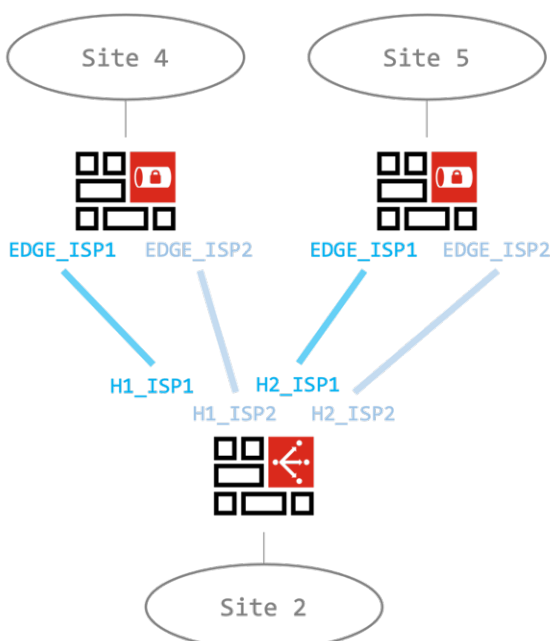
- BGP on loopback on page 42
- Segmentation over single overlay on page 53

Basic SD-WAN/ADVPN design

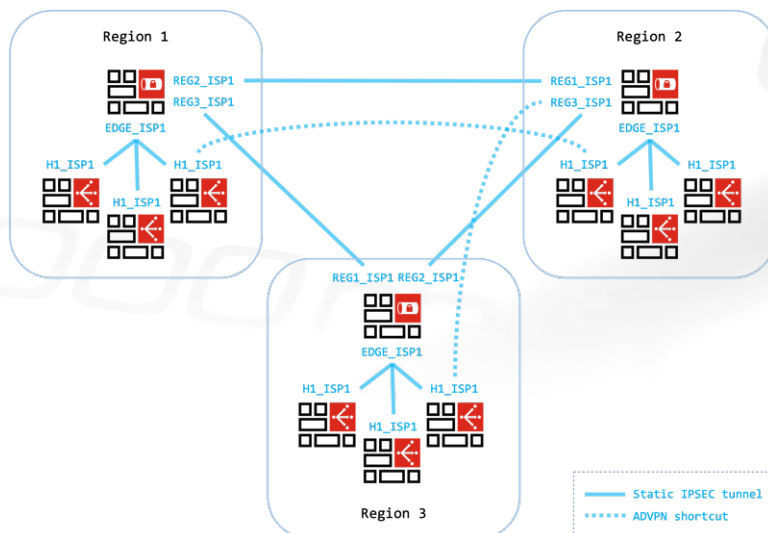
We have already noted that the fundamental building block of our SD-WAN/ADVPN solution is the Hub-and-Spoke overlay topology that securely interconnects the SD-WAN sites:



We call each Hub-and-Spoke block a **region**. Every region is typically served by either one or two Hubs. **Dual-Hub** regions are the most recommended, for redundancy reasons:



It is common for the entire SD-WAN network to consist out of a single region, but large-scale deployments (with thousands of sites) will be **multi-regional**:



ADVPN can be enabled within each region, but it can also *stretch* across the regions, allowing to build **inter-regional shortcuts**, both Spoke-to-Spoke (between the Spokes in different regions) and Spoke-to-Hub (towards the Hubs serving other regions), as demonstrated on the above diagram.

Let's see how the SD-WAN nodes are configured:

- [IPsec overlay on page 35](#)
- [BGP routing on page 36](#)

IPsec overlay

Within each region, a standard ADVPN configuration is used to build a Hub-and-Spoke topology over each available underlay transport:

- The Hubs act as **Dial-Up IPsec endpoints** for the Spokes in the region they serve. A separate Dial-Up IPsec endpoint is configured on each available underlay transport. Every such endpoint defines a point-to-multipoint overlay.
- Every Spoke builds a separate **static IPsec tunnel** over each available underlay transport towards each of the Hubs. For example, in a Dual-Hub region with two underlay transports (for example, Internet and MPLS), every Spoke will build four static IPsec tunnels.
- ADVPN is enabled on all Hubs and Spokes.

In order to interconnect multiple regions, the Hubs build IPsec tunnels between them:

- Usually those are **static IPsec tunnels**, forming a Full Mesh, although alternative topologies can be considered when the number of regions becomes too large.
- Optionally, ADVPN can be enabled (in *forwarder* mode) on all the Hub-to-Hub tunnels to allow inter-regional shortcuts.
- The Hub-to-Hub tunnels can be built either over all available underlay transports or only over some of them. There is no strict relation between the transports used within each region and those used for the Hub-to-Hub connectivity.

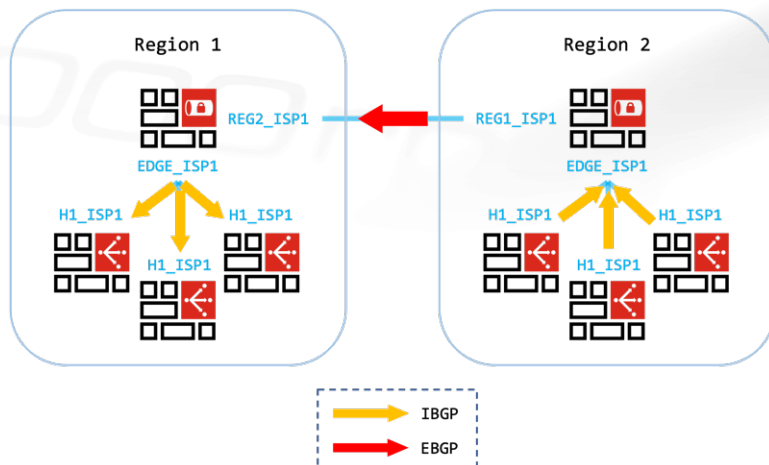


This flexibility is made possible by the ADVPN 2.0 framework. We discuss it in more detail in a dedicated section.

BGP routing

Once the overlay network is built, routing information must be exchanged between all the SD-WAN nodes (Hubs and Spokes), to ensure site-to-site reachability. We use **BGP** for this purpose—an industry-standard dynamic routing protocol, famous for its flexibility and rich feature set.

On a very high level, the routing information is exchanged as follows:



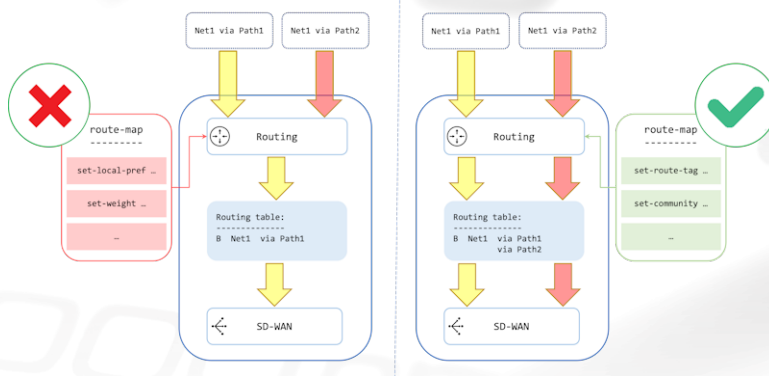
The following guidelines apply:

- A single BGP Autonomous System number (AS) is assigned to each region.
- Every Hub uses Dial-Up BGP configuration (`neighbor-group` feature).
- Every Spoke establishes an IBGP session towards each of the Hubs and advertises its LAN prefixes.
- The Hubs can (optionally) act as BGP Route Reflectors (RR), to advertise the routes between the Spokes. They can also summarize the routes. Alternatively, the Spokes can simply use default routes towards their Hubs, via all available overlays. We return to these options later.
- Additionally, the Hubs advertise their own LAN prefixes, as well as any destinations learned from outside of the SD-WAN network.
- While IBGP is used to propagate routing information within each region, EBGP sessions are established between the Hubs to propagate routing information between the regions.
- All the learned BGP routes are recursively resolved through all available overlays, therefore providing complete reachability information to each SD-WAN node.

The routing design is discussed in much more detail in a dedicated section. At this point, we shall conclude that, by the end of this process, every SD-WAN node learns about all available paths towards all corporate destinations. Then the SD-WAN rules can select an optimal overlay path for each application, at each given moment.

It is important to highlight the difference in the *role* of the routing in the SD-WAN network compared to a traditional network. In the traditional network, the routing is in charge of steering the traffic. It is, therefore, the responsibility of the routing to select the best path out of all available options. To achieve this, multiple route selection techniques can be used. Some are protocol-agnostic (for example, weight) and others are protocol-specific (for example, BGP local-preference, MED, AS_PATH prepending, and so on). While all these techniques remain available on a FortiGate device, we must recall that our goal is only **to learn about all available paths towards all destinations!**

Remember that the duty to steer the traffic in our solution is delegated to the SD-WAN rules! Therefore, it is (generally) not recommended to apply any route selection techniques to the routes learned through BGP within the overlay network. Rather than selecting a single best route, we would like to end up with **Equal-Cost Multi-Path (ECMP)** routes to all remote sites, through all available overlays.



As the picture above highlights, it is still perfectly valid to **tag** certain incoming routes, for example, by applying a route-tag. It can be then used as a matching criteria in your SD-WAN rules, proving useful in many scenarios. Note the difference: in this case all the routes remain in the routing table and visible to the SD-WAN!

In other words, what we (generally) don't want is to *hide* some of the BGP routes from the SD-WAN, because the SD-WAN uses wider range of inputs to select the best path per application at any given moment, and is therefore generally more intelligent than conventional BGP routing.



To avoid any confusion, it is still perfectly valid to apply any above-mentioned route selection techniques at the **boundaries** of the SD-WAN network, when learning or advertising routes from/to the outside world.

ADVPN 2.0

In this section we look into the new, intelligent framework called SD-WAN/ADVPN 2.0 or simply **ADVPN 2.0**.

This is a new generation of ADVPN designed for SD-WAN and natively integrated with it. Its basic aim—just as that of the earlier version of ADVPN—is to dynamically build direct IPsec tunnels (known as **shortcuts**) between Spokes, triggered by Spoke-to-Spoke traffic. When the traffic between the two Spokes in question stops, the shortcuts between them are automatically torn down.

Compared to the earlier version of ADVPN, the new ADVPN 2.0 implements a far stronger intelligence to dynamically decide *when* and *what* shortcuts to build. This intelligence is distributed across the entire SD-WAN network, with each SD-WAN node being responsible for the shortcut management for all the sessions originated behind it. The responsible SD-WAN node analyzes multiple sources of information, including local and remote topology and health status, local SD-WAN Rules and more.

In a nutshell, ADVPN 2.0 consists of three control-plane mechanisms:

1. **Discovery.** The originating node discovers the remote node. It learns about its topology and the current health status of all its participating SD-WAN Members.
2. **Path Selection.** After the discovery, the originating node locally selects an optimal shortcut and triggers it.
3. **Health Updates.** Periodic health updates are sent over the active shortcuts, allowing the Path Selection to revisit its previous choice on per-rule basis, possibly triggering new shortcuts.

The following sections describe some important aspects of the ADVPN 2.0 operation:

- [Transport groups on page 38](#)
- [Health measurement on page 38](#)
- [Shortcut management on page 39](#)

- SD-WAN node reachability on page 41
- Routing design on page 42

Transport groups

Consider an SD-WAN topology where the SD-WAN sites have a mixture of Internet and MPLS connections:

- The Internet and the MPLS transports are physically separate from each other. We often call them **segregated transports**.
- All the Internet connections, on the other hand, are naturally interconnected by the public Internet, even if they are of different types (for example, a broadband Internet connection, a backup LTE connection, and so on). Hence, we often call them **interconnected transports**.

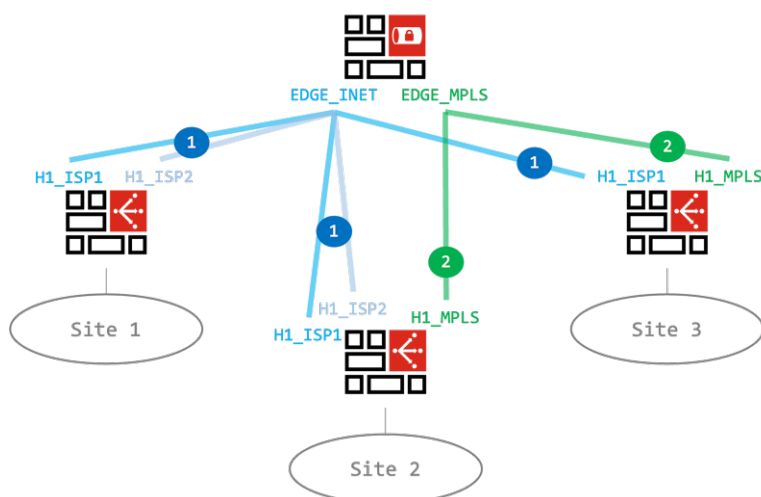
We can make the following observation: a Spoke cannot build an ADVPN shortcut across two segregated transports, but it can do so across the interconnected transports. For example, a Spoke cannot build an ADVPN shortcut from its Internet WAN link to another Spoke's MPLS WAN link or the reverse. But it can easily build an ADVPN shortcut from its broadband Internet link to another Spoke's LTE link.



In the [Additional topics on page 59](#) section, we discuss other differences between segregated and interconnected transports.

To express this distinction, ADVPN 2.0 introduces a concept of a **transport group**. Each set of interconnected transports forms a separate transport group. When a user configures an SD-WAN/ADVPN 2.0 Member, it specifies its transport group ID, so that any two members corresponding to segregated transports will be assigned to *different* transport groups. This information is exchanged as part of the Discovery process, allowing the originating Spoke to compare its local values with those of the remote Spoke. The Path Selection mechanism will never attempt to build a shortcut between members belonging to different transport groups.

Following is an example of the transport group assignment. Note that all the Internet overlays belong to transport group 1, while all the MPLS overlays belong to transport group 2:



Health measurement

Measuring current health status of each SD-WAN/ADVPN 2.0 Member is important for optimal shortcut management.

ADVPN 2.0 relies on the following two methods:

- Every Spoke actively probes each overlay path towards its Hub(s), using the standard Performance SLA configuration. The standard recommendation is to configure a dedicated loopback interface on each Hub for the health probing using Ping protocol. Later, these health measurements are exchanged between Spokes during the Discovery process, and they serve as an *estimation* of the health status during the Path Selection.



While the health measured between each Spoke and its Hub does not necessarily correlate to the health of the actual Spoke-to-Spoke path, it is nevertheless a good estimation. It is capable of detecting both local and remote connection issues and helps the Path Selection mechanism avoid triggering shortcuts over unhealthy WAN links.

- Once a shortcut is built, an additional health probing is automatically activated over it (known as *ADVPN shortcut monitoring*). This probing (also using Ping protocol) is more accurate, because it measures health of the actual path between the two Spokes' loopbacks.

Both measurements are considered by the Path Selection mechanism. For example, even if both Spokes report good health between each one of them and its respective Hub, a shortcut between them might be found out of SLA. In this situation, the Path Selection mechanism will attempt to trigger another shortcut.

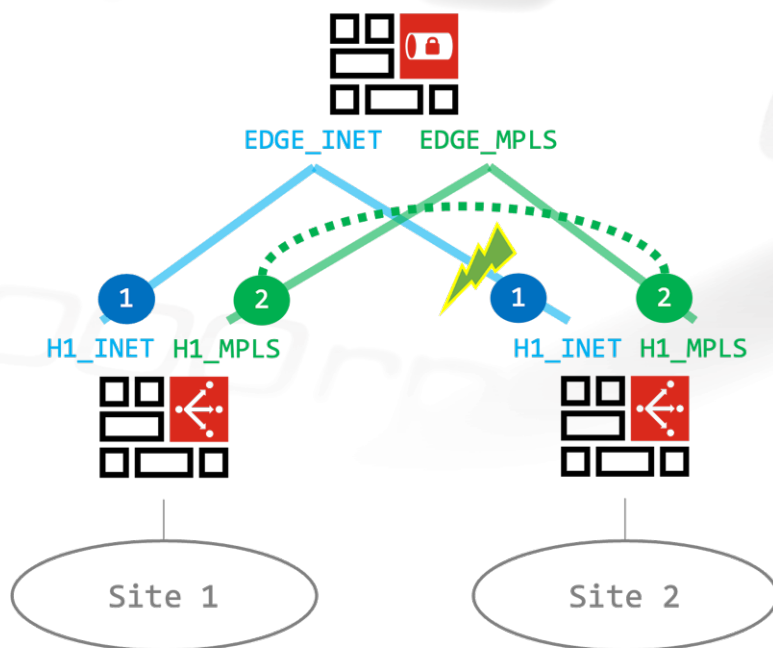
Shortcut management

As in any Hub-and-Spoke topology, Spokes have virtually no knowledge about each other, as long as there is no traffic flow between them. When user traffic starts flowing between a pair of Spokes, initially it flows through the Hub(s).

But it also triggers the Discovery process. The two Spokes exchange information between them, and the originating Spoke learns the topology of the remote Spoke, including the transport groups and the current health status of its WAN connections. The Path Selection mechanism can then trigger an optimal shortcut between the two Spokes, combining the following pieces of information:

- The local WAN connections, their transport group IDs, and current health status (as measured by the local Spoke probing its Hub)
- The remote WAN connections, their transport group IDs, and current health status (as measured by the remote Spoke probing its Hub)
- The local SD-WAN Rule matched by the user traffic, dictating the desired steering strategy and other parameters

For example, consider two SD-WAN sites, each having an Internet connection and an MPLS connection:



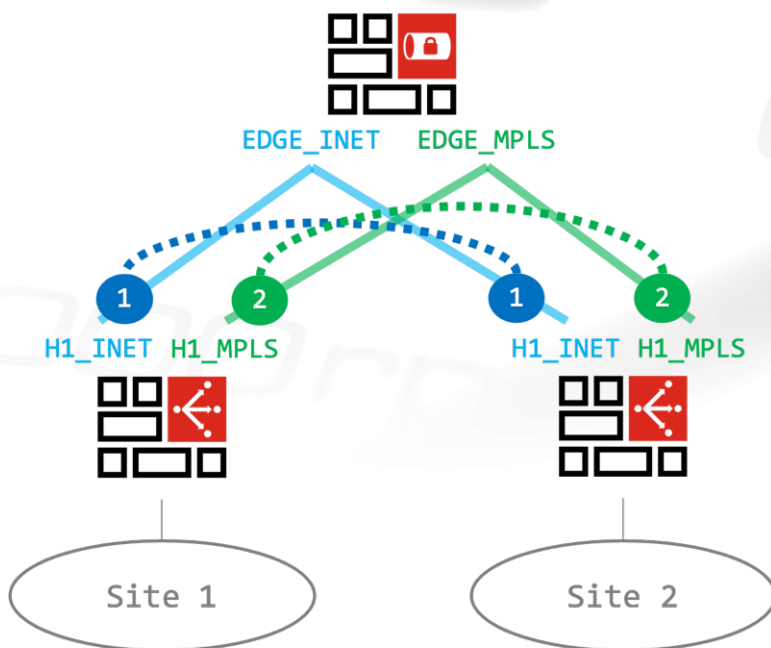
SD-WAN Rules:

Corp. Traffic => H1_INET, H1_MPLS

As the above diagram shows, the Internet link on the remote Spoke is currently out of service (or out of SLA). This information is communicated to the originating Spoke during the Discovery process. As a result, the originating Spoke cannot trigger a shortcut within transport group 1, even though this would be preferred by the matched SD-WAN Rule. Hence, the Path Selection mechanism makes a decision to trigger the MPLS shortcut, within transport group 2.

After the first shortcut is triggered, Health Updates will be periodically sent over it, updating the previously exchanged information as necessary. Also, as explained in the previous section, a more accurate end-to-end health status will be now measured over the shortcut, adding another piece of information to the Path Selection intelligence. This may cause the originating Spoke to trigger another shortcut, if and when it is deemed optimal.

In our example, when the Internet link becomes available (and healthy), the originating Spoke can trigger a new Internet shortcut, within transport group 1, to respect the preference expressed in the matched SD-WAN Rule:



SD-WAN Rules:

Corp. Traffic => H1_INET, H1_MPLS

The Health Updates continue to be exchanged, and the Path Selection mechanism continues to revisit its choices, as long as at least one shortcut remains active between the two Spokes. When the user traffic between the two Spokes stops, all the shortcuts will be eventually torn down (based on an idle timeout), and the Spokes will flush all the information previously learned about each other, effectively returning back to the initial state.

SD-WAN node reachability

For the correct operation of ADVPN 2.0 mechanisms, the SD-WAN nodes must be able to communicate with each other across the overlay network. This is required for the Discovery process, but also for a subsequent triggering of additional shortcuts.

There are no constraints on a specific path that the ADVPN 2.0 control messages take on their way between the nodes, as long as they reach their destination through the overlay network. Typically, they will be routed through one or more Hubs on their way between Spokes. Neither the results of the Discovery process nor the selected shortcut path depend on the route taken by the control messages.



This is unlike the earlier version of ADVPN, where the route taken by the control messages determined the shortcut path selection. This decoupling, implemented in ADVPN 2.0, eliminates the need for a *overlay stickiness* policy (common in the earlier SD-WAN/ADVPN deployments) and allows greater flexibility, both in a single region and in multi-regional deployments.

Routing design

The ADVPN 2.0 mechanisms are independent from the routing design. However, the routing design must properly perform its direct duties, in order to guarantee correct traffic steering with and without the shortcuts.

Let us summarize these duties:

- **Reachability for user traffic.** All the LAN prefixes must be reachable across the entire SD-WAN overlay network. This is the basic requirement for any routing design.
- **Shortcut steering.** These are the basic requirements for the ADVPN support (unchanged compared to the earlier version of ADVPN):
 - When no shortcuts exist towards a given remote Spoke, all its LAN prefixes should be resolved through the Hubs, to guarantee initial reachability once the user traffic starts flowing.
 - Whenever a shortcut is built, all the LAN prefixes belonging to the remote Spoke in question must be automatically resolved through the new shortcut, so that the traffic can be steered through it.
 - When multiple, active shortcuts exist towards the same remote Spoke, its LAN prefixes must be resolved through all of them, to allow the SD-WAN intelligence to select an optimal path at any given moment.
 - When all the shortcuts towards a given remote Spoke are torn down, the routing must automatically revert to the initial state—that is, its LAN prefixes must be resolved via the Hubs.
- **Loopback reachability.** All the SD-WAN node loopbacks must be reachable across the entire SD-WAN overlay network. This is the only requirement specific to ADVPN 2.0. It guarantees that the SD-WAN nodes (mainly the Spokes) can communicate with each other, for example, when triggering additional shortcuts between them.

In the next section we demonstrate how our recommended routing design performs all these duties.



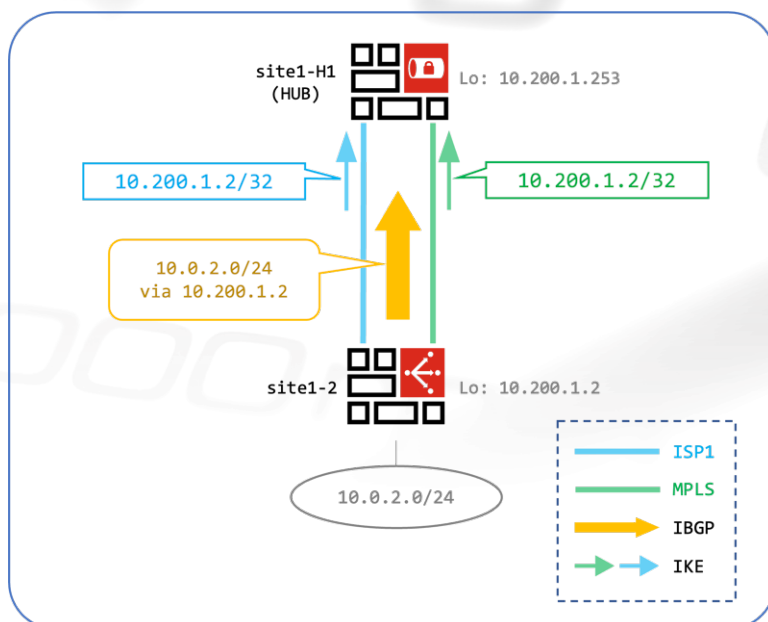
The FOS CLI configuration for ADVPN 2.0 can be found in [ADVPN 2.0 on page 78](#).

BGP on loopback

As we have already mentioned, our overlay routing design is called *BGP on Loopback*.

- Every Spoke establishes a single IBGP session towards each of the Hubs serving its region.
- This IBGP session is terminated on the loopback interface, which uniquely identifies each SD-WAN node (Hub and Spoke).
- The Spoke advertises its LAN prefix(es) over this single IBGP session per Hub.

The following diagram illustrates a LAN prefix 10.0.2.0/24, which is advertised by a Spoke “site1-2” to its Hub “site1-H1”:

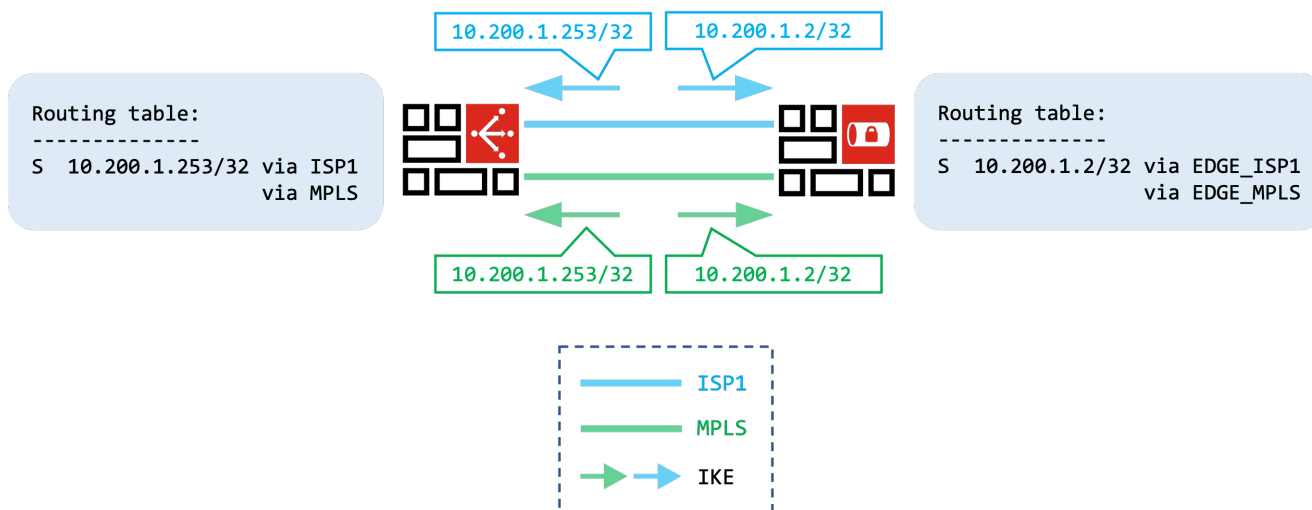


- As can be seen, a single BGP route is generated for each prefix (in our example: 10.0.2.0/24).
- BGP next-hop (NH) is the loopback IP of the originating Spoke (in our example: 10.200.1.2).
- In a Dual-Hub region, a copy of this route is advertised to each Hub.

To fully understand how this route is resolved, we must elaborate on a proprietary extension that has been implemented to support this design. See [IKE extension: exchange-ip-addrv4](#) on page 43.

IKE extension: exchange-ip-addrv4

This feature allows IKE to inject a custom /32 route on both sides of an IPsec tunnel:



In [BGP on loopback on page 42](#) design, we use it to provide loopback reachability without the use of additional routing protocols. It injects the /32 loopback route both over static IPsec tunnels (Spoke-to-Hub) and over dynamic ADVPN shortcuts (Spoke-to-Spoke).

In the above example, it helps the Hub “site1-H1” to resolve the route towards 10.0.2.0/24, by injecting the loopback route 10.200.1.2/32, which serves as a BGP NH. The same happens in the opposite direction: the Hub’s loopback route is injected on the Spoke, allowing it to resolve the routes advertised by the Hub.

As for the ADVPN shortcut support, we will see the resolution examples in the following topics:

- ADVPN support with dynamic BGP (RR-less) on page 44
- ADVPN support with BGP RR on page 45

ADVPN support with dynamic BGP (RR-less)

Starting from the release 7.4, we support a new method to exchange routes directly between a pair of Spokes, by dynamically building a BGP session between them over an ADVPN shortcut. This new mechanism is called **Dynamic BGP**.

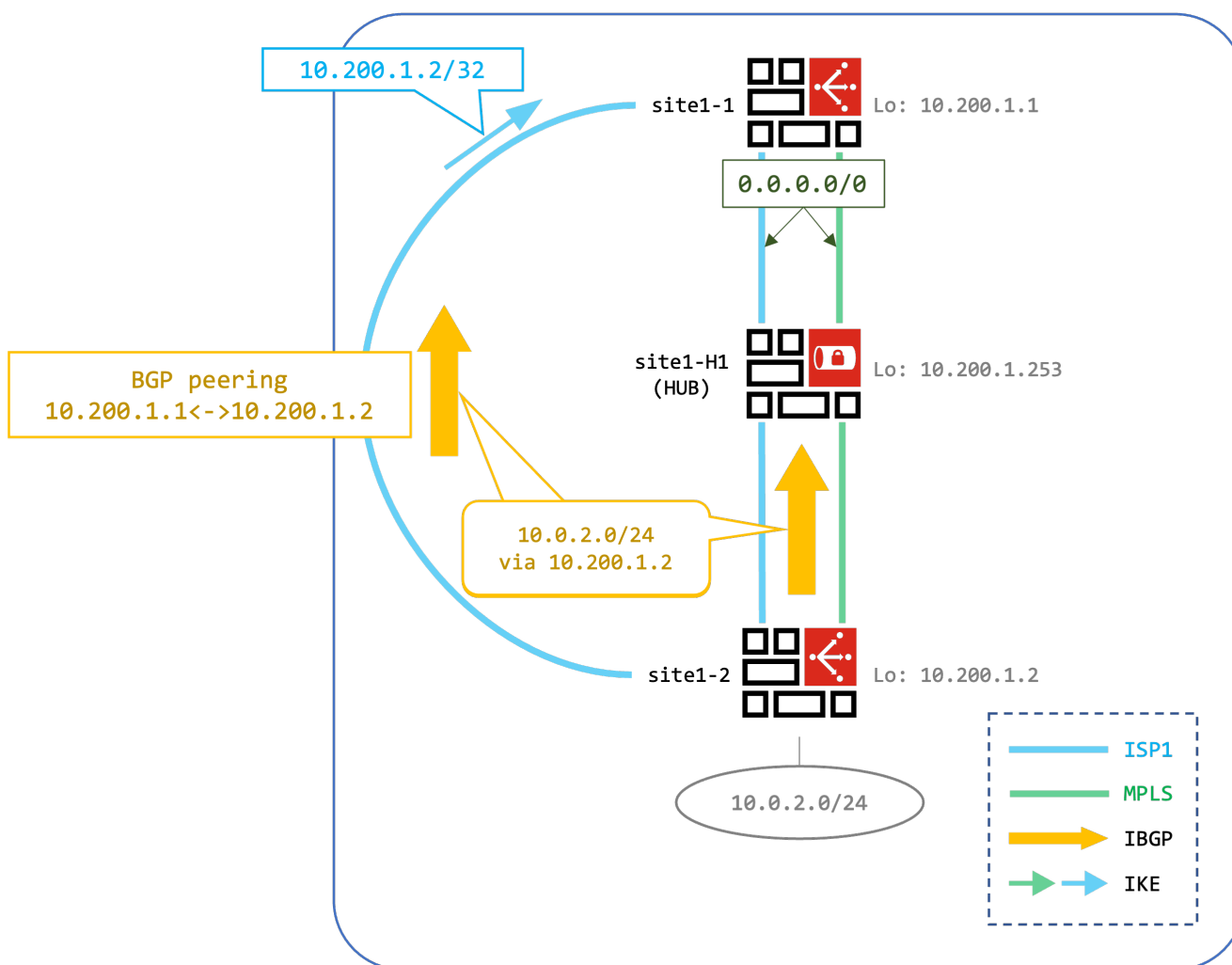
While the Hubs learn all the Spoke routes, they do not have to readvertise them to other Spokes. BGP Route Reflection (RR) on the Hubs is therefore not required. The Spokes will typically have default routes pointing to the Hubs.

Note that this simple routing configuration guarantees both reachability for the user traffic and loopback reachability across the SD-WAN overlay network, which are the two requirements listed in the [Routing design on page 42](#) section.



Instead of the default routes, a combination of a LAN summary and a loopback summary can be used on the Spokes. It can be either configured statically or advertised by the Hubs. As long as it satisfies the two reachability requirements mentioned above (by correctly routing the traffic via the Hubs), the rest of the design will not change.

The following diagram illustrates how the LAN prefix 10.0.2.0/24 from our previous example is advertised by the Spoke "site1-2" to another Spoke "site1-1", when there is an active ADVPN shortcut between them:



- As can be seen, when the ADVPN shortcut is established, the already familiar IKE feature (`exchange-ip-addrv4`) injects a /32 route towards the loopback IP of the remote Spoke.
- The two Spokes then establish a direct BGP session between them, over the shortcut, terminated on the loopback interfaces on both sides.
- This is where the LAN prefix 10.0.2.0/24 is advertised, and its BGP NH can be successfully resolved through the shortcut (thanks to the same /32 loopback route injected by IKE).

Here is the route resolution example for this case:

B 10.0.2.0/24 via 10.200.1.2

=> S 10.200.1.2/32 via **ISP1_0**

- When multiple, active shortcuts exist between the same pair of Spokes, only a single dynamic BGP session will be established. The exchanged routes will be recursively resolved through all the shortcuts, since the /32 loopback routes are injected via all of them.

Here is the route resolution example when two, active shortcuts exist:

B 10.0.2.0/24 via 10.200.1.2

=> S 10.200.1.2/32 via **ISP1_0**

=> S 10.200.1.2/32 via **MPLS_0**

- Finally, when all the active shortcuts are torn down, the BGP routes are withdrawn. The Spokes return to their initial state, where all remote LAN prefixes are reachable via the Hubs, thanks to the default routes (or the LAN summaries).



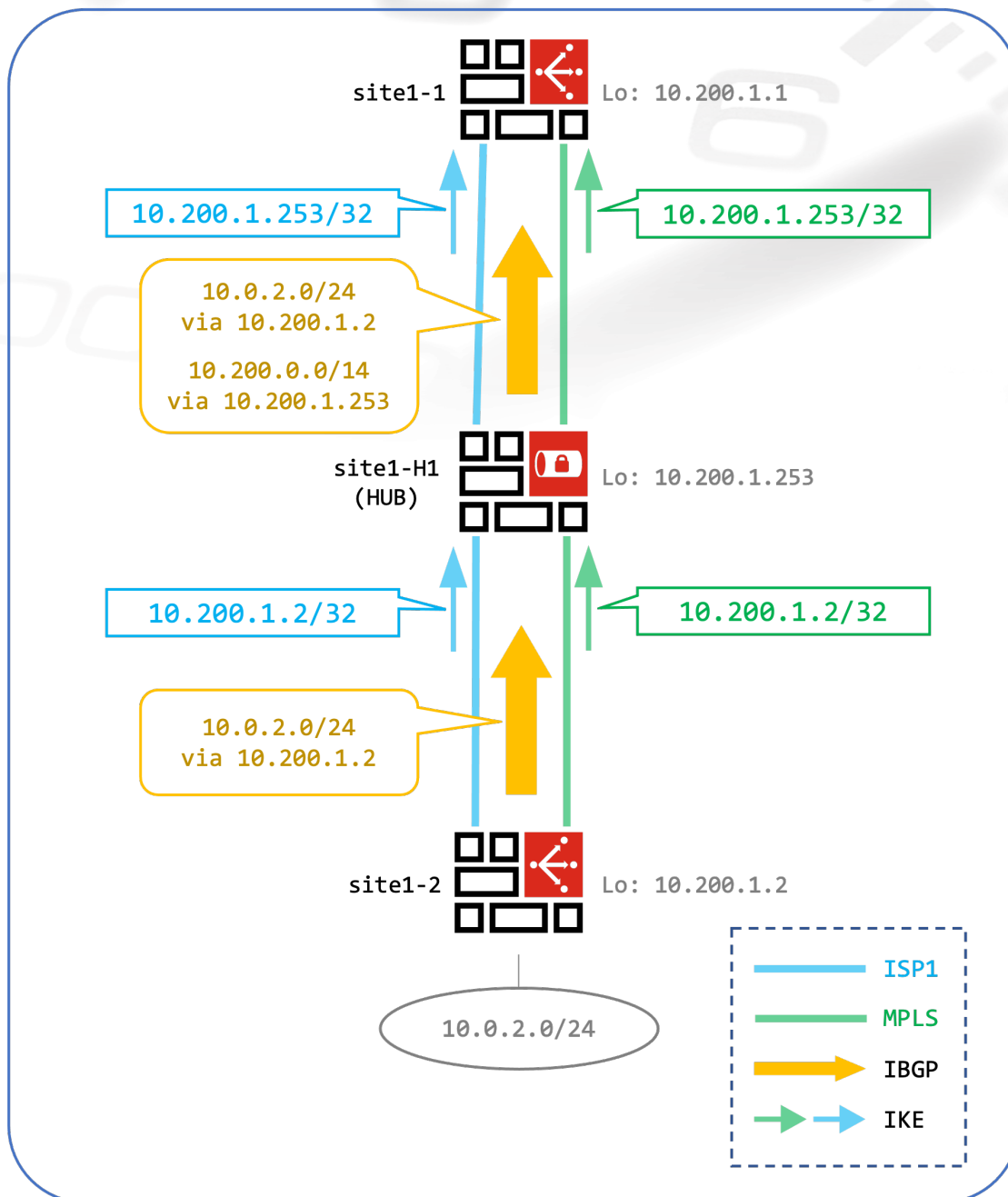
The complete FortiOS CLI configuration for this design can be found in [Dynamic BGP on loopback \(dual-hub region\)](#) on page 72.

ADVPN support with BGP RR

Another method to support ADVPN is to rely on the BGP Route Reflection (RR) and on the recursive BGP NH resolution via the shortcut.

This is the standard method used in the previous releases, and therefore it is still widely deployed.

The following diagram illustrates how the LAN prefix 10.0.2.0/24 is propagated to another Spoke "site1-1" in this case:



- The Hubs act as BGP Route Reflectors (RRs), reflecting the Spoke routes to all other Spokes in the region. This provides reachability for the user traffic across the SD-WAN overlay network.
- Besides reflecting the routes, the Hubs also advertise a **loopback summary** to the Spokes. It summarizes all the loopback addresses in the overlay network (10.200.0.0/14 in our example). BGP NH of this summary route is the loopback of the Hub itself. This provides the loopback reachability across the SD-WAN overlay network.
- On the receiving Spokes, the BGP NH of the LAN prefix is recursively resolved via the loopback summary, which in turn is resolved via a static /32 route towards the Hub loopback, injected by the already familiar IKE feature (`exchange-ip-addrv4`).

Here is the route resolution example from the "site1-1" point of view:

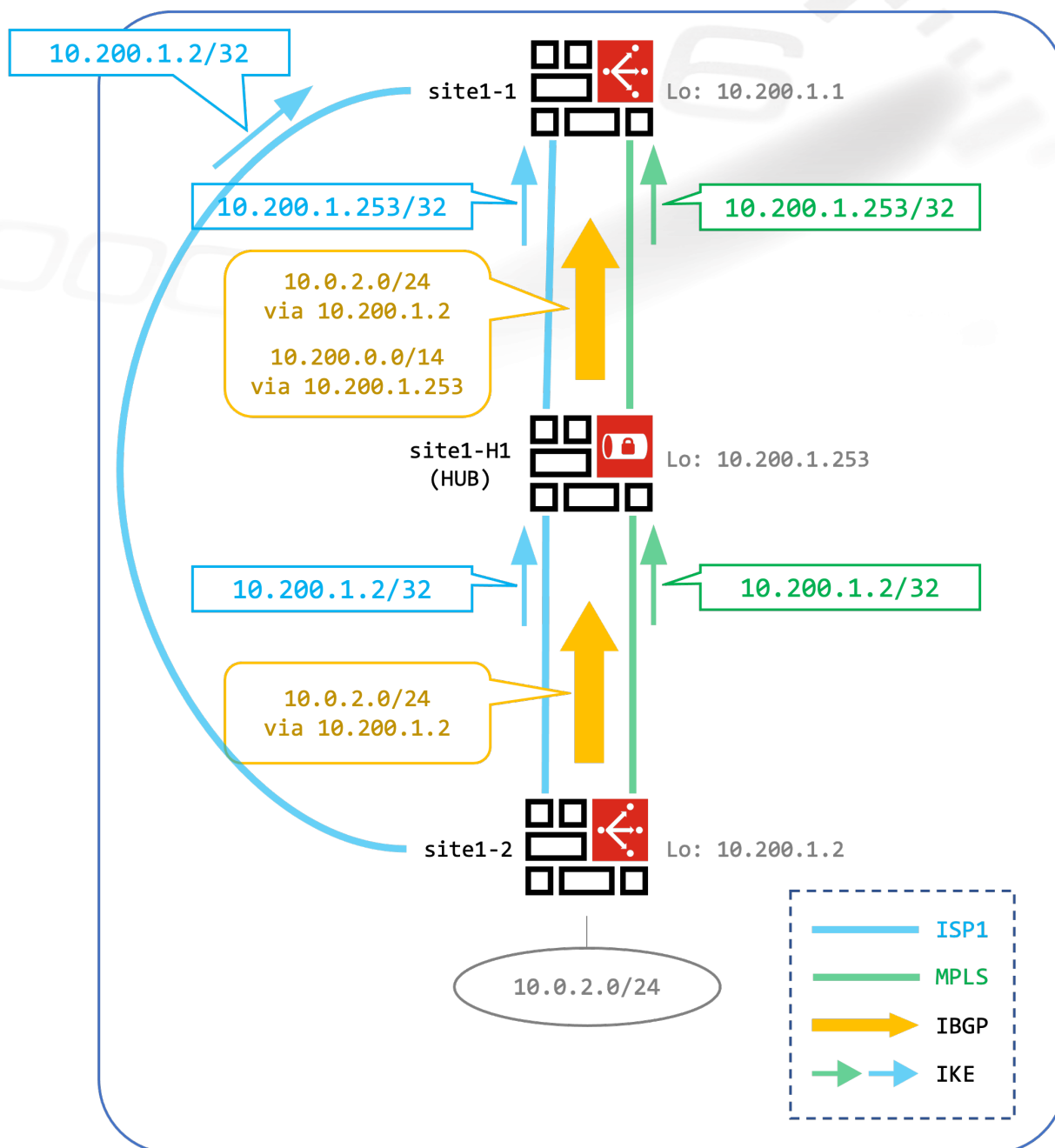
B 10.0.2.0/24 via 10.200.1.2

=> B 10.200.0.0/14 via 10.200.1.253

=> S 10.200.1.253/32 via **ISP1**

=> S 10.200.1.253/32 via **MPLS**

The next diagram shows the same topology with an active ADVPN shortcut between "site1-1" and "site1-2":



- As we have already seen, when the ADVPN shortcut is established, a /32 route towards the loopback IP of the remote Spoke is injected by IKE (the `exchange-ip-addrv4` feature).
- This allows the receiving Spoke to recursively resolve the previously learned BGP routes via the shortcut. Here is the route resolution example for this case (note that it is identical to the one achieved by the *Dynamic BGP* in the [ADVPN support with dynamic BGP \(RR-less\) on page 44](#) section):
B 10.0.2.0/24 via 10.200.1.2
=> 10.200.1.2/32 via **ISP1_0**
- When multiple active shortcuts exist between the same pair of Spokes, the same BGP routes will be recursively resolved via all the shortcuts, since the /32 loopback routes are injected via all of them. The route resolution example is again identical to the one achieved by the *Dynamic BGP* in the [ADVPN support with dynamic BGP \(RR-less\) on page 44](#) section.

- Finally, when all the active shortcuts are torn down, the Spokes return to their initial state, in which the same BGP routes are recursively resolved via the Hubs.

Multi-regional deployment

While both ADVPN support methods described above can be extended to support inter-regional ADVPN, in any multi-regional deployment it is highly recommended to implement the RR-less *Dynamic BGP* method. It is precisely in the inter-regional routing where the benefits of this design become the most evident. And for this reason, in this section we focus only on this design.

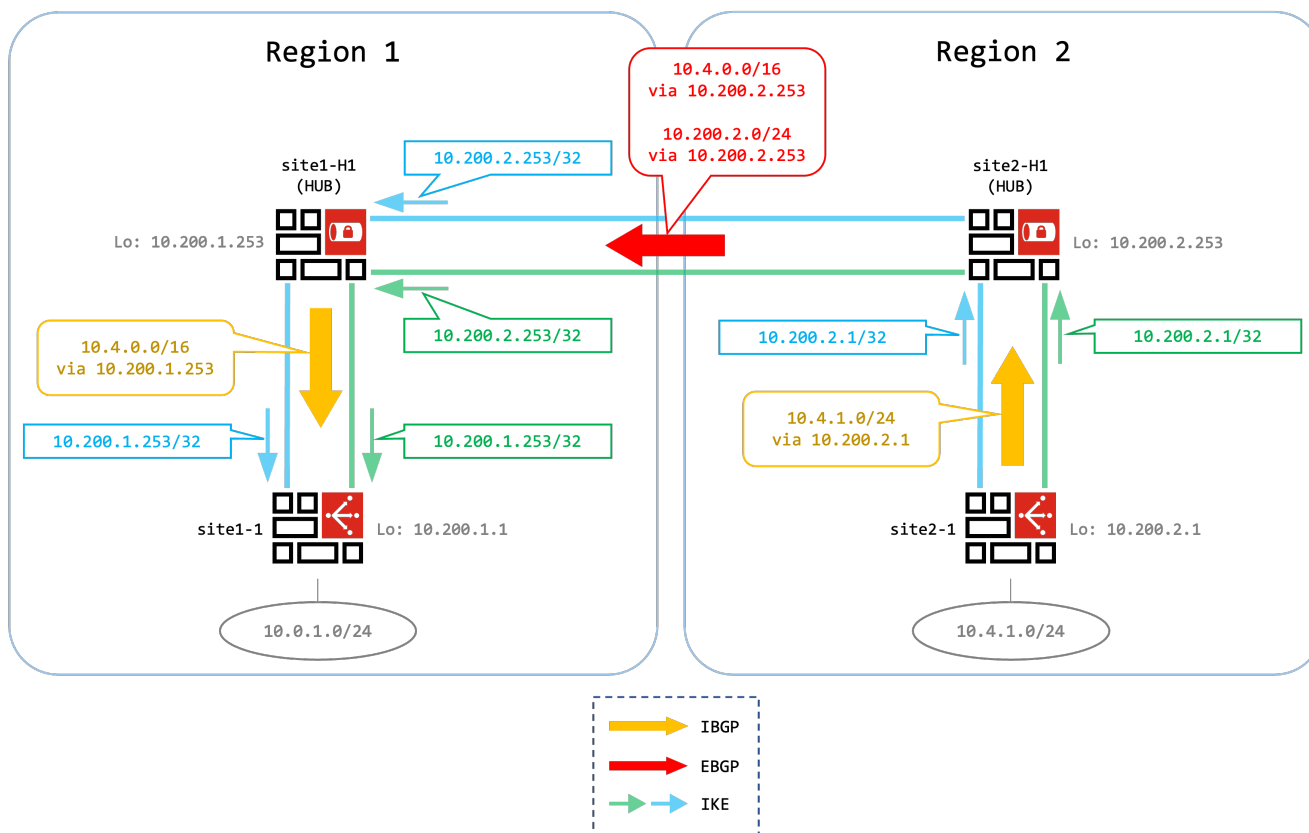
In a multi-regional deployment, every pair of Hubs serving different regions establishes a single EBG session between them. This EBG session is terminated on the same loopback used for the Spokes' IBGP sessions. The Hubs rely on the same IKE extension described earlier (`exchange-ip-addrv4` feature) to guarantee mutual loopback reachability over the Hub-to-Hub IPsec tunnels.

The following prefixes are advertised between the Hubs:

- The **regional LAN summaries** to provide reachability for the user traffic between the regions
- The **regional loopback summaries** to provide the loopback reachability between the regions
- Additionally, each Hub may also advertise any specific prefixes located behind it.

One of the benefits of the Dynamic BGP design is that the Hub-to-Hub routing configuration does not depend on whether inter-regional ADVPN support is required or not. The routing can always be optimized, by summarizing the regional LAN prefixes on the Hubs. The above description is therefore valid for both cases.

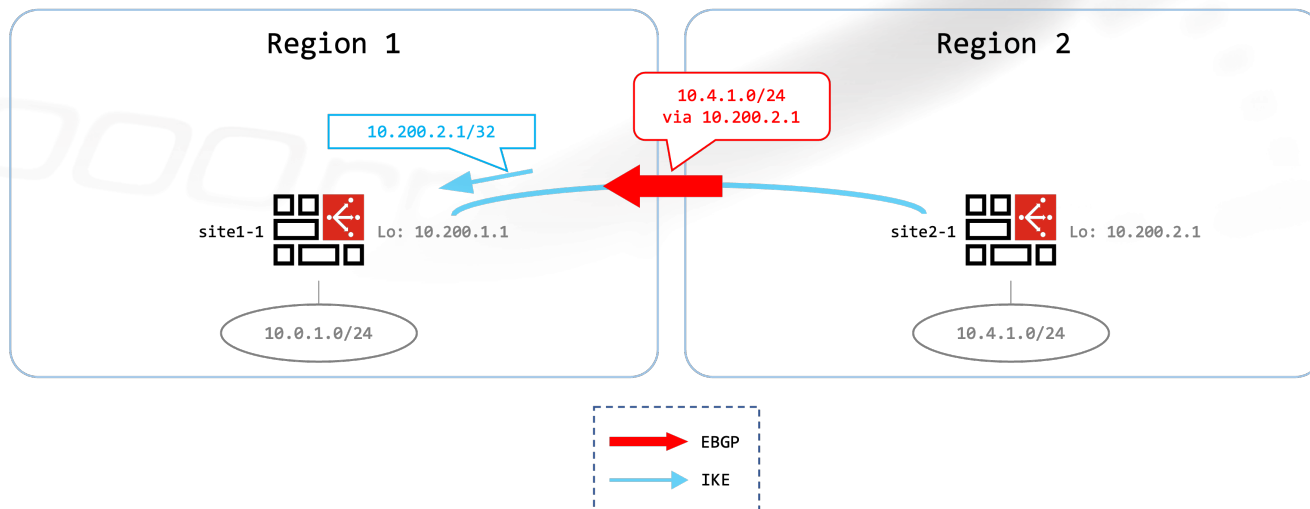
The following diagram illustrates inter-regional routing between Region 1 and Region 2:



- The Hub in Region 2 ("site2-H1") advertises a regional LAN summary (10.4.0.0/16) towards the Hub in Region 1 ("site1-H1"), which in turn readvertises it towards its Spokes. As a result, the Spoke "site1-1" (from Region 1) will be able to reach the LAN network 10.4.1.0/24 behind the Spoke "site2-1" (from Region 2).

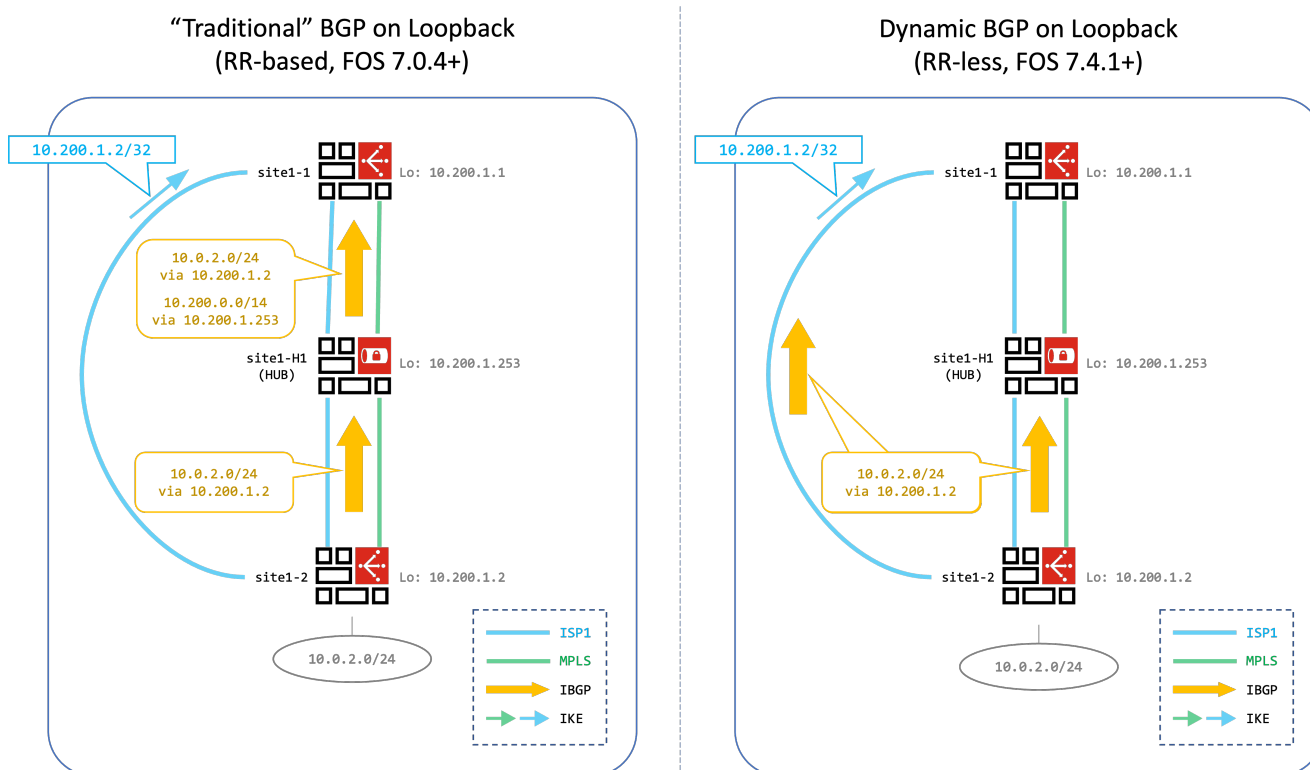
- The Hub in Region 2 also advertises a regional loopback summary (10.200.2.0/24) to guarantee loopback reachability (necessary, for example, for the ADVPN 2.0 operation).

If the inter-regional ADVPN is enabled between these two regions, the ADVPN 2.0 will be able to build a shortcut between "site1-1" and "site2-1". The two Spokes will then establish a direct BGP session over the shortcut, and "site2-1" will advertise its LAN prefix (10.4.1.0/24) to "site1-1", as demonstrated on the following diagram:



Choosing ADVPN support method

The difference between the two presented ADVPN support methods is summarized in the following diagram:



- Both methods rely on the `exchange-ip-addrv4` feature to inject the /32 loopback route through the shortcut.
- In the RR-based design, the Hubs reflect the LAN prefixes between Spokes, and the Spokes then use the injected /32 loopback route to recursively resolve the BGP NH of those reflected routes.

- In the RR-less design (with Dynamic BGP), the Spokes use the injected /32 loopback route to establish a BGP session between them over the shortcut, which is then used to exchange the LAN prefixes directly.

As a rule of thumb, **Dynamic BGP is recommended for most of the new deployments** for the following reasons:

- Eliminating the need for BGP Route Reflection greatly improves the scalability of the solution, which removes the scaling bottleneck from the Hubs.
- It also greatly simplifies the routing design, allowing route summarization on the Hubs, both within the region and—even more crucially—between the regions.

Nevertheless, there are several use cases in which the *traditional* RR-based design will be necessary or indeed preferred:

- [Existing deployments on page 50](#)
- [Preserving route attributes on page 50](#)
- [Inability to summarize LAN subnets on page 50](#)

Existing deployments

The most common use case for using the *traditional* RR-based overlay design is simply an existing SD-WAN/ADVPN deployment.

The RR-based *BGP on Loopback* design is supported starting from FortiOS 7.0.4. It provides a good degree of scalability and flexibility. Therefore, there is no urgency to change it, even after upgrading the SD-WAN nodes to FortiOS 7.4.

Furthermore, the new Dynamic BGP design is backward compatible with the RR-based design. This means that the migration to the new RR-less design can (and should) be done gradually: new SD-WAN nodes can be deployed with the Dynamic BGP method, while still being able to communicate with the existing SD-WAN nodes running the RR-based design.

We explore this backward compatibility in more detail in the [Mixed deployment on page 51](#) section.

Preserving route attributes

Another reason for using the RR-based design is to be able to advertise a certain property of the (group of) routes across the SD-WAN overlay network. This is typically done by attaching **BGP communities** to the routes in question.

Summarizing these routes on the Hubs would result in a loss of the attached BGP communities. Hence, relying on BGP Route Reflection is preferred in this situation.

We explore one practical example in the [SD-WAN strategy and dynamic routing on page 68](#) section.

It should be noted, however, that this use case does not require abandoning the Dynamic BGP entirely. As we have already mentioned (and as will be elaborated in the [Mixed deployment on page 51](#) section), the Dynamic BGP design is backward compatible with the RR-based design. It is possible to limit the scope of the BGP RR only to those routes that require preserving the BGP communities (or other attributes), while using the RR-less method for the rest of the SD-WAN nodes.

Inability to summarize LAN subnets

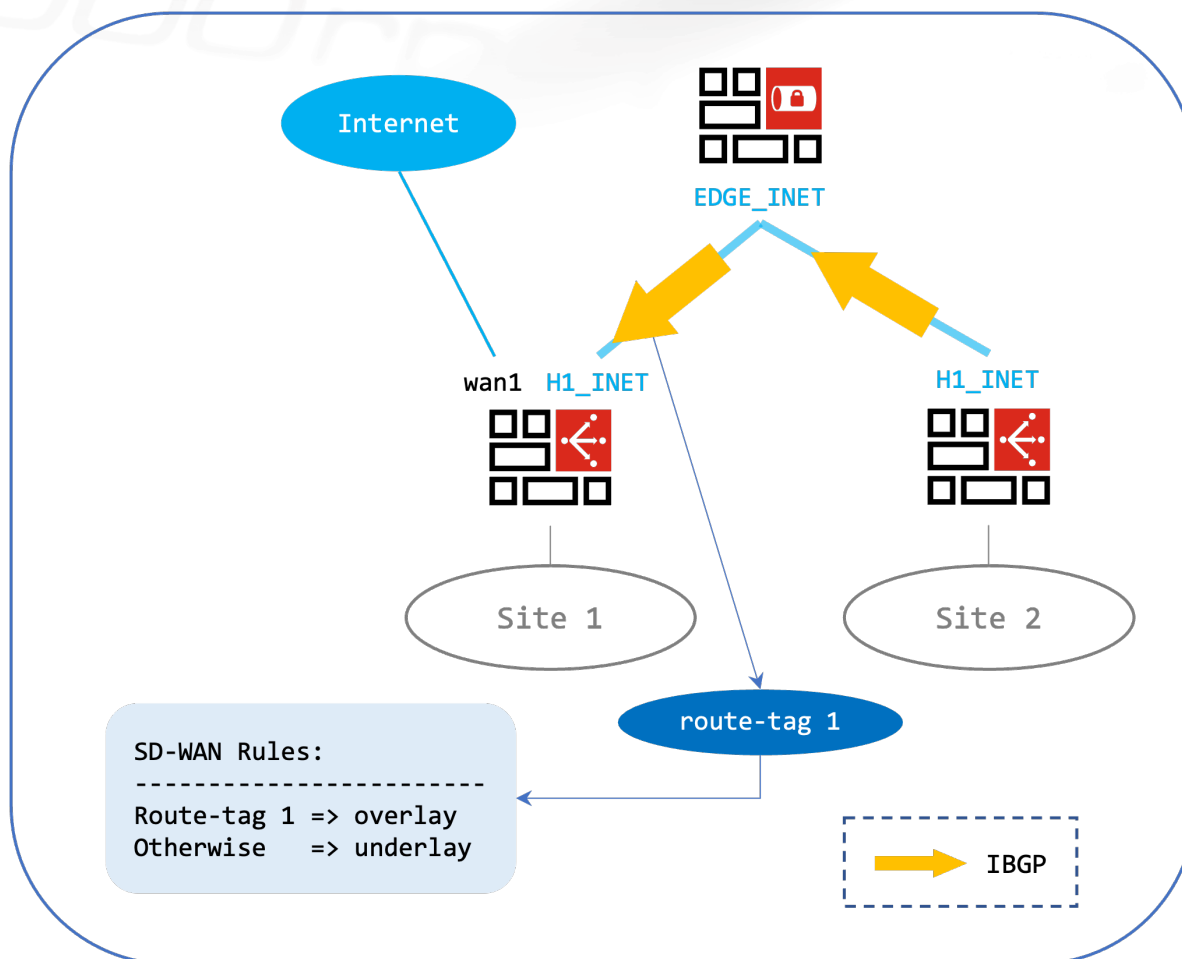
Usually it is important for an SD-WAN node to be able to distinguish between the corporate traffic and the Internet traffic. They are often subject to a different steering policy and/or a security policy.

For example, from a Spoke perspective, all the corporate traffic must be steered using the overlay tunnels (towards the other SD-WAN nodes), while the Internet traffic is typically steered using the underlay (DIA).

Most often the corporate LAN subnets used by SD-WAN sites can be easily summarized, for example, by using a private IP range (such as 10.0.0.0/8). This gives you an easy way to achieve the above goal: you can either use this summary range directly in the SD-WAN rules or simply rely on the routing.

However, it is not always possible to summarize the LAN subnets used by the SD-WAN sites. For example, public IP ranges might be used by some of them, making it difficult (or impossible) to distinguish the traffic destined to these sites from the Internet traffic. In this case, how will a Spoke be able to guarantee that the corporate traffic is steered using the overlay tunnels, while the Internet traffic is steered using the underlay?

One solution is similar to the previous use case: the Hubs can reflect the specific LAN subnets advertised by the remote sites and the Spokes will use this information in the SD-WAN rules. The following diagram illustrates this approach:



Also here it is important to remember that the scope of the BGP RR can be limited only to those sites that require it (which may not necessarily be the entire SD-WAN network). The rest of the SD-WAN nodes can still rely on the RR-less method.

Mixed deployment

Backward compatibility between the new RR-less "Dynamic BGP" design and the more traditional RR-based design can be implemented in two ways:

1. You can limit the scope of the Route Reflection by adding a custom route-map on the Hubs. This method is especially useful when you cannot summarize the LAN prefixes of *some* of your SD-WAN sites, while you can still summarize the others. The Spokes will then receive the LAN summaries, as well as *some* of the specific LAN prefixes.

Note that with this method you should still keep "Dynamic BGP" enabled on all the Spokes, to guarantee direct communication between the Spokes whose LAN prefixes are summarized and those whose LAN prefixes are not.

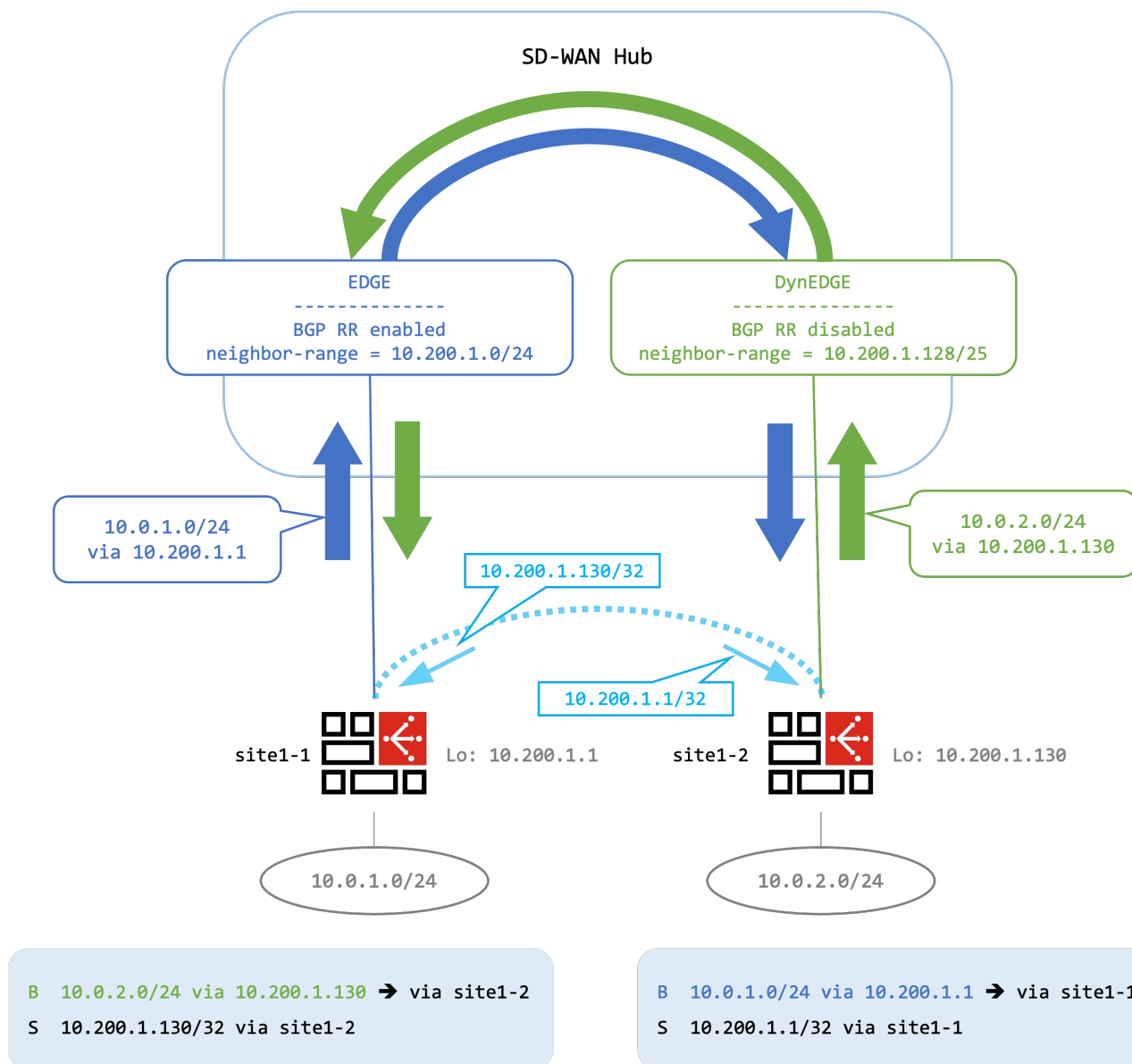
- Another method is more comprehensive, because it allows having "Dynamic BGP" disabled on some of the Spokes, while still guaranteeing their direct communication with the Spokes using "Dynamic BGP". This method is, therefore, a good fit for the gradual migration (when some of the Spokes are still running older FOS releases or simply older configuration).

Let us take a closer look at the second method. It relies on the standard behavior of the IBGP protocol: **IBGP automatically readvertises prefixes between the RR-clients and non-clients.**

We can therefore split the Dial-Up BGP configuration on the Hubs into two different neighbor-groups:

- One neighbor-group will be used for the Spokes with the *Dynamic BGP* feature configured. The BGP RR will be disabled on this group.
- Another neighbor-group will be used for the Spokes without the *Dynamic BGP* feature. The BGP RR will be enabled on this group.

The following diagram illustrates the route exchange between two Spokes using the two different ADVPN support methods:



- Both Spokes advertise their LAN prefixes to the Hub using IBGP.
- The LAN prefix 10.0.2.0/24 is reflected to "site1-1" (non-client to RR-client), and the LAN prefix 10.0.1.0/24 is reflected to "site1-2" (RR-client to non-client).
- When an ADVPN shortcut comes up, no dynamic BGP session is established, because Dynamic BGP is not configured on "site1-1".
- However, both sides recursively resolve the BGP NH of the remote LAN prefix through the shortcut, thanks to the /32 loopback route injected by IKE (the `exchange-ip-addrv4` feature).

In other words, both Spokes use the traditional RR-based mechanism to recursively resolve the routes' next-hops through the shortcuts.

At the same time, "site1-2" can use Dynamic BGP to communicate to other Spokes that have Dynamic BGP enabled. Similarly, "site1-1" can use the traditional mechanism to communicate to other Spokes that do not have Dynamic BGP enabled.

In summary, any Spoke-to-Spoke connectivity is possible in this mixed deployment.

Planning guidelines

1. Our design relies heavily on the **loopback**, which uniquely identifies each SD-WAN node within the overlay network. It is used for BGP termination, ADVPN 2.0 operation, ADVPN shortcut monitoring and more.

- During the design stage, we must allocate a single subnet summarizing all the loopbacks (the **loopback summary**).
- Additionally, in multi-regional deployments with Inter-Regional ADVPN support, the loopbacks must be split between the regions in such a way that they can be summarized also per-region (the **regional loopback summary**). As you have seen previously, this summary will be advertised between the Hubs.
- Finally, in a mixed deployment, each regional loopback summary must be further split between the Spokes using the RR-based design and those using the RR-less *Dynamic BGP* design. This is needed to simplify the configuration of the BGP peering on the Hubs.

For example, the loopback summary might be 10.200.0.0/14, which is further split into multiple regional loopback summaries 10.200.1.0/24, 10.200.2.0/24, and so on. Then, assuming that the existing (RR-based) sites are using the lower loopback IPs, we can allocate the 10.200.1.128/25 range to the new sites that will use the RR-less design.

Note that the loopbacks discussed here have significance only within the SD-WAN overlay network. They will never be advertised outside of the overlay network and will not interfere with the rest of the environment.

The size of the loopback subnet is determined by the size of the SD-WAN network: it must accommodate all the SD-WAN nodes (Hubs and Spokes). Be careful when splitting it into regional loopback subnets! Each regional subnet must be large enough to accommodate all the SD-WAN nodes in that region!

2. It is also highly recommended to plan the **LAN subnets**, especially in a multi-regional deployment. With the Dynamic BGP design, summarization is always allowed on the Hubs. If the regional LAN networks can be summarized, this will greatly improve the overall scaling of the SD-WAN overlay network.

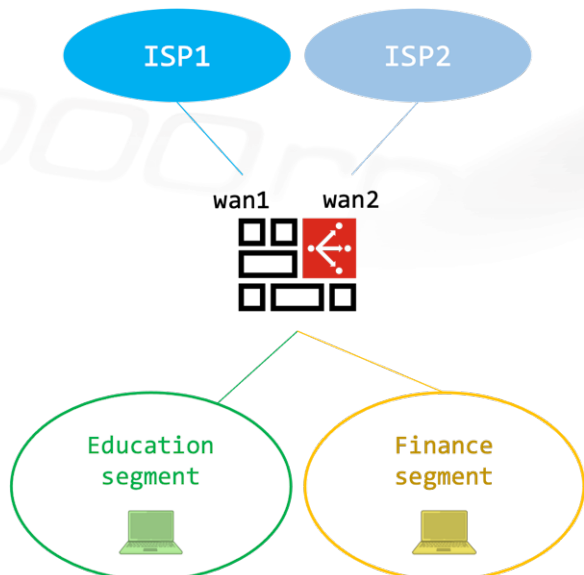
Unfortunately, it is understood that in many real-world networks this will not be possible. Unlike the loopback addressing (which resides only within the SD-WAN overlay network), the LAN addressing can be directly dictated by the network environment beyond our control.

Segmentation over single overlay

We now turn to the topic of **segmentation** across the SD-WAN network. The use case can be summarized as follows:

- Customer sites have multiple *segments* segregated from each other. Within each site, each segment may be connected to separate network equipment (such as separate switches and routers), but segments may also share the same equipment that supports multiple **VRFs**.

For example, the Customer may want to segregate data and OAM traffic in the network. Another typical requirement is to segregate data traffic of different sensitivity (classification levels), such as Education and Finance, Guest and Corporate, and so on.



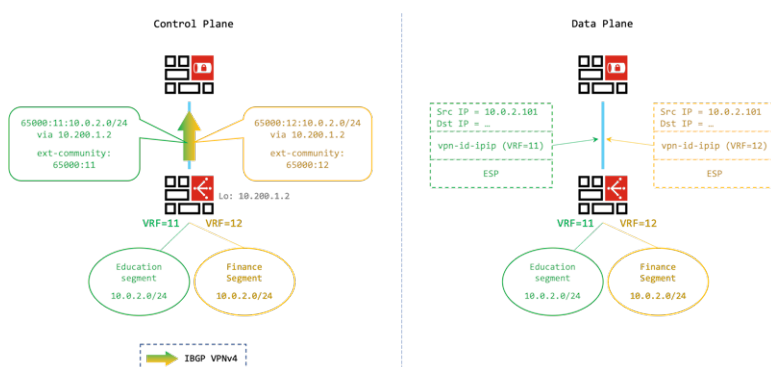
- Not only traffic flow, but also route propagation is permitted only *within* a segment. In other words, segmentation means also **control plane segregation**. It also means, for example, that IP addressing overlap is possible between different segments!



For this reason, segmentation cannot be implemented only using firewall policies! While they can provide granular segregation of traffic flows, they do not restrict route propagation and do not allow IP overlap. (In other words, they do not implement control plane segregation.)

- The requirement is to *preserve* this segregation across the entire SD-WAN network, when we interconnect the sites. In other words, the different segments should not *mix* with each other. Their segment identity must be preserved end-to-end. IP addressing overlap must be supported and so on.

How is this implemented in our SD-WAN Solution?



- FortiGate devices support multiple VRFs. Therefore, on each SD-WAN node, different Customer segments will be attached to different VRFs.



Different VRFs effectively mean different routing tables.

- On the control-plane, we use `MP-BGP VPNv4` to advertise VRF information together with each LAN prefix over the entire SD-WAN overlay network.
- On the data-plane, the traffic is *tagged* using a new `vpn-id-ipip` encapsulation, when it is forwarded between the SD-WAN nodes. This encapsulation is enabled on all the IPsec overlay tunnels (including ADVPN shortcuts). In other words, we send the traffic from multiple segments (VRFs) over the same IPsec tunnel, while *tagging* it with its VRF information, so that the receiving SD-WAN node can preserve the segregation. This is why we sometimes call this feature *VRF-Aware Overlays*.



The `vpn-id-ipip` encapsulation is a variation of IPIP. An extra IP header is added inside the ESP payload (encrypted), encoding the VRF ID into the IP address field.

Note that all our current ASIC and SoC generations (NP6, NP7, SoC4, SoC5, and so on) support acceleration of IPIP. This was one of our main considerations when designing the new encapsulation.

- [Routing design on page 55](#)
- [CE and PE VRFs on page 56](#)
- [Planning guidelines on page 56](#)
- [Providing Internet access on page 57](#)

Routing design

There is no change in the way the SD-WAN nodes establish BGP sessions between them. The difference is only in the *content* of the BGP advertisements: we use a different BGP address family (VPNv4), and every advertised BGP route now includes the following important elements:

- The prefixes are prepended with a **route distinguisher (RD)**, to make them look unique.

Remember that IP overlap is permitted between different segments (VRFs). On the diagram in the [Segmentation over single overlay on page 53](#) topic, we see how the two VRFs both contain the same LAN prefix 10.0.2.0/24. By prepending two unique RDs, "10.200.1.2:**11**" and "10.200.1.2:**12**" respectively, we make these prefixes look unique: "10.200.1.2:**11**:10.0.2.0/24" and "10.200.1.2:**12**:10.0.2.0/24". Even though both BGP routes are originated by the same Spoke (and hence having the same BGP NH), these are nevertheless two different BGP routes now.



By including the loopback IP (10.200.1.2), we make the RD unique not only per VRF, but also per SD-WAN node. For example, the RD for the same VRF=11 on another SD-WAN node, with the loopback IP 10.200.1.254, would be "10.200.1.254:11". This property is recommended to avoid unwanted BGP behavior in some use cases involving multi-homing (when the same LAN are advertised by multiple SD-WAN nodes).

- An extended community called **route target (RT)** is attached to each route, and it signals from which VRF this route is exported (advertised) and into which VRF the receiving node must import it.



The RT must be unique only per VRF. But for each given VRF, it must remain identical across the entire SD-WAN network. That is why we recommend using the Autonomous System number (ASN). Note that in a multi-regional environment it should be a "fictitious" ASN, so that it remains identical in all regions.

The combination of RD and RT allows us to advertise Customer LAN prefixes, while preserving their VRF information across the entire SD-WAN overlay network.



The exact values of RT/RD don't matter, as long as they meet the requirements stated above. These values will never be readvertised to any peers outside of the SD-WAN overlay network. Therefore, they will not conflict with any existing MP-BGP deployments (such as existing BGP/MPLS L3VPNs in Customer network).

CE and PE VRFs

To complete the picture, we need to introduce one more piece of terminology. We define the following types of VRFs on the SD-WAN nodes:

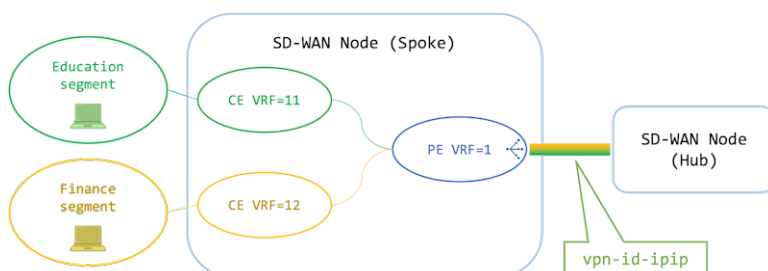
- **CE VRF** is a Customer VRF to which the actual LAN segment is attached. Each LAN-facing interface is assigned to one of the CE VRFs.
- **PE VRF** is an Edge VRF in which the VRF-Aware Overlays (with `vpn-id-ipip` encapsulation) are located. The tunnel interfaces are assigned to the PE VRF.

Thus, in a typical multi-VRF deployment, we will have one PE VRF and multiple CE VRFs.



The maximum number of VRFs supported by FOS 7.6.1+ is 512 per VDOM (from 0 to 511). We put VRF=0 aside. (It is a special case, as will be pointed out below.) In a typical multi-VRF deployment this means 1 PE VRF + up to 510 CE VRFs.

The following diagram summarizes the above:



Now when we have explained the design, let us list some of the important [Planning guidelines on page 56](#).

Planning guidelines

The most important question is how to assign VRF IDs to the Customer segments. Let us start by listing all the constraints that we have:

- **Avoid using VRF=0 for PE/CE.** Due to its special role in FortiOS architecture, there are good reasons not to use VRF=0 in the multi-VRF SD-WAN overlay network. One such reason is explained in the next section, where we discuss the Internet access.
To be clear: it is perfectly fine to use VRF=0 for other local purposes, without making it part of the SD-WAN overlay network. (Think of an out-of-band management, for example.) These use cases are valid.
- **Use the same VRF IDs everywhere.** The VRF IDs have *global significance* in our solution. All the SD-WAN nodes (Hubs and Spokes!) must use the same CE/PE VRF IDs!
This means that a particular segment (for example, Finance) must be attached to the same CE VRF across the entire SD-WAN network.
This also means that the PE VRF must be the same across the entire SD-WAN network.

- **Use PE VRF for Internet access (and for all WAN underlays).** In theory, we could talk about a separate WAN VRF to which the WAN-facing underlay interfaces are assigned. Apart from terminating the IPsec overlays, it would have another very important duty of providing Internet access to the CE VRFs. But we recommend using the PE VRF for this purpose, unless you have a very good reason to do it otherwise.

In other words, our recommendation is that the PE VRF includes all the WAN-facing interfaces (underlays and overlays). Simply put: PE VRF = WAN VRF = Internet VRF.



See also [Providing Internet access on page 57](#), where we discuss Internet access in multi-VRF deployments in more detail.

With these constraints in mind, we recommend the following VRF allocation:

VRF ID	Role
0	Preferably not part of the SD-WAN overlay network (for example, OOB management)
1	PE VRF
2-511	CE VRFs

Finally, let us repeat our recommendations regarding the assignment of the RD/RT values to the CE VRFs:

- The RD must be unique per VRF and per SD-WAN node. We therefore recommend using the VRF ID and the loopback IP:
RD = <loopback>:<vrf_id>
- The RT must be unique only per VRF. That is, for each given VRF, the RT must be identical on all the SD-WAN nodes. For this reason, we recommend selecting a single Autonomous System number (such as 65000) and using it for all the CE VRFs on all the SD-WAN nodes.
RT = 65000:<vrf_id>.
- Note that if you extend the solution to multiple regions, the VPNv4 routes will be exchanged also between the regions (including their RDs and RTs). Recall that each region has its own AS number. However, the RT must be configured consistently on all the SD-WAN nodes across the entire network!

Hence, when selecting the AS number for your RT, it is recommended NOT to use your region's AS! Do not confuse the AS number used for BGP operation with the "fictitious" AS number used for the RT value, it does not have to (and should not) be the same. For example, you can use the value 65000 for the RTs (as shown above), while your regions will use AS numbers 65001, 65002, and so on for BGP operation.

Providing Internet access

As we have explained previously, Customer segments are attached to the CE VRFs, while all the WAN-facing interfaces (both underlays and overlays) are typically attached to the PE VRF. This raises an important question: how can we provide Internet access to the Customer network?

While MP-BGP control plane together with the `vpn-id-ipip` data plane facilitate corporate (site-to-site) connectivity, there is simply no route (and no outgoing interface!) in any of the CE VRFs to reach Internet destinations!

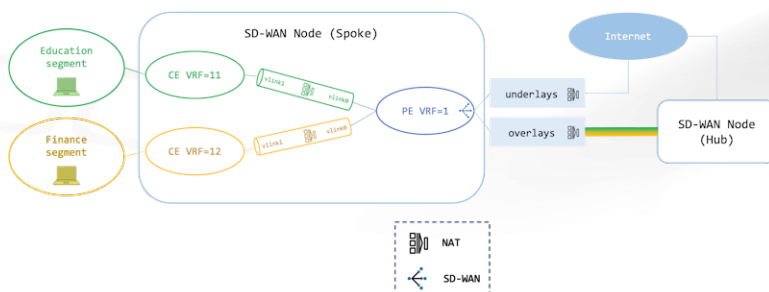
Since only the PE VRF is physically connected to the Internet, the traffic from the CE VRFs will have to **"jump" to the PE VRF**, in order to get Internet access. In other words, we will need to connect the CE VRFs to the PE VRF. On FOS, we can do that in the same way as we interconnect different VDOMs: using *VDOM links*. We will call them **VRF links** in this section.



It is important to highlight that it is NOT required to configure any additional VDOMs! We are only going to reuse the same internal links to interconnect the VRFs.

On the hardware appliances we will prefer using the NPU links (which are accelerated), while on the VMs we will use the software-based `vdom-link`.

The following diagram illustrates this design:



Notes:

- With this design, each flow is processed twice by the FortiGate, creating **two independent sessions**: one from the LAN to the VRF link and another from (the other side of) the VRF link to the WAN. The first session belongs to the CE VRF and the second session to the PE VRF.
- This design is used for both Direct and Remote Internet Access: once the traffic reaches the PE VRF, it can egress either through an underlay (DIA) or through an overlay (RIA). Note that in the RIA case the traffic reaches the Hub already in the PE VRF context (not mixing up with the internal CE VRF traffic on the Hub).
- Notice the SD-WAN icon on the diagram in the PE VRF. This is where the SD-WAN rules are applied. We can select the best breakout option in the same way we do it in the *usual* (single-VRF) deployments. The VRF links do not need to be configured as SD-WAN Members. They become *ingress* interfaces from the SD-WAN perspective. Only the session in the PE VRF is subject to the SD-WAN processing.
- It is not the same for the Firewall Policies, because each of the two sessions must be independently permitted by the Firewall rules. Therefore, we will need to create separate rules for the CE VRF session and for the PE VRF session.
- Finally, we need to address the routing carefully:
 - CE VRFs can simply have a static default route towards the VRF link, directing all the Internet traffic towards the PE VRF.
 - It is trickier in the opposite direction, because the replies arriving from the Internet to the PE VRF must find their way back to the right CE VRF. We could copy the LAN prefixes from the CE VRFs to the PE VRF, but this would hardly be a good idea: the whole point of the VRFs is to keep the LAN prefixes segregated inside their respective segments!

A better approach is to **NAT** the traffic on the VRF links. This way, the PE VRF doesn't need to know the LAN prefixes.



Source NAT on the VRF link is supported only when the target VRF is not 0. This is another reason why you should not use VRF=0 as your PE VRF!

- Finally, there will typically be the *usual* Source NAT on the egress from the PE VRF (that is, before the traffic leaves the Edge device). In other words, we are going to NAT twice!



The complete FortiOS CLI configuration for this design can be found in [Segmentation over single overlay on page 80](#).

Additional topics

This section covers important use cases widely seen in practice:

- [Topologies with multiple Internet links on page 59](#)
- [Hub-to-Edge SD-WAN on page 62](#)
- [SD-WAN strategy and dynamic routing on page 68](#)

Topologies with multiple Internet links

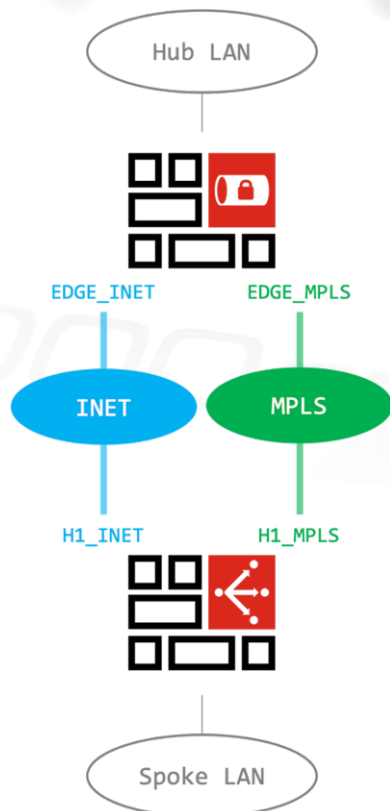
We have already mentioned the topic of **segregated versus interconnected transports** in the [Transport groups on page 38](#) section in the context of ADVPN shortcut creation. Here we will look into another aspect of this topic, that of the Spoke-to-Hub connectivity.

Let us look again at the *classical* case of the segregated transports, namely the SD-WAN topology in which every SD-WAN site has two WAN links: an Internet connection and an MPLS connection. When we build an SD-WAN overlay network in this case, we can make the following two observations:

- A Spoke cannot build an IPsec tunnel from its Internet WAN link to the Hub's MPLS WAN link or the reverse.
- A Spoke cannot build an ADVPN shortcut from its Internet WAN link to another Spoke's MPLS WAN link or the reverse.

Consequently:

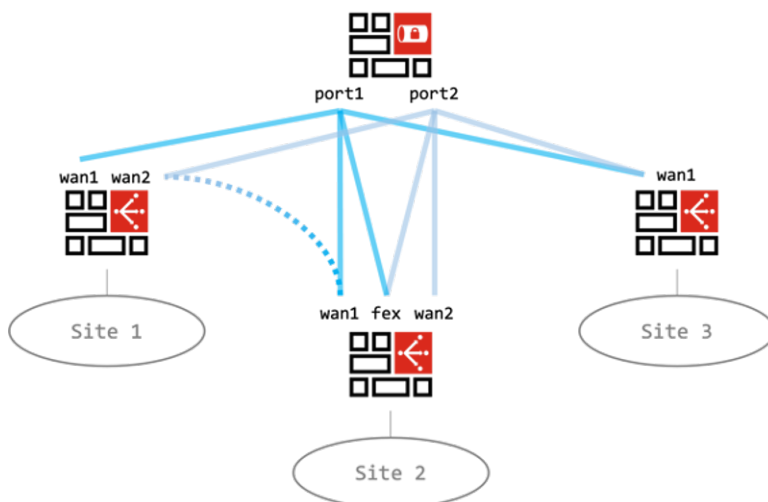
- The only option for the overlay network in this case is to have a *one-to-one mapping* between the underlay and the overlay. That is, we would build an Internet overlay and an MPLS overlay:



- Cross-overlay communication is only possible through the Hub.
- Cross-overlay ADVPN shortcuts are not possible.

As time passed, however, different sorts of topologies emerged—those with multiple Internet connections. An SD-WAN site can have multiple Internet links, sometimes of different types (for example, one fixed line and one mobile connection). An MPLS connection can still be present, but a growing number of projects consists of Internet-only sites. As we said before, all the Internet connections are, in fact, interconnected transports.

It can be easily seen that the above-listed constraints of the segregated transports no longer apply to the interconnected transports. The following diagram illustrates several examples:



Notes:

- A Spoke can build IPsec tunnels to any of the Hub's WAN links. It can even build multiple IPsec tunnels from the same local Internet connection towards multiple Hub's endpoints. For example, Site 3 on the above diagram has just a single Internet connection, but nevertheless builds two Internet overlays.

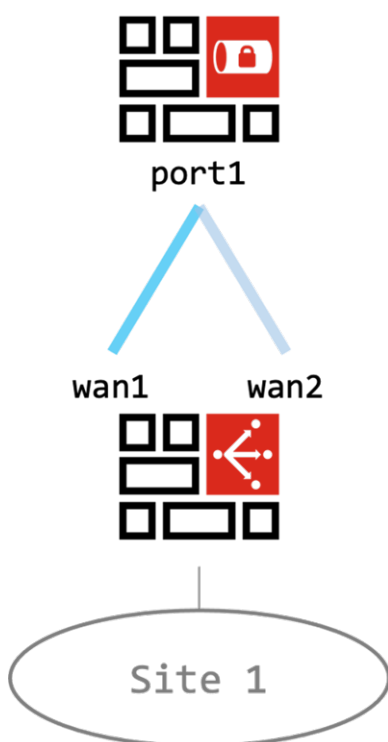
- Typically, SD-WAN sites (Spokes and Hubs) do not have the same number of WAN links. Different SD-WAN sites may have different number of Internet connections, as can be clearly seen on our diagram.
- Spokes can build ADVPN shortcuts using any pair of Internet links (including cross-overlay).

It is important to keep in mind that not all the physically possible options should be recommended for use. For simplicity reasons, it may still be beneficial in some projects to keep the one-to-one mapping between the underlays and the overlays, just as it was used in the classic SD-WAN topologies (Internet + MPLS).

However, the newer [BGP on loopback on page 42](#) design allows us to configure several new variants without increasing complexity (and sometimes even simplifying the configuration!). Let's discuss some of these options:

- [Hub connection: many-to-one on page 61](#)
- [Hub connection: full-mesh on page 62](#)

Hub connection: many-to-one

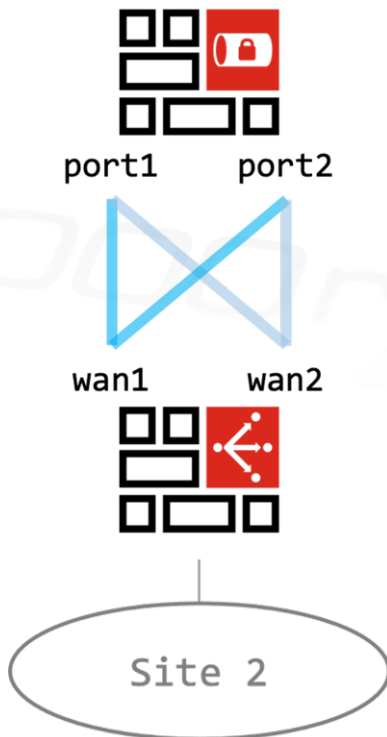


In this example, Site 1 has two Internet connections, while the Hub has only a single one. With the [BGP on loopback on page 42](#) design, we can configure a single Dial-Up IPsec endpoint on the Hub to which the Spoke will establish two static IPsec tunnels.

Remember that, unlike with the legacy *BGP per overlay design*, we no longer need to configure tunnel subnets, and this is why the same Spoke can establish multiple IPsec tunnels towards the same Dial-Up endpoint without tunnel subnet conflicts. We also do not need to configure additional BGP sessions, since there is always a single session between the Spoke and the Hub.

The IKE `exchange-ip-addrv4` feature will inject the /32 loopback route through both of the established tunnels, allowing the BGP routes to be recursively resolved through all available overlay paths.

Hub connection: full-mesh



In this example, we enhance the Spoke-to-Hub connection redundancy, by interconnecting the two nodes with a full mesh of static IPsec tunnels.

Do not confuse this with a Full-Mesh overlay network topology! Our overlay network still remains Hub-and-Spoke, as described earlier. We are only talking about the overlay connectivity between a given Spoke and a Hub.

Also in this example, with [BGP on loopback on page 42](#) design, we can configure a single Dial-Up endpoint terminated on each of the Internet connections on the Hub (two Dial-Up endpoints in total). Each Spoke will establish two static IPsec tunnels towards each of these endpoints (four tunnels in total). There will be no tunnel subnets and no additional BGP sessions. BGP routes will be recursively resolved through all available overlay paths.

It must be noted that this topology nevertheless introduces certain added configuration complexity on the Spoke side, because all the static IPsec tunnels must become SD-WAN Members. In our example, the Spoke will have four overlay SD-WAN Members that must be correctly ordered in the SD-WAN rules.

Hub-to-Edge SD-WAN

The term **Hub-to-Edge SD-WAN** refers to the problem of intelligent steering of the traffic originated behind the Hub(s). We often refer to this intelligence as **Remote Health Signaling**, due to the nature of our solution to this problem, as will become clear shortly.

Looking deeper, we can distinguish between two separate aspects of this problem. We will look into each aspect separately.

- [Hub-to-Spoke sessions on page 63](#)
- [Advertising preferred Hub on page 64](#)

Hub-to-Spoke sessions

As you have learned from the reference designs described so far, the Spokes are responsible for the sessions originated behind them. They constantly measure the health of all available paths and apply the configured SD-WAN rules to all the outgoing traffic, deciding where to steer it at any given moment.

This technique cannot be applied to the Hub-to-Spoke (Hub-to-Edge) sessions, that is, the sessions originated behind the Hub. While, in theory, the Hub could measure the health of all the overlay paths towards all the Spokes, this design would be suboptimal for at least two reasons:

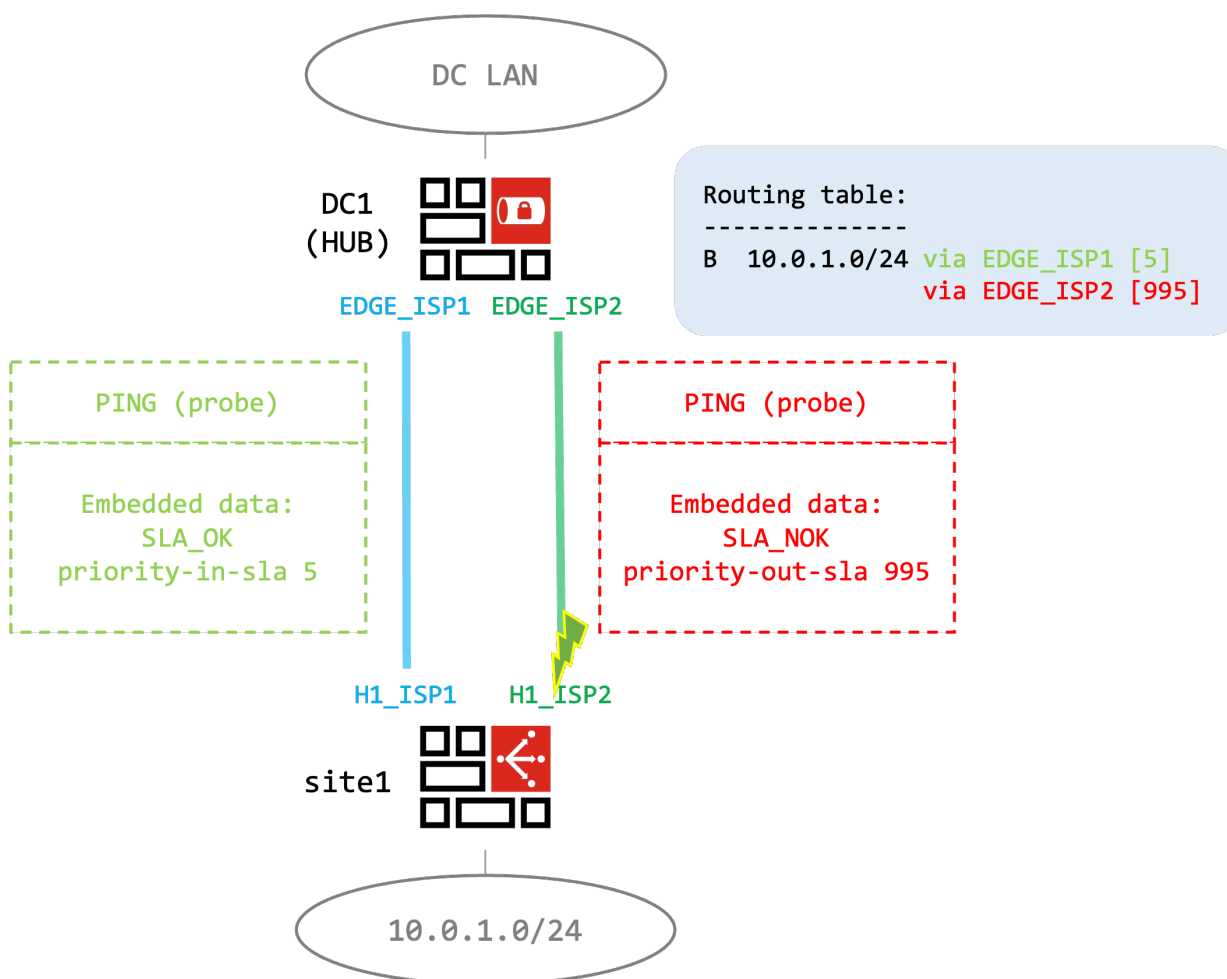
1. It is a clear scaling bottleneck, because a single Hub should be able to serve hundreds and thousands of Spokes
2. It is a duplicate effort, because all the Spokes already measure the health of the overlay paths towards their serving Hubs by actively sending PING probes over all the static overlay tunnels.



It is important to highlight that we are NOT talking about the *replies* to the sessions originated by the Spokes! This problem does not exist for the replies, because they automatically follow a symmetrical path. That is, once the originating Spoke selects a particular overlay for the Spoke-to-Hub session, the reply packets will automatically prefer the same overlay on their way back from the Hub.

In this section, however, we are specifically addressing the sessions **originated** behind the Hubs!

Our latest solution to this problem, available starting from release 7.6, is depicted on the following diagram:



1. For each SD-WAN overlay member, a pair of priority values can be configured, known as **priority-in-sla** and **priority-out-sla**. They define the desired priority of each given overlay member when it meets its SLA target and when it does not.
2. Each Spoke performs the usual health measurement of each overlay member (latency, jitter and packet loss), applies its configured SLA target and determines which overlay members meet the SLA and which do not. Based on this, it determines the **current priority** of each overlay member (applying either the priority-in-sla or the priority-out-sla value to it, depending on its current health status).
3. Each Spoke will then **embed** the current priority of each overlay member into the probe packets themselves. When these probes arrive at the Hubs, the Hubs will "read" the priorities, *without* actively probing the Spokes.
4. Finally, the Hubs will apply these priorities directly to the routes (LAN prefixes) learnt from the Spokes. This way, for example, the routes via *healthy* overlays will get better priorities than those via *unhealthy* overlays.

Note that, since the priority-in-sla / priority-out-sla pairs are configured independently on the Spokes, as well as the SLA targets, it means that different Spokes can "signal" different behavior to the Hubs, depending on their local requirements, usually dictated by the network topology of each Spoke.

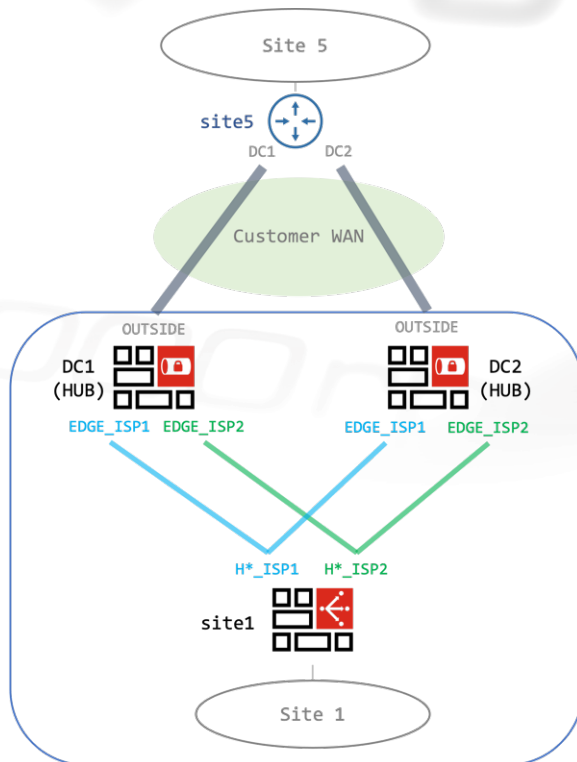
Consider a few examples:

- A Spoke with multiple Internet connections without any particular preference between them (e.g. two equally priced broadband connections) could set the same priority-in-sla = 5 and priority-out-sla = 995 for all of them. As a result, any healthy overlay would have the same priority = 5, while any unhealthy overlay would have the priority = 995. The lower priority is better, hence this logic would simply allow the Hubs to skip unhealthy overlays for the Hub-to-Spoke sessions. Note that when all the overlays are healthy, they will all have the same priority, and the path will be selected for each session in ECMP manner. In the other words, the Hubs will load-balance across all the healthy overlays towards the Spoke in question.
- At the same time, another Spoke might have a primary broadband Internet connection and an additional 4G/5G connection that must only be used as a backup, when the primary connection is out of SLA. In this case, the Spoke could set the priority-in-sla value to 5 for the broadband overlay and to 6 for the 4G/5G overlay. Then, even when both overlays meet the SLA target, the broadband will be preferred for the Hub-to-Spoke sessions, because its current priority (5) will be better than that of the 4G/5G overlay (6).
- Note that also the SLA target is determined on each Spoke. Hence, it can be different on different Spokes. For example, if one group of Spokes is connected to the Internet using fast FTTH connections, while another group of Spokes is using much slower DSL lines, it is very likely that these two groups of Spokes will apply different SLA targets to determine what overlays are considered healthy.

In summary, each Spoke will **signal** its individual preferences to the Hubs, and the Hubs will use the conventional routing to steer the Hub-to-Spoke sessions accordingly. Generally, no SD-WAN rules will be required on the Hubs.

Advertising preferred Hub

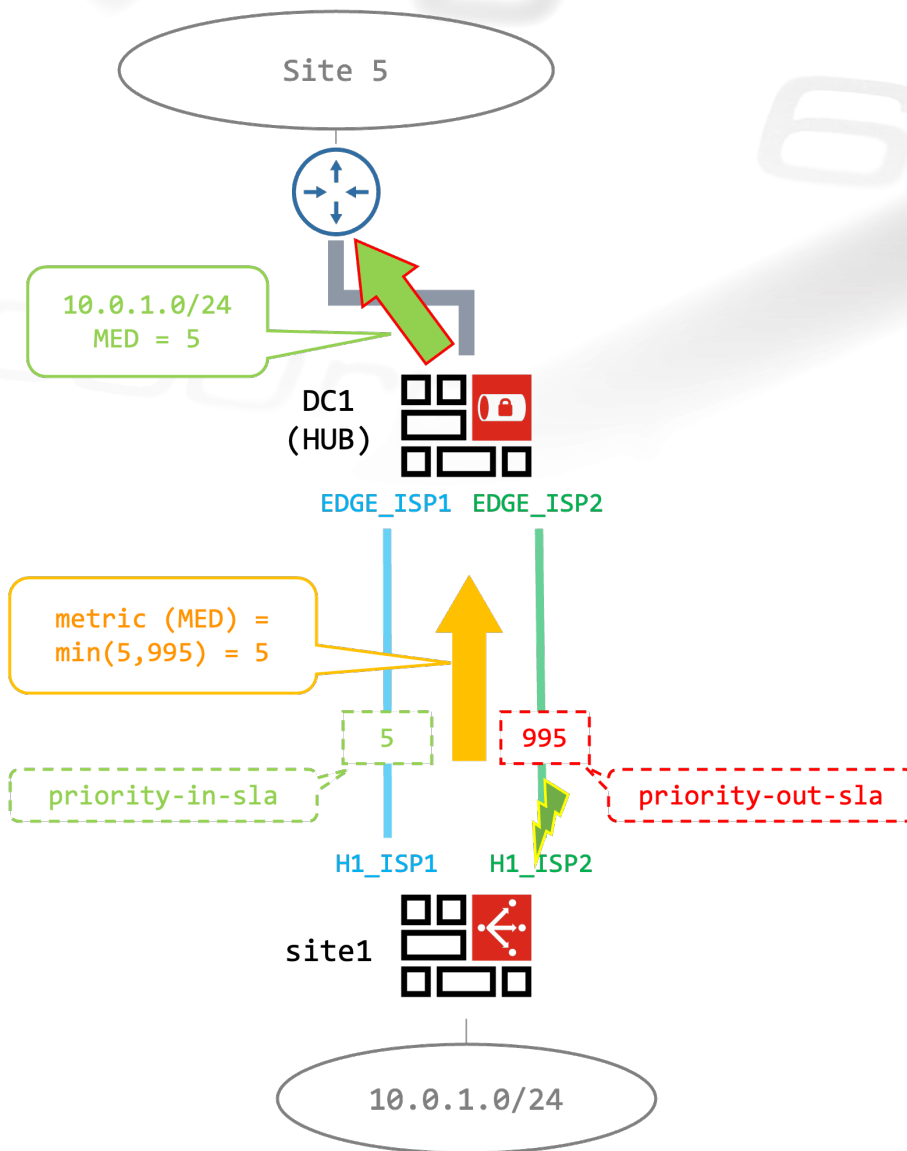
The Hub-to-Edge SD-WAN problem can be extended beyond the SD-WAN network itself. Take a look at the following diagram:



We can see a standard Dual-Hub SD-WAN/ADVPN region. But let's focus on the traffic **from Site 5 towards Site 1**. The originating site (Site 5) is not part of the SD-WAN overlay network and it is served by a third-party device. Our goal is to **signal** to this site our currently preferred **entry point** into the SD-WAN network.

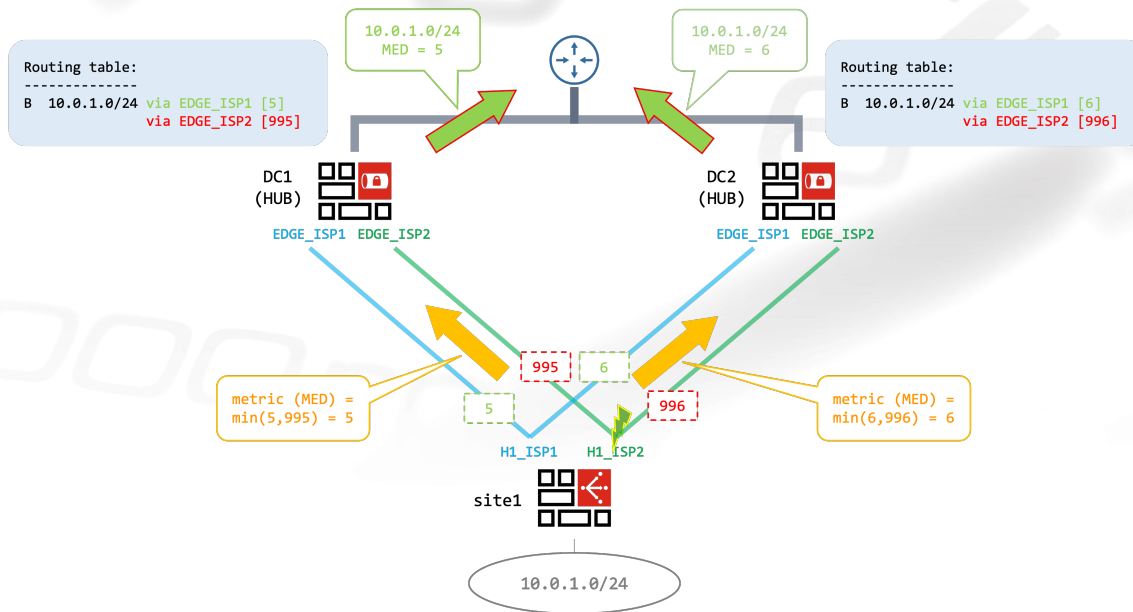
In other words, should it use the DC1 or the DC2 for the sessions towards the SD-WAN Site 1?

Our solution to this problem is to let the SD-WAN health measurements influence the advertisement of the LAN prefixes (the depicted solution is available starting from release 7.6):

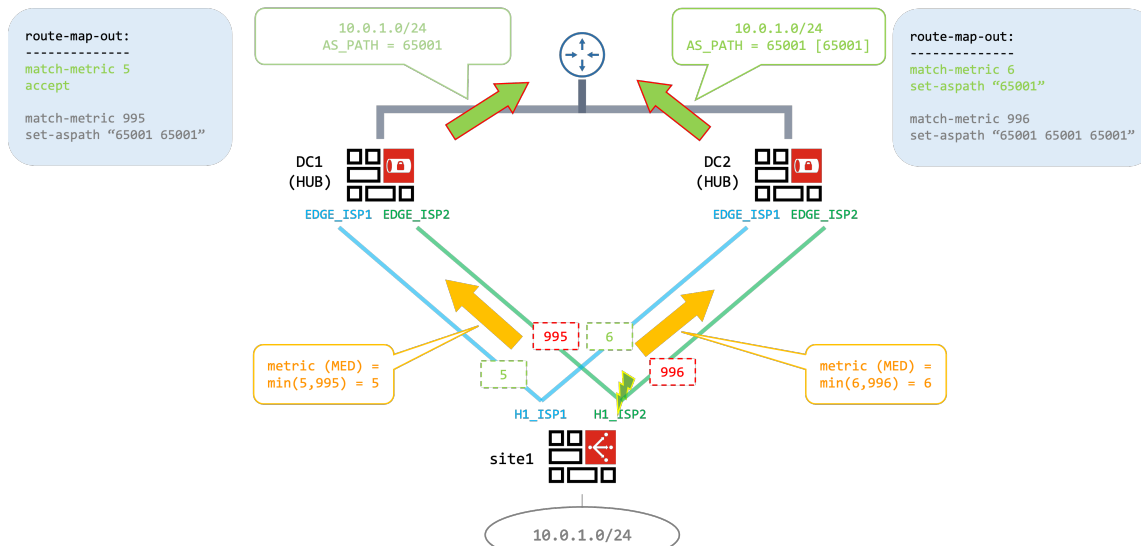


1. After determining the *current priority* of each overlay member (as explained in the previous section), every Spoke will calculate a *minimum* of all the current priorities towards each Hub, and this value will become the **current metric** of that respective Hub.
2. This metric will be applied to all the routes (LAN prefixes) advertised by the Spoke towards that respective Hub, in a form of a standard BGP MED attribute.
3. The Hubs will readvertise this metric outside of the SD-WAN network, thus "signaling" the currently preferred entry point.
Optionally, the Hubs can also *translate* this metric into another BGP attribute (e.g. prepending AS_PATH or attaching a BGP Community).

The following diagram illustrates this solution with two Hubs - note how the different MED values signal our currently preferred entry point:



Similarly, when prepending AS_PATH instead of using the MED attribute:



As can be seen, also this part of the solution is based on the calculations performed individually by the Spokes, directly influenced by the configuration of the priority-in-sla / priority-out-sla pairs. The Hubs merely pass this information outside of the SD-WAN network boundaries.

Summary

In both parts of the presented solution, the control remains with each individual Spoke. Just as the Spoke determines (using the SD-WAN rules) how to steer the traffic originated behind it, it also determines how the traffic must be steered towards it, when it originates behind the Hubs or outside of the SD-WAN network.

As we have shown, the Spoke uses two mechanisms to signal its preferences:

- Per-overlay priorities are embedded into the Spoke-to-Hub probes, to signal preferences to the Hubs
- Per-Hub metric is advertised as a standard BGP attribute, to signal preferences outside of the SDWAN network

This solution enables a high degree of flexibility, allowing different groups of SD-WAN nodes to signal different desired behavior. Nevertheless, it must be noted, that the flexibility of the Hub-to-Edge SDWAN is still inferior compared to the

sessions originated behind the Spokes: while the former allows only per-overlay granularity, the latter enjoys the full granularity provided by the SD-WAN rules, including per-Application steering, User-based steering, several supported steering strategies and so on.



The complete FOS CLI configuration for the Hub-to-Edge SD-WAN can be found in [Hub-to-edge SD-WAN on page 86](#).

SD-WAN strategy and dynamic routing

In our SD-WAN Solution, traffic steering decisions are taken primarily by the originating sites (that is, by the SD-WAN node behind which a session in question has originated). But there are use cases when it is convenient to base the traffic steering decision on the information received from the dynamic routing (that is, on the information advertised by *other* sites, either SD-WAN sites or not!).

Route-Tags provide a useful mechanism for this integration. The idea is simple:

- Route-tags are assigned locally on the SD-WAN node to the incoming routes, based on a user-defined criteria. For example, all the routes having a certain BGP community can be *marked* with a certain route-tag.
- This route-tag, in turn, can be used as a *matching criteria* in an SD-WAN rule.
- This way, we can use dynamic routing to *signal* certain site properties to the SD-WAN nodes, that will use this information to choose the best suitable traffic steering strategy.

Let us stress again that this signaling can come either from within the SD-WAN network (from another SD-WAN node) or from outside. We have already seen in the [Hub-to-Edge SD-WAN on page 62](#) section how we can advertise *our* preferences outside of the SD-WAN network. Here we show the signaling in the opposite direction:

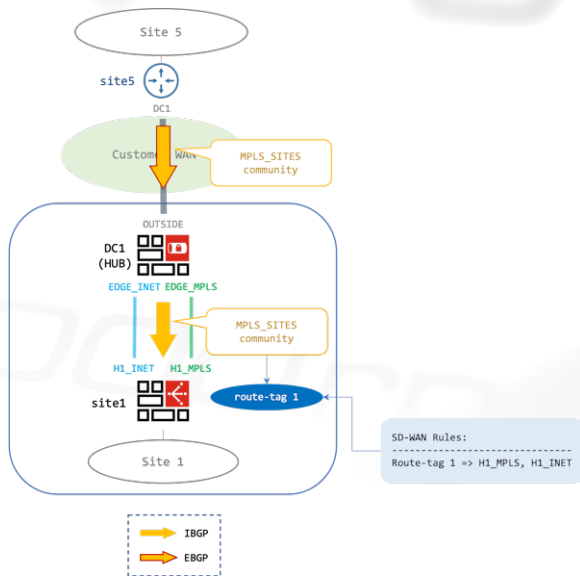
- [Practical example on page 68](#)
- [Deep-dive: route-tags in SD-WAN rules on page 69](#)

Practical example

Let us consider the practical example of preferring the MPLS overlay when sending traffic towards a certain group of sites.

- Instead of statically configuring these destinations on all the SD-WAN nodes, we can attach a BGP community to all the routes originated by this group of sites.
- This BGP community will be dynamically translated into a route-tag on all the receiving SD-WAN nodes.
- Finally an SD-WAN rule will match this route-tag and apply a strategy preferring the MPLS overlay for this traffic.

The following diagram illustrates this approach:



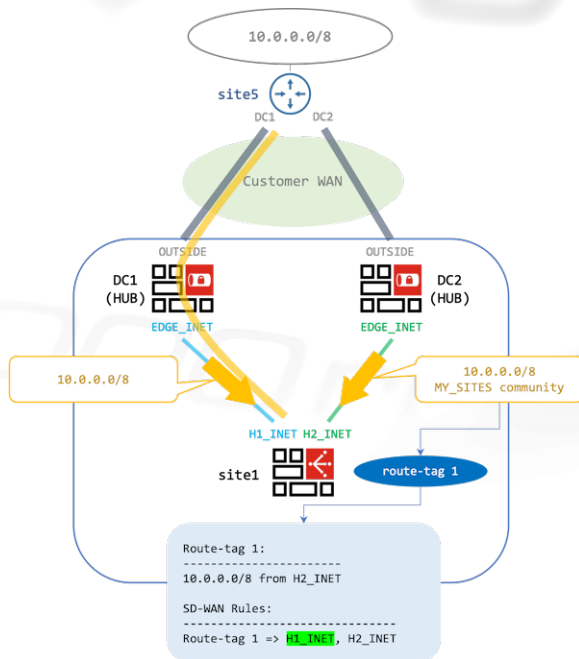
Whenever a new destination must be added to this group, we can attach the same BGP community to it on the originating site. The route can be originated either by another SD-WAN node or by a third-party device outside of the SD-WAN network. The latter is possible, because the method relies on the industry-standard BGP functionality (namely, BGP communities). Note that the route-tags themselves have only local significance and are never advertised outside of the SD-WAN nodes.

Deep-dive: route-tags in SD-WAN rules

On release 7.4 of FortiOS, the following two rules apply:

1. Once the route-tag is applied, it becomes a property of the *destination*, not of a particular *path* towards this destination. For instance, it does not matter from which BGP session the community has been learned. The `route-tag` will simply apply to the subnet (prefix).
2. As a result, an SD-WAN rule will match a specified `route-tag` (expressed as a new type of an address object), even if the SD-WAN Member from which the `route-tag` has been learned is not listed in this rule.
3. Furthermore, even if the SD-WAN Member from which the `route-tag` has been learned and listed in the rule, the rule can still select another SD-WAN Member, from which the `route-tag` has not been learned.

This is best understood with examples:

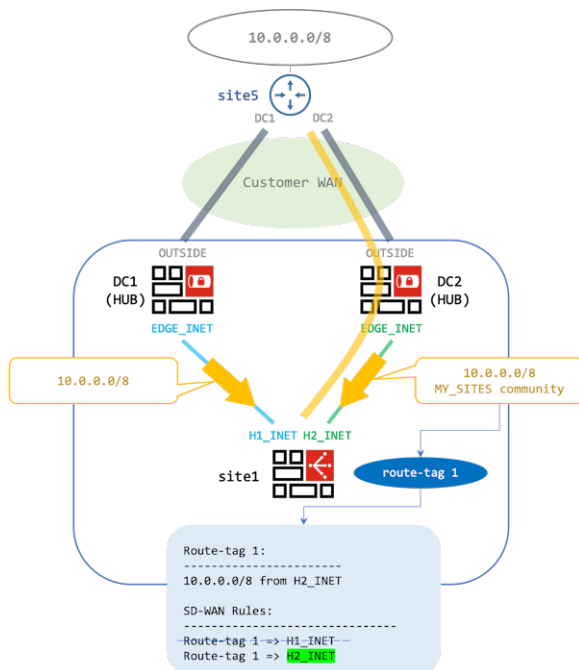


- Here the node "site1-1" has an SD-WAN rule that matches on `route-tag 1` and applies the *Lowest Cost (SLA)* strategy, with a primary overlay `H1_INET` and a secondary overlay `H2_INET`.
- The Hub H2 is advertising a route towards `10.0.0.0/8` with a BGP community, which is translated into the `route-tag 1`.
- The Hub H1 is also advertising a route towards `10.0.0.0/8`, but without any BGP community.

Assuming that both overlays are healthy, what will happen to the traffic destined to `10.0.0.0/8`?

- The destination `10.0.0.0/8` is *marked* with a `route-tag 1`. Hence the rule will be successfully matched.
- The rule strategy instructs it to prefer `H1_INET`. Even though the `route-tag` is not learned from the BGP session running over `H1_INET`, nevertheless the traffic will be steered to `H1_INET`!

Now let us configure two separate SD-WAN rules with a *Manual* strategy:



What will happen to the traffic destined to `10.0.0.0/8` in this case?

- The destination 10.0.0.0/8 is *marked* with a `route-tag 1`. Even though this `route-tag` is not learned from the BGP session running over H1_INET, the first rule will nevertheless be matched, and the traffic will be steered to H1_INET!
- The second rule will never be matched, as long as the H1_INET member is alive.



This behavior differs from the earlier releases! In the Release 7.2, the first rule would be skipped.

Reference configurations

In this appendix, we will demonstrate step-by-step how to configure some of the discussed designs using the FOS CLI. These instructions can serve as a reference for the final result that should be achieved when following a deployment workflow of your choice:

- [Dynamic BGP on loopback \(dual-hub region\) on page 72](#)
- [ADVPN 2.0 on page 78](#)
- [Segmentation over single overlay on page 80](#)
- [Hub-to-edge SD-WAN on page 86](#)

Please consult our [SD-WAN Deployment for MSSPs](#) guide to learn about the recommended deployment workflow.

Dynamic BGP on loopback (dual-hub region)

This reference configuration topic contains the following sections:

- [Spoke on page 72](#)
- [Hub on page 76](#)
- [Optimizing failover on page 78](#)

Spoke

1. Configure Loopback to be used for BGP termination and for ADVPN shortcut monitoring:

```
config system interface
  edit "Lo"
    set vdom "root"
    set type loopback
    set ip 10.200.1.1/32
    set allowaccess ping
  next
end
```

2. Configure a unique Location ID (best practice: use Loopback!)

```
config system settings
  set location-id 10.200.1.1
end
```

3. Configure IKE to automatically inject the static route to reach the Loopback on all the phase1-interfaces towards the Hub:

```
config vpn ipsec phase1-interface
  edit "H1_ISP1"
    set exchange-ip-addr4 10.200.1.1
  next
  edit "H1_MPLS"
    set exchange-ip-addr4 10.200.1.1
  next
```

```

edit "H2_ISP1"
    set exchange-ip-addr4 10.200.1.1
next
edit "H2_MPLS"
    set exchange-ip-addr4 10.200.1.1
next
end

```

4. There is no need to configure any tunnel IPs (no IPs on the interfaces H1_ISP1, H1_MPLS, H2_ISP1 and H2_MPLS)

5. Configure BGP:

- Single neighbor per Hub (using the Hub's Loopback), terminated on the Loopback
- No ADD-PATH is needed
- Must configure `set recursive-next-hop enable`

```

config router bgp
    set as 65001
    set router-id 10.200.1.1
    set keepalive-timer 15
    set holdtime-timer 45
    set ibgp-multipath enable
    set recursive-next-hop enable
    set graceful-restart enable
    config neighbor
        edit 10.200.1.253
            set soft-reconfiguration enable
            set capability-graceful-restart enable
            set advertisement-interval 1
            set interface "Lo"
            set update-source "Lo"
            set connect-timer 1
            set remote-as 65001
        next
        edit 10.200.1.254
            set soft-reconfiguration enable
            set capability-graceful-restart enable
            set advertisement-interval 1
            set interface "Lo"
            set update-source "Lo"
            set connect-timer 1
            set remote-as 65001
        next
    end
    config network
        edit 1
            set prefix 10.0.1.0/24
        next
    end
end

```

6. Configure Dynamic BGP (if desired):

- In a single-region deployment, the following configuration is enough:
 - It enables dynamic IBGP sessions from remote Spokes within the same region
 - Must configure `set passive disable`

```

config router bgp
  config neighbor-group
    edit "DYN_EDGE"
      set soft-reconfiguration enable
      set capability-graceful-restart enable
      set advertisement-interval 1
      set next-hop-self enable
      set interface "Lo"
      set update-source "Lo"
      set passive disable
      set remote-as 65001
    next
  end
  config neighbor-range
    edit 101
      set prefix 10.200.0.0/14
      set neighbor-group "DYN_EDGE"
    next
  end
end

```

- To make the configuration ready for a multi-regional deployment, additional steps are necessary:
 - Must use `set remote-as-filter` to enable both IBGP and EBGP sessions from remote Spokes
 - Must tag the local LAN prefixes to restrict route advertisement via dynamic EBGP sessions

```

config router aspath-list
  edit "SDWAN_AS"
    config rule
      edit 1
        set action permit
        set regexp "6500."
      next
    end
  next
end

config router route-map
  edit "LAN_TAG"
    config rule
      edit 1
        set set-tag 100
      next
    end
  next
  edit "LAN_OUT"
    config rule
      edit 1
        set match-tag 100
      next
    end
  next
end

config router bgp
  config neighbor-group

```

```

    edit "DYN_EDGE"
        set ebgp-enforce-multihop enable
        set soft-reconfiguration enable
        set capability-graceful-restart enable
        set advertisement-interval 1
        set next-hop-self enable
        set interface "Lo"
        set update-source "Lo"
        set passive disable
        set remote-as-filter "SDWAN_AS"
        set route-map-out "LAN_OUT"
    next
end
config neighbor-range
    edit 101
        set prefix 10.200.0.0/14
        set neighbor-group "DYN_EDGE"
    next
end
config network
    edit 1
        set prefix 10.0.1.0/24
        set route-map "LAN_TAG"
    next
end
end

```

7. In SD-WAN Member configuration, must set `set source` on all the overlays, to ensure that the Loopback IP is used as a source of health probes:

```

config system sdwan
    config members
        edit 2
            set interface "H1_ISP1"
            set source 10.200.1.1
        next
        edit 3
            set interface "H1_MPLS"
            set source 10.200.1.1
        next
        edit 4
            set interface "H2_ISP1"
            set source 10.200.1.1
        next
        edit 5
            set interface "H2_MPLS"
            set source 10.200.1.1
        next
    end
end

```

8. In Firewall Policy configuration, must add a rule to permit incoming health probes (for the ADVPN shortcut monitoring) and Dynamic BGP sessions (if enabled), both destined to the Loopback:

```

config firewall policy
  edit 4
    set name "Peering"
    set srcintf "overlay"
    set dstintf "Lo"
    set srcaddr "all"
    set dstaddr "all"
    set action accept
    set schedule "always"
    set service "PING" "BGP"
  next
end

```

Hub

1. Configure Loopback to be used for BGP termination:

```

config system interface
  edit "Lo"
    set vdom "root"
    set type loopback
    set ip 10.200.1.253/32
    set allowaccess ping
  next
end

```



DO NOT use the same loopback as Spokes' health-check server! Using the same loopback is not supported, because the health-check routes are statically injected on the Spokes, remaining even when the respective overlay is down. As a result, the BGP session will not be able to switchover to another overlay upon failure!

2. Configure a unique Location ID (best practice: use Loopback!)

```

config system settings
  set location-id 10.200.1.253
end

```

3. Configure IKE to automatically inject the static route to reach the Loopback on all the Dial-Up phase1-interfaces towards the Spokes:

```

config vpn ipsec phase1-interface
  edit "EDGE_ISP1"
    set exchange-ip-addr4 10.200.1.253
  next
  edit "EDGE_MPLS"
    set exchange-ip-addr4 10.200.1.253
  next
end

```

4. There is no need to configure any tunnel IPs (no IPs on the interfaces EDGE_ISP1 and EDGE_MPLS)
5. Configure BGP:
 - Single neighbor-group for all Spokes, terminated on the Loopback
 - No ADD-PATH is needed
 - No RR is needed, if Dynamic BGP is enabled on the Spokes

- Must configure `set recursive-next-hop enable`
- Must advertise a **loopback summary** to all the Spokes (10.200.0.0/14 in the snippet below):

```
config router route-map
  edit "LOCAL_REGION"
    config rule
      edit 1
        set set-community "no-export"
      next
    end
  next
end
config router bgp
  set as 65001
  set router-id 10.200.1.253
  set keepalive-timer 15
  set holdtime-timer 45
  set ibgp-multipath enable
  set ebgp-multipath enable
  set recursive-next-hop enable
  set graceful-restart enable
  config neighbor-group
    edit "EDGE"
      set soft-reconfiguration enable
      set capability-graceful-restart enable
      set advertisement-interval 1
      set next-hop-self enable
      set remote-as 65001
      set interface "Lo"
      set update-source "Lo"
    next
  end
  config neighbor-range
    edit 1
      set prefix 10.200.0.0/14
      set neighbor-group "EDGE"
    next
  end
  config network
    edit 1
      set prefix 10.200.0.0/14
      set route-map "LOCAL_REGION"
    next
    edit 2
      set prefix 10.1.0.0/24
    next
  end
end
```



If Dynamic BGP is NOT used, then RR must be enabled on the neighbor-group for the correct ADVPN operation:

```
config router bgp
  config neighbor-group
    edit "EDGE"
      set route-reflector-client enable
    next
  end
end
```

6. In Firewall Policy configuration, must add a rule to permit incoming BGP sessions destined to the Loopback:

```
config firewall policy
  edit 6
    set name "Peering"
    set srcintf overlay
    set dstintf "Lo"
    set srcaddr "all"
    set dstaddr "all"
    set action accept
    set schedule "always"
    set service "PING" "BGP"
  next
end
```

Optimizing failover

The following best-practice recommendations improve failover times during different network issues:

- On all devices, DPD timers must be **shorter** than BGP hold-timer. This way, if one overlay fails, the BGP session will seamlessly switchover to another available overlay.
- On the Hub, `dpd on-idle` is recommended, with timers **shorter** than those on the Spokes. This will allow the Hub to quickly detect a failed overlay on a Spoke, to avoid resolving routes via dead overlays.
- It is recommended to configure **default route** on the Spokes towards the overlays. One easy way of doing this is using `set sdwan-zone "underlay" "overlay" static route` on the Spokes.



Please note that the deployment workflow discussed in our [SD-WAN Deployment for MSSPs](#) guide already incorporates all the above recommendations.

ADVPN 2.0

This reference configuration topic contains the following sections:

- [Spoke on page 79](#)
- [Hub on page 80](#)

Spoke

1. Enable ADVPN 2.0 on the overlay SD-WAN Zone, assigning the Performance SLA used to probe the overlay SD-WAN Members:

```
config system sdwan
  config zone
    edit "overlay"
      set advpn-select enable
      set advpn-health-check "HUB"
    next
  end
end
```

2. It is recommended enabling shared idle timeout for the shortcuts towards the same remote Spoke, on all the phase1-interfaces towards the Hubs:

```
config vpn ipsec phase1-interface
  edit "H1_ISP1"
    set shared-idle-timeout enable
  next
  edit "H1_MPLS"
    set shared-idle-timeout enable
  next
end
```

3. For the topologies with segregated transports, configure different Transport Groups for the overlay SD-WAN Members:

```
config system sdwan
  config members
    edit 2
      set interface "H1_INET"
      set zone "overlay"
      set transport-group 1
    next
    edit 3
      set interface "H1_MPLS"
      set zone "overlay"
      set transport-group 2
    next
    edit 4
      set interface "H2_INET"
      set zone "overlay"
      set transport-group 1
    next
    edit 5
      set interface "H2_MPLS"
      set zone "overlay"
      set transport-group 2
    next
  end
end
```

4. In some use cases (such as "Backup of Last Resort"), it is also required to configure the link costs:
 - Must configure the cost both on the overlay SD-WAN Member and on the corresponding phase1-interface

```
config vpn ipsec phase1-interface
    edit "H1_LTE"
        set link-cost 10
    next
    edit "H2_LTE"
        set link-cost 10
    next
end

config system sdwan
    config members
        edit 3
            set interface "H1_LTE"
            set cost 10
        next
        edit 5
            set interface "H2_LTE"
            set cost 10
        next
    end
end
```

Hub

No additional configuration required.

Segmentation over single overlay

This reference configuration topic contains the following sections:

- [Spoke on page 80](#)
- [Hub on page 83](#)
- [Providing Internet access on page 85](#)

Spoke

1. Enable subnet overlap:

```
config system settings
    set allow-subnet-overlap enable
end
```

2. Assign all the WAN-facing interfaces to the PE VRF, including underlays, overlays and the Loopback used for BGP terminationn:

```
config system interface
    edit "Lo"
        set vrf 1
    next
    edit "port1"
        set vrf 1
```

```
next
edit "port4"
    set vrf 1
next
edit "H1_ISP1"
    set vrf 1
next
edit "H1_MPLS"
    set vrf 1
next
edit "H2_ISP1"
    set vrf 1
next
edit "H2_MPLS"
    set vrf 1
next
end
```

3. Assign all the LAN-facing interfaces to the respective CE VRFs:

```
config system interface
edit "port5"
    set vrf 11
next
edit "port6"
    set vrf 12
next
end
```

4. Enable vpn-id-ipip encapsulation on the overlays:

```
config vpn ipsec phase1-interface
edit "H1_ISP1"
    set encapsulation vpn-id-ipip
next
edit "H1_MPLS"
    set encapsulation vpn-id-ipip
next
edit "H2_ISP1"
    set encapsulation vpn-id-ipip
next
edit "H2_MPLS"
    set encapsulation vpn-id-ipip
next
end
```

5. Configure VRF parameters for BGP:

```
config router bgp
config vrf
edit "1"
    set role pe
next
edit "11"
    set role "ce"
    set rd "10.200.1.1:11"
```

```

        set import-rt "65000:11"
        set export-rt "65000:11"
    next
    edit "12"
        set role "ce"
        set rd "10.200.1.1:12"
        set import-rt "65000:12"
        set export-rt "65000:12"
    next
end
end

```



The VRF ID and RT configuration must be identical on all Spokes and Hubs in the SD-WAN network.

The RD configuration must be unique per node (hence using the loopback IP).

6. Configure VPNv4 settings for BGP neighbors and advertise all the CE VRF networks:

```

config router bgp
    config neighbor
        edit 10.200.1.253
            set soft-reconfiguration enable
            set soft-reconfiguration-vpnv4 enable
            set capability-graceful-restart enable
            set capability-graceful-restart-vpnv4 enable
        next
        edit 10.200.1.254
            set soft-reconfiguration enable
            set soft-reconfiguration-vpnv4 enable
            set capability-graceful-restart enable
            set capability-graceful-restart-vpnv4 enable
        next
    end
config network
    edit 1
        set prefix 10.0.1.0/24
    next
    edit 2
        set prefix 10.0.101.0/24
    next
end
end

```

7. If Dynamic BGP is enabled, configure VPNv4 settings also on the dynamic peering:

```

config router bgp
    config neighbor-group
        edit "DYN_EDGE"
            set soft-reconfiguration-vpnv4 enable
            set capability-graceful-restart-vpnv4 enable
            # Ready for multi-regional deployment
            set route-map-out-vpnv4 "LAN_OUT"
        next
    end
end

```

```
end
end
```



The remaining configuration is identical to the standard "Dynamic BGP on Loopback" configuration from an earlier section.

Hub

1. Enable subnet overlap:

```
config system settings
    set allow-subnet-overlap enable
end
```

2. Assign all the WAN-facing interfaces to the PE VRF, including underlays, overlays and the Loopback used for BGP termination:

```
config system interface
    edit "Lo"
        set vrf 1
    next
    edit "port1"
        set vrf 1
    next
    edit "port4"
        set vrf 1
    next
    edit "EDGE_ISP1"
        set vrf 1
    next
    edit "EDGE_MPLS"
        set vrf 1
    next
end
```

3. Assign all the LAN-facing interfaces to the respective CE VRFs:

```
config system interface
    edit "port5"
        set vrf 11
    next
    edit "port6"
        set vrf 12
    next
end
```

4. Enable vpn-id-ipip encapsulation on the overlays:

```
config vpn ipsec phase1-interface
    edit "EDGE_ISP1"
        set encapsulation vpn-id-ipip
    next
    edit "EDGE_MPLS"
        set encapsulation vpn-id-ipip
```

```
    next
end
```

5. Configure VRF parameters for BGP:

```
config router bgp
  config vrf
    edit "1"
      set role pe
    next
    edit "11"
      set role "ce"
      set rd "10.200.1.253:11"
      set import-rt "65000:11"
      set export-rt "65000:11"
    next
    edit "12"
      set role "ce"
      set rd "10.200.1.253:12"
      set import-rt "65000:12"
      set export-rt "65000:12"
    next
  end
end
```



The VRF ID and RT configuration must be identical on all Spokes and Hubs in the SD-WAN network.

The RD configuration must be unique per node (hence using the loopback IP).

6. Configure VPNv4 settings for BGP neighbors and advertise all the CE VRF networks:

```
config router bgp
  config neighbor-group
    edit "EDGE"
      set soft-reconfiguration-vpnv4 enable
      set route-reflector-client-vpnv4 enable
      set capability-graceful-restart-vpnv4 enable
    next
  end
  config network
    edit 2
      set prefix 10.1.0.0/24
    next
    edit 3
      set prefix 10.1.100.0/24
    next
  end
end
```



The remaining configuration is identical to the standard [Dynamic BGP on loopback \(dual-hub region\)](#) on page 72.

Providing Internet access

The following configuration must be added on the Spoke, in order to provide Direct Internet Access to the CE VRF=11:

1. On hardware appliances only, unhide the `npu_link` interface:

```
config system global
    set single-vdom-npuvlink enable
end
```

2. Configure an inter-VRF link between the CE VRF=11 and the PE VRF=1.

On hardware appliances, use the `npu_link` for this purpose.

On virtual appliances, use the software-based `vdom-link` (shown in the snippets below):

```
config system vdom-link
    edit "vrf11_leak"
    next
end
config system interface
    edit "vrf11_leak0"
        set type vdom-link
        set vrf 1
        set vdom 'root'
        set ip 10.200.255.23/31
        set allowaccess ping
    next
    edit "vrf11_leak1"
        set type vdom-link
        set vrf 11
        set vdom 'root'
        set ip 10.200.255.22/31
        set allowaccess ping
    next
end
```

3. Add a static default route to the CE VRF=11, pointing to the newly created inter-VRF link:

```
config router static
    edit 10011
        set gateway 10.200.255.23
        set device vrf11_leak1
    next
end
```

4. In Firewall Policy configuration, allow each traffic hop independently. Below we enable SNAT both on the inter-VRF link and on the Internet-facing interface.

We also use System Zones to simplify Firewall Policy configuration.

```
config system zone
    edit "lan_zone"
        append interface port5
    next
    edit "pevrf_leak_zone"
        append interface "vrf11_leak0"
    next
    edit "vrfs_leak_zone"
```

```

        append interface "vrfl1_leak1"
    next
end
config firewall policy
    edit 2
        set name "Internet (to PE)"
        set srcintf 'lan_zone'
        set dstintf 'vrfs_leak_zone'
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile certificate-inspection
        set application-list "default"
        set logtraffic all
        set nat enable
    next
    edit 3
        set name "Internet (DIA)"
        set srcintf 'pevrf_leak_zone'
        set dstintf "underlay"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile certificate-inspection
        set application-list "default"
        set logtraffic all
        set nat enable
    next

```

Hub-to-edge SD-WAN

This reference configuration topic illustrates the complete solution supporting Hub-to-Spoke sessions and advertising the preferred Hub outside of the SD-WAN network. It contains the following sections:

- [Spoke on page 86](#)
- [Hub on page 87](#)

Spoke

1. Configure the priority-in-sla / priority-out-sla pairs for each SD-WAN overlay member, expressing the desired preferences:

```

config system sdwan
    config members
        # For each overlay member

```

```

        edit 4
            set priority-in-sla ...
            set priority-out-sla ...
        next
        ...
    end
end

```

2. Enable embedding of the SLA information into the probes sent to the Hubs, specifying the SLA target to use:

```

config system sdwan
    config health-check
        edit "HUB"
            set embed-measured-health enable
            set sla-id-redistribute 1
            config sla
                edit 1
                    # ...
                next
            end
        next
    end
end

```

3. Configure the SD-WAN Neighbor feature to apply the route metric based on the current overlay priorities, separately for each Hub:

```

config system sdwan
    config neighbor
        edit "10.200.1.253"
            set member 4 5
            set route-metric priority
            set health-check "HUB"
        next
        edit "10.200.1.254"
            set member 7 8
            set route-metric priority
            set health-check "HUB"
        next
    end
end

```

Hub

1. Create a new remote Health Check (specify all the overlay members):

```

config system sdwan
    config health-check
        edit "EDGE"
            set detect-mode remote
            set members 2 3
            set sla-id-redistribute 1
            config sla
                edit 1
                    set link-cost-factor remote
            end
        end
    end
end

```

```

        # Default priority pair (when not sent by a Spoke)
        set priority-in-sla 5
        set priority-out-sla 995
    next
end
next
end
end

```

2. In the BGP configuration, enable inheriting route priority during recursive resolution:

```

config router bgp
    set recursive-inherit-priority enable
end

```

3. To guarantee correct advertisement outside of the SD-WAN network:

- If you rely on BGP MED, the only additional configuration is to ensure that its value is preserved when readvertising the routes from the Spokes (IBGP) to the external peers (EBGP):

```

config router bgp
    config neighbor
        edit "192.168.1.1"
            set attribute-unchanged med
        next
    end
end

```

- If you rely on another BGP attribute, the translation can be done by matching on the metric (MED). The most common example is to translate the MED into AS_PATH prepending (the exact values depend on the priority pairs configured on the Spokes):

```

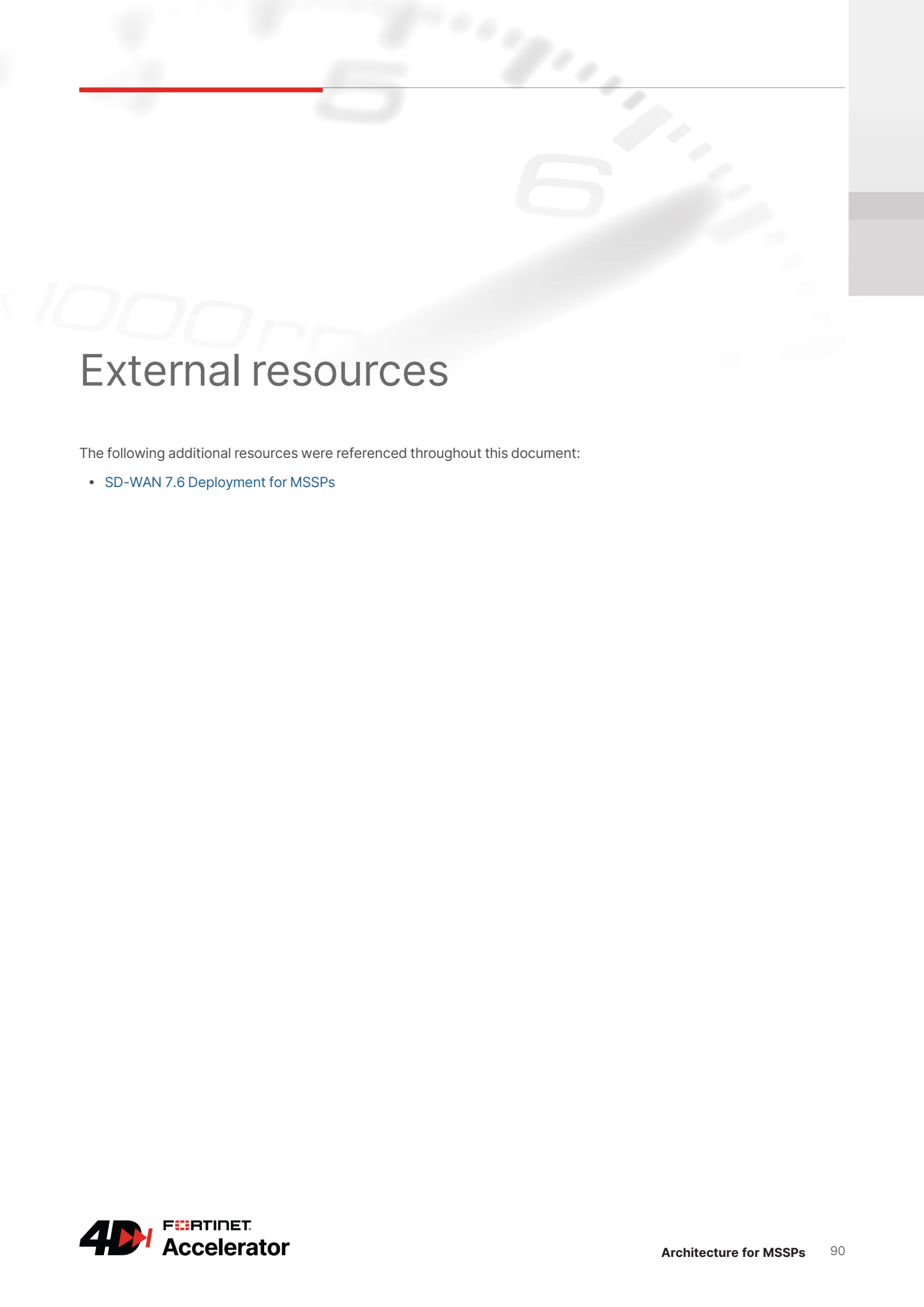
config router route-map
    edit "TO_OUTSIDE"
        config rule
            edit 1
                set match-metric 5
            next
            edit 2
                set match-metric 995
                set set-aspath "65001 65001"
            next
        end
    next
end
config router bgp
    config neighbor
        edit "192.168.1.1"
            set route-map-out "TO_OUTSIDE"
        next
    end
end

```

Product models and firmware versions

The following product models and firmware versions were considered when writing this document:

Product	Model	Firmware
FortiGate (FortiOS)	All models supported by FortiManager	7.6.x
FortiManager	All models	7.6.x
FortiAnalyzer	All models	7.6.x
FortiSwitch	All models supported by FortiGate	7.6.x
FortiAP	All models supported by FortiGate	7.6.x
FortiExtender	All models supported by FortiGate	7.6.x



External resources

The following additional resources were referenced throughout this document:

- [SD-WAN 7.6 Deployment for MSSPs](#)



www.fortinet.com

Copyright © 2025 Fortinet, Inc. All rights reserved. Fortinet®, FortiGate®, FortiCare® and FortiGuard®, and certain other marks are registered trademarks of Fortinet, Inc., and other Fortinet names herein may also be registered and/or common law trademarks of Fortinet. All other product or company names may be trademarks of their respective owners. Performance and other metrics contained herein were attained in internal lab tests under ideal conditions, and actual performance and other results may vary. Network variables, different network environments and other conditions may affect performance results. Nothing herein represents any binding commitment by Fortinet, and Fortinet disclaims all warranties, whether express or implied, except to the extent Fortinet enters a binding written contract, signed by Fortinet's SVP Legal and above, with a purchaser that expressly warrants that the identified product will perform according to certain expressly-identified performance metrics and, in such event, only the specific performance metrics expressly identified in such binding written contract shall be binding on Fortinet. For absolute clarity, any such warranty will be limited to performance in the same ideal conditions as in Fortinet's internal lab tests. Fortinet disclaims in full any covenants, representations, and guarantees pursuant hereto, whether express or implied. Fortinet reserves the right to change, modify, transfer, or otherwise revise this publication without notice, and the most current version of the publication shall be applicable.