Engineering Simplicity

WAN Edge with the Session Smart Router —Juniper Validated Design (JVD)

Published
2025-07-31

Table of Contents

WAN Edge with the Session Smart Router —Juniper Validated Design (JVD)

Juniper Networks Validated Designs provide you with a comprehensive, end-to-end blueprint for deploying Juniper solutions in your network. These designs are created by Juniper's expert engineers and tested to ensure they meet your requirements. Using a validated design, you can reduce the risk of costly mistakes, save time and money, and ensure that your network is optimized for maximum performance.

About this Document

When building a modern software-defined WAN (SD-WAN) environment to overlay existing networks and transport technologies for an enterprise, there are several important design considerations. Juniper WAN edge for Juniper® Session Smart® Routers provides a solution to meet the specific demands of the enterprise. Before implementing a robust VPN with sophisticated SD-WAN path selection features for the enterprise and leveraging these individual designs, some choices need to be made.

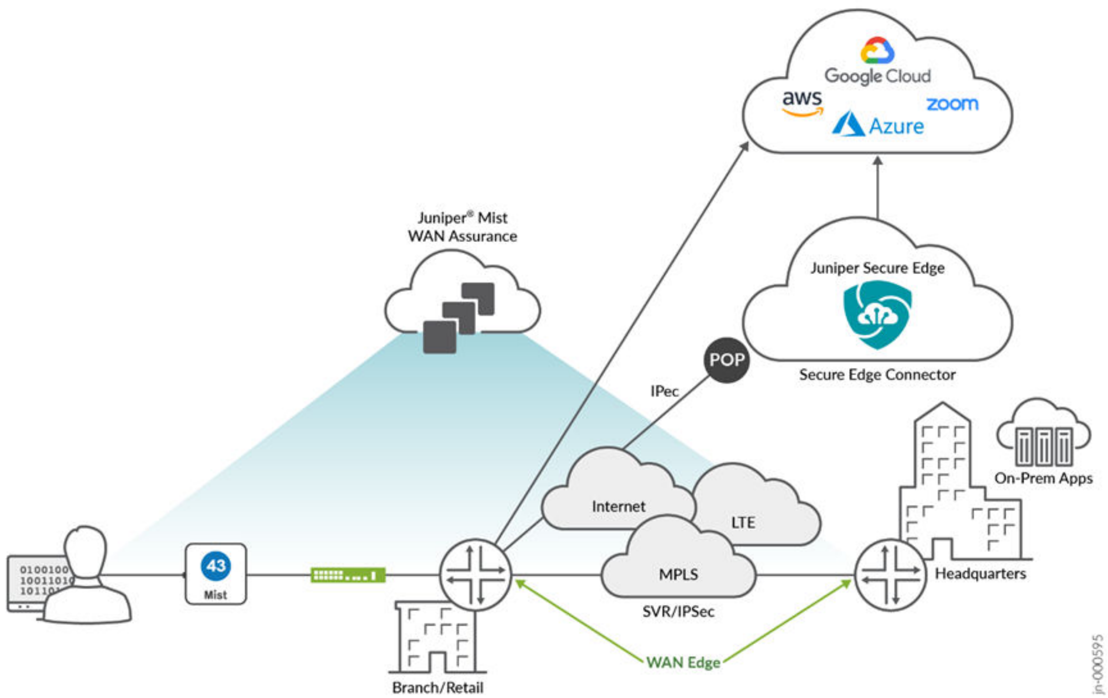
This JVD describes the various ways of WAN edge for Session Smart Router integration and the test cases that are performed to ensure proper integration in an example network design. We provide information about the different topologies and features tested. Additionally, complete configuration examples, using the Juniper Mist™ portal, are provided in the appendix for your reference.

Solution Benefits

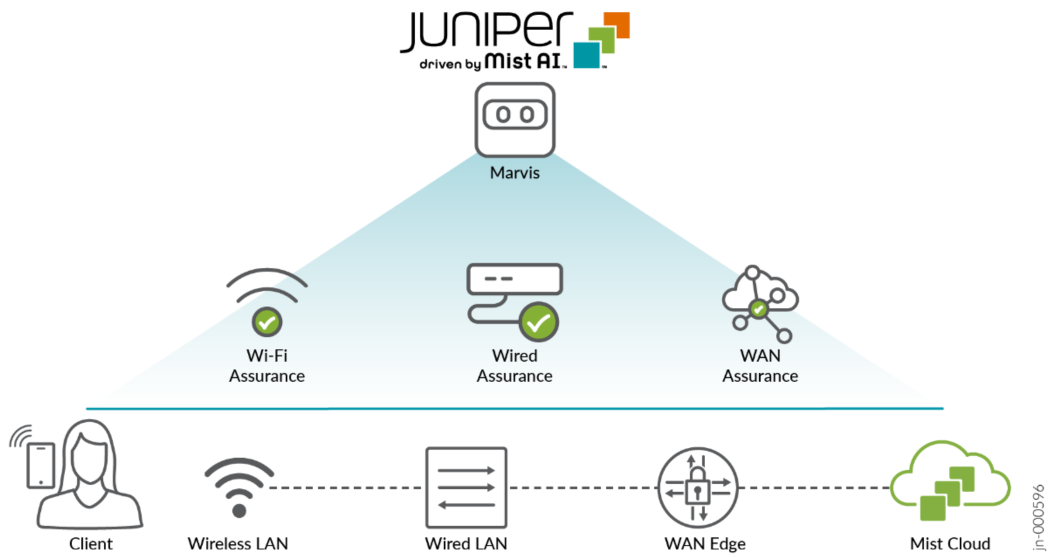
IN THIS SECTION

Introduction to Juniper Mist WAN Assurance

Juniper Mist™ WAN Assurance is a cloud-managed solution designed to optimize and simplify wide area network (WAN) operations. It is part of Juniper Mist's AI-Native networking platform, providing high performance, tunnel-free forwarding, enhanced AI operations, and automation for WAN management.



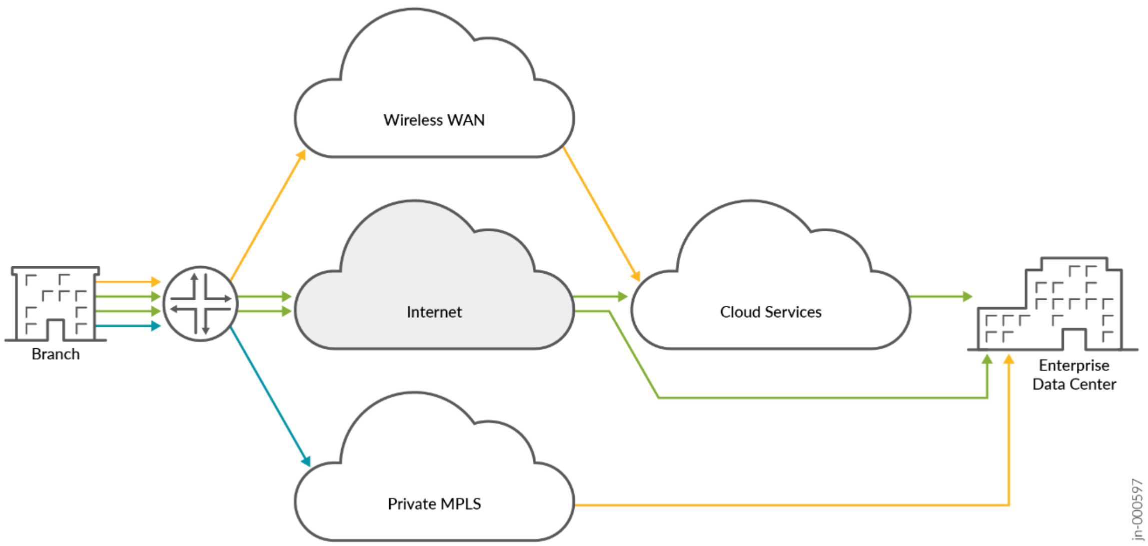
The network links providing site to datacenter, cloud, and public connectivity paths are joined by WAN edge devices to form the fabric of the WAN. The WAN edges are transformed with Juniper's AI-driven SD-WAN solution and act as your distributed policy enforcement points managed centrally from the cloud. Juniper Mist WAN Assurance solves many of the legacy SD-WAN solutions' security, monitoring, and troubleshooting challenges. Integrate Juniper Mist™ Wired Assurance, Juniper Mist™ Wireless Assurance, and now Juniper Mist WAN Assurance into a unified Mist AI™ dashboard to streamline deployment, monitoring, and troubleshooting across your network. Juniper Mist WAN Assurance securely connects branch offices with Session Smart Routers as WAN edges.



Watch the following video for an overview of the Juniper Mist WAN Assurance feature.

Site-to-Site Connectivity (SD-WAN)

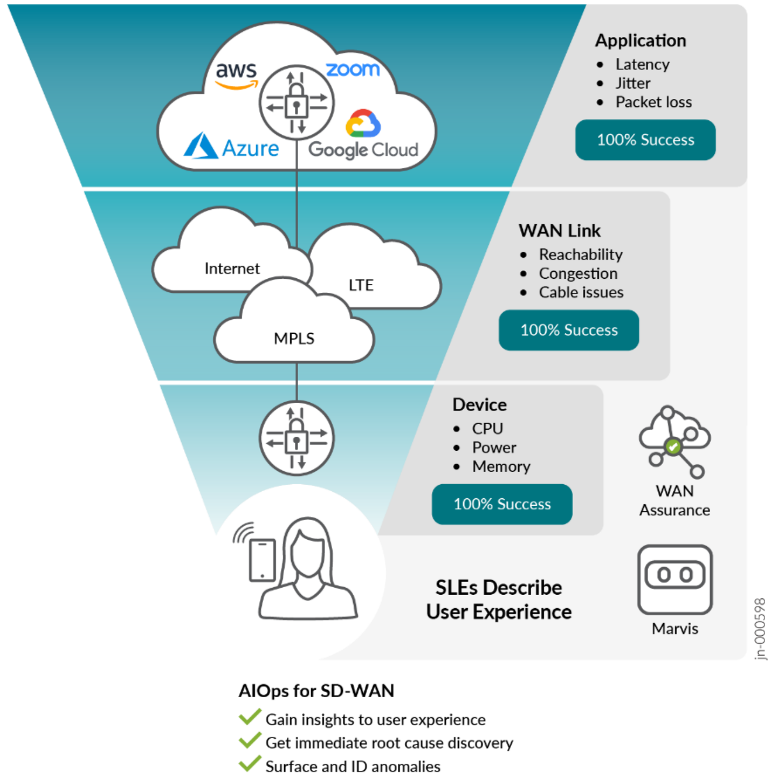
Your WAN edge transforms when integrated with Juniper® SD-WAN driven by Mist AI. Your edge device becomes fast, secure, and application-aware with Juniper Mist WAN Assurance. Through an abstracted overlay, SD-WAN traffic is efficiently routed across a variety of cost-effective broadband service providers, offering a modern alternative to costly legacy MPLS solutions. The architecture ensures uninterrupted service with stateful failovers between diverse connection types such as MPLS, broadband, satellite, and LTE, ensuring seamless transitions for critical applications that are virtually undetectable to end-users. Moreover, Juniper Mist WAN Assurance enriches the WAN edge experience by providing deep visibility into network health, tunnel activity, connectivity, and active session metrics. This strategic insight empowers administrators to fine-tune the network, influencing traffic at the application level to ensure optimal access and enhanced security measures. Such integration epitomizes the shift towards a more agile, intelligent, and cost-effective network infrastructure capable of adapting to the growing demands of modern business landscapes.



Juniper Mist WAN Assurance SLEs

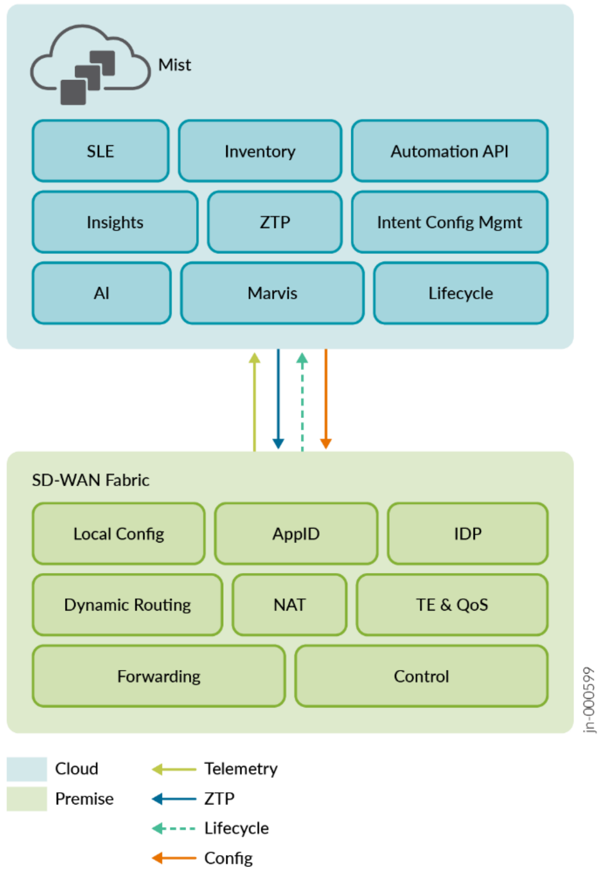
Juniper Mist™ captures, analyzes, correlates, and classifies event and performance data from your network and devices. It then provides you with an assessment of the quality of users' experiences on your network. Many factors contribute to positive or negative user experiences. Juniper Mist organizes these factors into Service-Level Expectations (SLEs). When user experiences fail to meet your SLE success thresholds, Juniper Mist identifies the root cause of each poor experience and provides complete details so that you can address the issues.

By employing WAN SLE metrics such as WAN Edge Health, WAN Link Health, and Application Health, Juniper Mist WAN Assurance adeptly pinpoints the underlying causes of WAN disruptions that negatively impact user experiences. This innovative approach facilitates streamlined operations, enhances visibility into end-user interactions, and simplifies the complexities of monitoring and troubleshooting your network, ultimately driving towards optimal network performance and user satisfaction.



Mist Management Model

Juniper's AI-driven SD-WAN solution unifies the management of branch wireless, wired, and SD-WAN networks within a single, intuitive platform. Experience the simplicity of zero-touch provisioning, life cycle management, and configuration—all seamlessly orchestrated through the comprehensive Juniper Mist dashboard. This integration streamlines operations, enhances network agility, and enables a smarter, more efficient network infrastructure tailored to the modern enterprise.



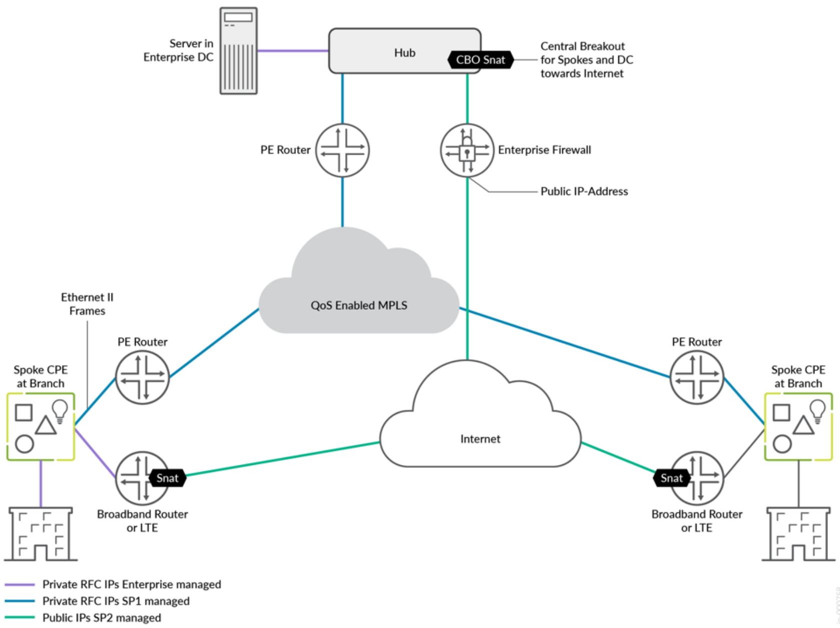
Use Case and Reference Architecture

IN THIS SECTION

In this chapter, we describe an example of SD-WAN implementation in a typical hub-and-spoke scenario that leverages different transport technologies to show how it is implemented. A similar lab was built to test this JVD which you can see in the test report.

Keep the following design goals in mind:

- Design for a hub-and-spoke scenario from day one on. Mesh designs always have scale limitations and are not usually friendly to cheap broadband Internet and LTE connections.
- Ensure your hubs have the right connectivity so that the spokes can reach them using the transport network.
  - If you have an MPLS network, your service provider usually provides a routed private IP address to you, or the end-customers manage their own private IP address range (VPLS).
  - When the device has any broadband or LTE connection, you can assume that there is a kind of source NAT applied on the path or the IP addresses on the local router are not permanent. In this case, the hub must have a statically assigned public IP address that is reachable from the spoke trying to make a secure vector routing (SVR) connection.
- Local country regulations should not filter or restrict communication on destination UDP port 1280. Because these ports must be at least open for spokes and hubs to establish secure vector routing for the overlay network.
- Consider allowing only VPN traffic inside your SD-WAN to lower the overall traffic. All traffic to services outside your VPN should use local breakout at the spoke.
- Use Session Smart Routers for their secure vector routing (SVR) capability, specifically the adaptive-encryption feature. This feature identifies HTTPS traffic that is already encrypted and avoids re-encrypting it for VPN transmission, conserving processing resources. As a result, the more VPN traffic that is already encrypted, the fewer resources need to be provisioned.



A lab that simulates the real world, having two underlay paths, each with different behavior:

- You can emulate an MPLS path (without the MPLS framing in-between) with private IP addresses that are visible end-to-end. In a real-world environment, those private IP addresses are managed and distributed by the MPLS service provider's route reflector. Hub-and-spoke interfaces are assigned static IP addresses.

- An Internet path that is subject to a lot of NAT might make the connection of devices difficult. However, this tends to be what you see in production environments today.

  - Spoke devices get a DHCP address lease from an emulated local broadband router. The emulated router applies symmetric source NAT, especially if the device is connected through Dual-Stack Lite (DS-Lite). This forces the spoke to open a tunnel toward the public IP address of the hub using secure vector routing (UDP destination port 1280).

  - Hub devices get a private static IP address that is assigned to the local interface. In front of the hub, there is an emulated public IP where all spokes must send the traffic to if they want to connect to the hub or the Internet. We then emulate a router or firewall that applies 1:1 NAT forwarding to the private interface IP address of the hub. This emulates the exact behavior you would see when the hub is a VM inside a public cloud. A public cloud provider would not give you the option of assigning a public IP directly to an interface on your hub device.

  - The LTE modem connection of a spoke device is expected to have the same topology requirements. Typically, the mobile service provider (MSP) does some kind of carrier-grade NAT (CGNAT) in its network. However, simulating an LTE Network is tricky as privately owned LTE networks and frequencies are rare. Hence, the simulated broadband router should implement a similar behavior where CGNAT is done in the network before traffic appears on the Internet.

- Both paths are assumed to be completely isolated from each other using an internal firewall. Any intentional cross-path communication needs to leverage the hub which has interfaces on both paths for failover.

Based on these two different path designs, we have implemented and tested five different topologies in this JVD:

- A base hub-and-spoke design with two independent hubs and three spokes. This serves as the foundational topology. The other topologies are extensions or changes to the base topology to achieve other goals. See "Base SD-WAN Topology with Three Spokes and Two Hubs" on page 9.

- A topology where the servers at the hub are not directly attached to the LAN interface and there is a router that is placed between the hub and server. This router then exchanges routes using BGP with the hub to advertise its servers and its VPN-reachable networks. We also enabled a hub-to-hub overlay using the hubs' WAN interfaces to implement a kind of hub redundancy on Layer 3 (L3). See "Extended Topology with Hub Overlay and BGP Peering" on page 12.

- A topology where we form redundant high-availability hub and spokes using the Session Smart Router cluster feature. Those clusters need local Layer 2 (L2) adjacency between the two devices. See "High Availability Hub-and-Spoke Using SSR Chassis Cluster Pairs Topology" on page 12.

- A topology where we add a Juniper Networks® EX Series Switch and a Juniper® Series High-Performance Access Point (AP) at the spoke. This is the most common scenario at a branch where Juniper provided the full-stack networking environment with all components controlled by a single UI in the Juniper Mist™ cloud. The Juniper EX Series Switch can be attached to the Session Smart Router using a single uplink or multiple uplinks via LAG. See "Full Stack Topology with Juniper EX Switch and Juniper Mist AP" on page 155.

- A topology where we extended the above full-stack networking environment with a Juniper EX Series Switch Virtual Chassis and Session Smart Router high-availability cluster. See "Extended Full-Stack Topology with Juniper EX Switch as Virtual Chassis and SSR HA Cluster" on page 14.

Base SD-WAN Topology with Three Spokes and Two Hubs

This lab represents the default structure where we set up the following:

- Installation of three spoke devices

- Installation of two hub devices

- Two underlay paths with different behavior. In the lab, the underlay address range is `192.168.0.0/16`.
  - MPLS path with private IP addresses.
  - Internet path, subjected to NAT.

- An overlay network managed by the enterprise. It is implemented on the LAN side of hub and spokes. In the lab, the overlay address range is `10.0.0.0/8`.

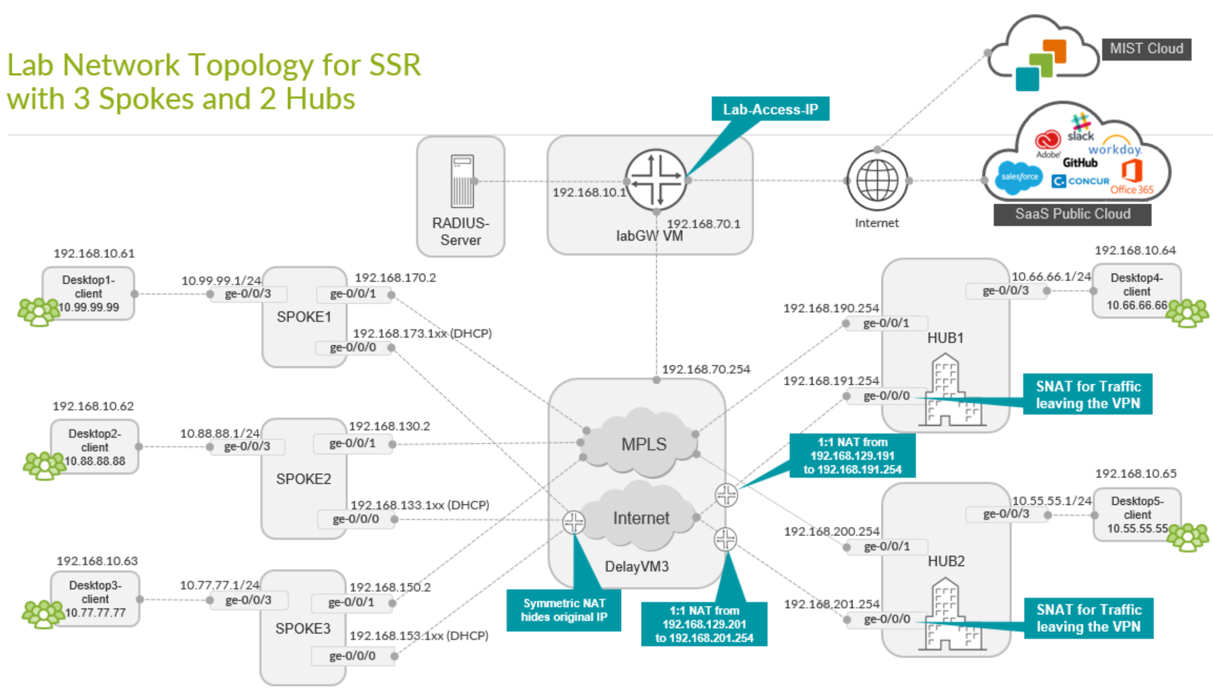
Table 1: Interfaces and IP Addresses Used in this Lab

Device	Interface	IF-Type	Path	IP Address	Assigned	NAT
Spoke1	ge-0/0/0	WAN	INET	192.168.173.1xx	DHCP	symmetric
Spoke1	ge-0/0/1	WAN	MPLS	192.168.170.2	static	none
Spoke1	ge-0/0/3	LAN	VPN	10.99.99.1/24	static	N/A
Spoke2	ge-0/0/0	WAN	INET	192.168.133.1xx	DHCP	symmetric
Spoke2	ge-0/0/1	WAN	MPLS	192.168.130.2	static	none
Spoke2	ge-0/0/3	LAN	VPN	10.88.88.1/24	static	N/A
Spoke3	ge-0/0/0	WAN	INET	192.168.153.1xx	DHCP	symmetric
Spoke3	ge-0/0/1	WAN	MPLS	192.168.150.2	static	none
Spoke3	ge-0/0/3	LAN	VPN	10.77.77.1/24	static	N/A
Hub1	ge-0/0/0	WAN	INET	192.168.191.254	static	Full Cone (1:1) 192.168.129.191
Hub1	ge-0/0/1	WAN	MPLS	192.168.190.254	static	none
Hub1	ge-0/0/3	LAN	VPN	10.66.66.1/24	static	N/A

Table 1: Interfaces and IP Addresses Used in this Lab *(Continued)*

Device	Interface	IF-Type	Path	IP Address	Assigned	NAT
Hub2	ge-0/0/0	WAN	INET	192.168.201.254	static	Full Cone (1:1) 192.168.129.201
Hub2	ge-0/0/1	WAN	MPLS	192.168.200.254	static	none
Hub2	ge-0/0/3	LAN	VPN	10.55.55.1/24	static	N/A

NOTE: In this lab, the emulated public IP addresses are 192.168.129.191 for Hub1 and 192.168.129.201 for Hub2. The spokes connect to these addresses.



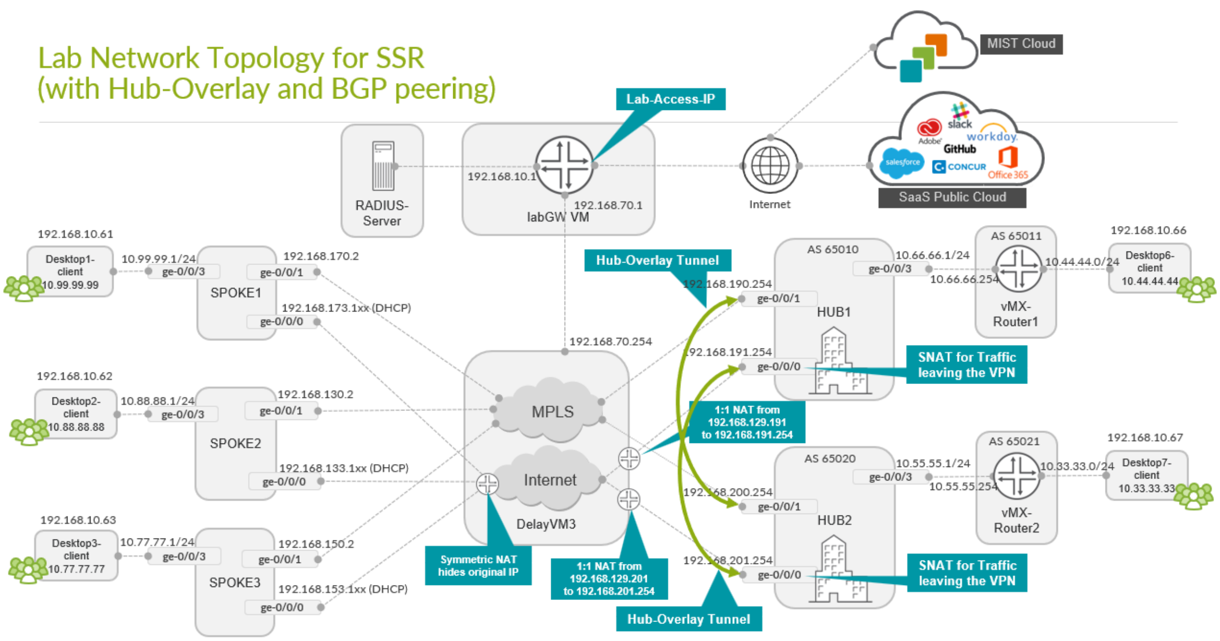
Extended Topology with Hub Overlay and BGP Peering

This is a topology where the servers at the hub are not directly attached to the LAN interface. There is a router that is placed between the hub and the server. This router exchanges routes over BGP with the hub to advertise its servers and the VPN-reachable networks. We also enabled a hub-to-hub overlay, using the WAN interfaces of the hubs, as a means of hub redundancy at L3. This prevents a direct connection between the datacenter routers in case services from Hub1 Datacenter need to communicate to services in Hub2 and vice versa. Instead, those communication can be now established through the WAN interfaces of the two hubs.

The two MX routers attached to the LAN interfaces and the following additional networks:

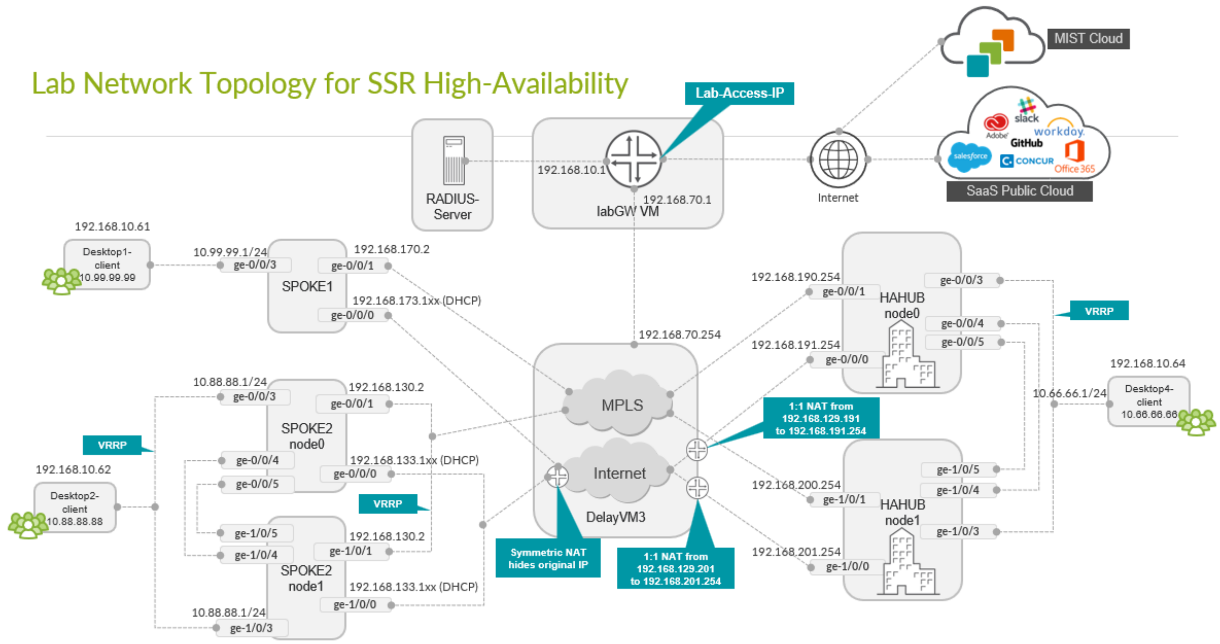
- `10.44.44.0/24` attached to the router of Hub1.
- `10.33.33.0/24` attached to the router of Hub2.

These networks are additionally defined. The hub overlay is an added configuration in the Juniper Mist portal.



High Availability Hub-and-Spoke Using SSR Chassis Cluster Pairs Topology

In this topology, we form redundant high-availability, hub-and-spokes using the Session Smart Router cluster feature. Each cluster is built using the same Session Smart Router device model plus local (L2 adjacency and two additional cables for HA control/fabric. Note that the LAN interfaces are shared with the same IP address and only one link is active at a time using VRRP. On the WAN interfaces, a similar setup is done for MPLS Links as they have a shared static IP address which is not the case for the Internet links.



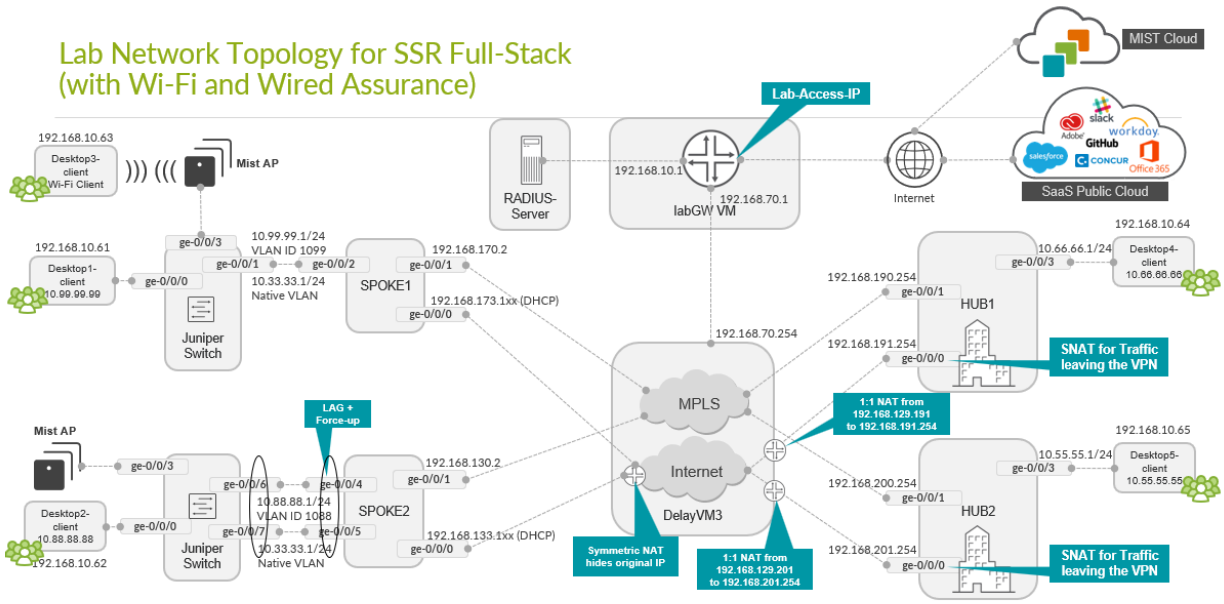
NOTE: This type of deployment for a hub is impossible in most public clouds since you might have a VM-based hub. This is because the strict rules governing public clouds usually do not allow MAC address moves between interfaces. Consider hub overlay instead.

Full Stack Topology with Juniper EX Switch and Juniper Mist AP

In this topology, we are adding Juniper EX Series Switches and Juniper Mist APs to provide an end-to-end, full stack solution to the branch. To boot the EX Series Switch up behind the Session Smart Router as WAN router, we also utilize:

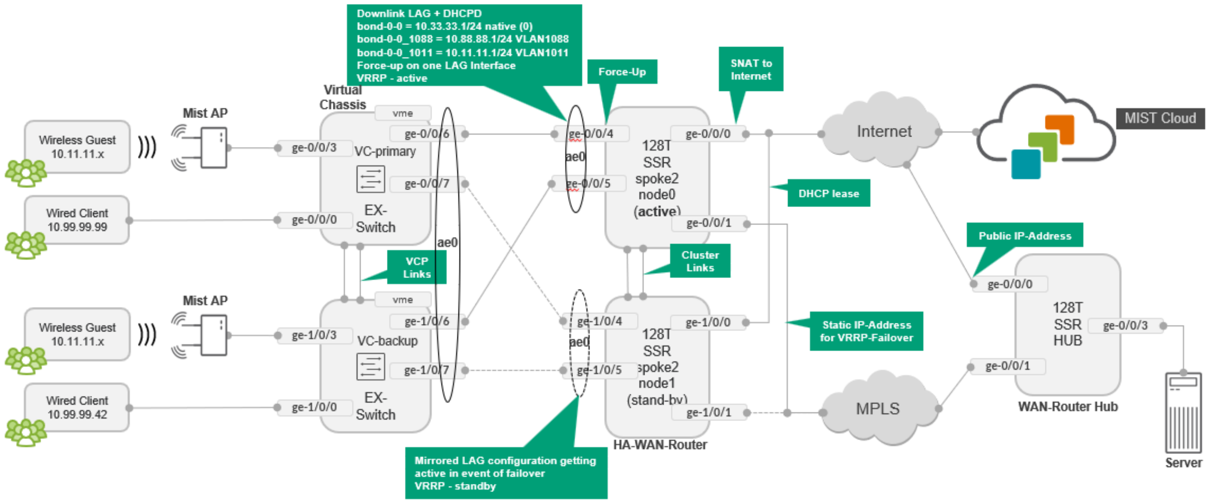
- A DHCP server on the spoke to hand out DHCP address leases to the EX Series Switch, Juniper Mist AP, and all wired and wireless clients.
- One uplink interface between the EX Series Switch and WAN router only.
- Two uplink interfaces between the EX Series Switch and WAN router with LAG and active LACP.

- Support for force-up is required on one of the uplink ports of the WAN router because initially, the LAG configuration on the EX Series Switch is not present. This configuration allows in-band management of the switch through its revenue ports. Without the force-up feature, you would need a dedicated cable from the management port of the switch to the WAN router or a more complex staging method to form the LAG without losing device management.



Extended Full-Stack Topology with Juniper EX Switch as Virtual Chassis and SSR HA Cluster

In this Topology, we extended the above full-stack topology using Juniper EX Switches forming a Virtual Chassis with a minimum of two members. To achieve the same redundancy on the WAN router side, we again formed a high-availability cluster using two Session Smart Routers. Also, a LAG was used from each Session Smart Router WAN router node towards the primary and backup nodes of the Virtual Chassis resulting in four uplinks from the Virtual Chassis.



Validation Framework

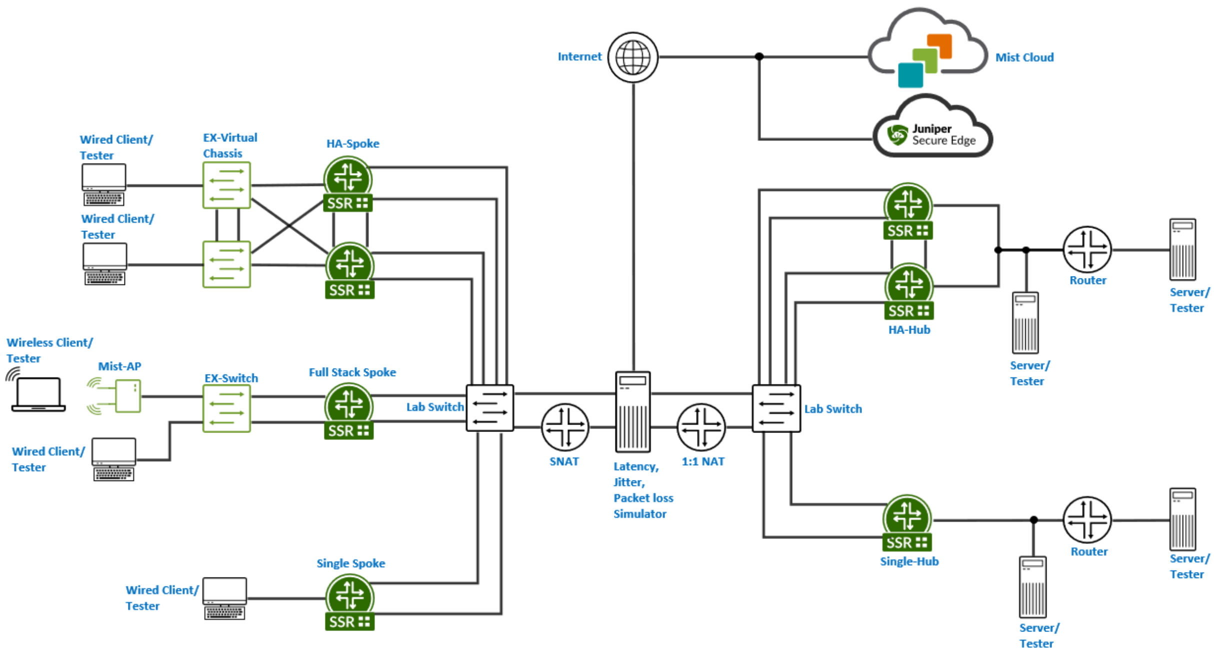
IN THIS SECTION

Test Bed Overview

In a production network, all hubs would need public IP addresses to be reachable for traffic from broadband or LTE networks as their traffic moves through the Internet as transport.

Figure 1 on page 16 shows a lab topology that provides all needed functionality locally without using the Internet for Juniper Mist cloud management and services that are hard to simulate locally such as Outlook 365, Facebook, and so on. A similar lab was built to test the five major topologies and the additional functions that WAN edge provides for Session Smart Routers.

Figure 1: Lab Topology of this JVD



Platforms / Devices Under Test (DUT)

To review the software versions and platforms on which this JVD was validated by Juniper Networks, see the Validated Platforms and Software section in this document.

Test Bed Configuration

We are sharing information about exactly how some of the tests were performed in the appendix section of this document. Contact your Juniper representative to obtain the full archive of the test bed configuration used for this JVD.

Test Objectives

IN THIS SECTION

Test Goals

The testing for this JVD was performed with the following goals. For more information, see the test report of this JVD.

The goal was to test the following features and functions:

- Testing was performed and passed on all five major topologies:
  - Base SD-WAN topology with 3 Spokes and 2 Hubs
  - Extended topology with hub overlay and BGP peering
  - High-availability hub-and-spoke using SSR chassis cluster pairs topology
  - Full-stack topology with Juniper EX Switch and Juniper Mist AP
  - Extended full-stack topology with Juniper EX Switch as Virtual Chassis and SSR HA cluster
- WAN link-related features:
  - Multiple WAN links
  - MTU
  - Auto-negotiation
  - Interface static IP
  - Interface DHCP IP
  - WAN source-NAT interface
  - WAN SLEs
  - Failover when WAN link interface was lost

- LAN link-related features:
  - VLAN tagging
  - DHCP server
  - DHCP-Relay
  - Multiple LANs on same interface (trunk)
  - IEEE 802.3ad LAG with active LACP
  - Using force-up option on one interface for EX Series Switch behind zero-touch provisioning (ZTP).
- VPN overlay features:
  - Spoke-to-hub overlay
  - Hub-to-spoke overlay
  - Spoke-to-spoke overlay (through hub)
  - Hub-to-hub overlays
- Traffic steering and forwarding features:
  - Central breakout at hub
  - Local breakout at spoke
  - Static route at spoke
  - Static route at hub
  - BGP route at hub
  - Failover when remote peer is unavailable (SVR internal BFD to remote)
  - Failover when WAN links no longer meet SLA (latency, jitter, and packet loss)
  - Secure Edge Connector–JSE
- Application policy features:
  - Source-attached LAN
  - Source non-attached user
  - Various applications as defined in the next section
  - IDP-enabled
  - Imported organization application policies

- Applications are defined using the following parameters:
  - Applications defined by IP prefixes
  - Applications defined by protocol and port
  - Applications defined by DNS-FQDN
  - Applications defined by predefined app
  - Applications defined by app categories
- Redundancy and high availability options:
  - Two or more independent hubs with failover at spoke
  - Chassis clustered hub
  - Chassis clustered spoke
  - Hub redundancy using hub overlay
- Security features:
  - Stateful firewalling
  - Application Tracking (AppTrack)
  - Web filtering
  - URL Subcategory
  - IDP engine service chaining
  - Secure Edge Connector
- General options and features:
  - EX Series Switch behind a Session Smart Router as WAN router
  - Juniper AP behind EX Series Switch
  - Site variables
  - Application path visibility
  - WAN edge insights
- Scale testing (see Test Report)

Test Non-Goals

Testing for this JVD was not performed, for various reasons, on the following items:

- No LTE and PPPoE testing were performed. This was a lab limitation.
- Cradlepoint integration. This was a lab limitation.
- Satellite link testing. This was a lab limitation.
- Integration into Campus Fabric as WAN router. It's intended to deliver this in a future version of the JVD-extension for Campus Fabric WAN Router Integration.

Recommendations

- Design for a hub-and-spoke topology from day one. It's the most scalable topology with the least connectivity issues.
- Hubs that need to be reachable through broadband connections, LTE, or other Internet services must have static and public IP addresses (directly or indirect assigned).
- Consider local breakout at the spoke for all services that are reachable on the Internet. Do not burden your VPN with that traffic.
- Check local regulations as they must not filter or restrict communication on destination port 1280 UDP towards the hub. This port is the minimum one needed to set up the secure vector routing between spokes and hubs.
- Avoid creating too many versions of your templates to account for small changes. Instead, make use of site variables to change settings.
- It's recommended to use the first interface of a Session Smart Router (`ge-0/0/0`) to obtain the IP address using a DHCP lease and then to be able to contact the Juniper Mist cloud through the Internet for device management. This will help to simplify the ZTP and onboarding process. Static interface configurations can then follow, if needed.
- Should you change the name of a hub profile after deployment then you also need to edit the WAN Interface configuration on your spoke templates. This is because the names of the VPN endpoints change as well that are needed to establish the VPN.

Appendix: Test Case Information

IN THIS SECTION

This appendix provides a step-by-step guide for creating the base SD-WAN topology, which serves as the foundation for deriving the four additional tested topologies. The base topology addresses common challenges frequently encountered in SD-WAN deployments, such as the use of site variables for scaling and the management of multiple paths typical in modern SD-WAN VPNs.

After the five different topologies, we describe how to build and test all additional features you may or may not use in an SD-WAN design.

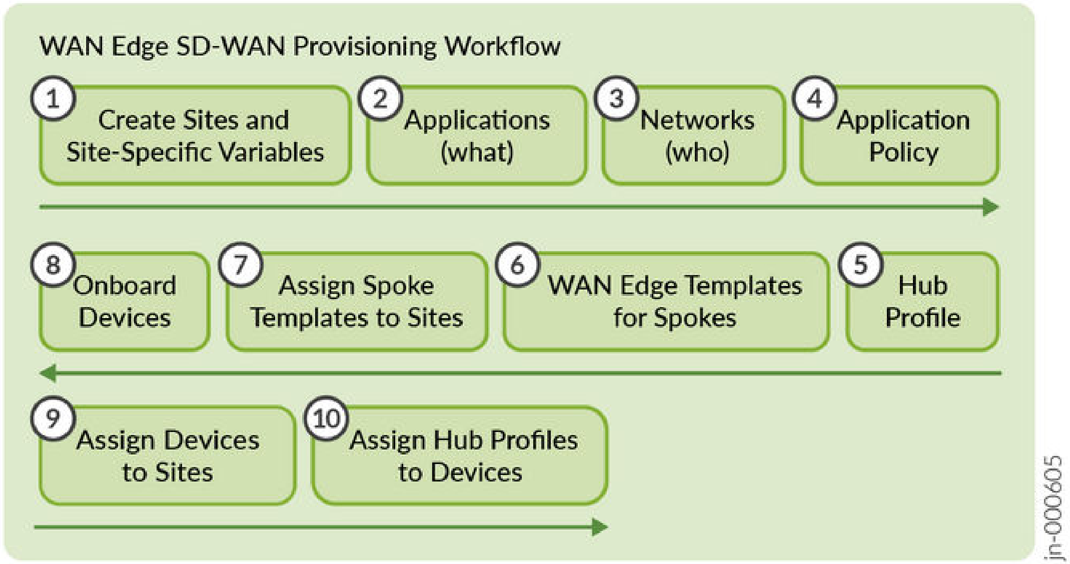
If you need help getting a full description of configuration items, please refer to the WAN-Edge for SSR description on the Juniper website which has all this detailed information.

General WAN Edge Workflow

This overview illustrates how to use the Juniper Mist portal to provision a simple hub-and-spoke network using Session Smart Routers. Conceptually, you can think of the network as an enterprise with branch offices connecting over a provider WAN to on-premises data centers. Examples include an auto parts store, a hospital, or a series of point-of-sale kiosks—anything that requires a remote extension of the corporate LAN for services such as authentication or access to applications.

We assume that before you begin configuring WAN Assurance in your sandbox, you have onboarded your hardware to the Juniper Mist cloud. We also assume and that the physical connections (cabling) needed to support the configuration are in place and that you know the interfaces, and VLANs are valid for your sandbox.

The figure below illustrates the workflow for configuring SD-WAN using the Juniper Mist portal.



The sequence of configuration tasks in this example:

1. Create Sites and Variables—Create a site for the hubs and spokes. Configure site variables for each site. You use these variables later in the templates for WAN edge devices and the hub profile.
2. Configure Applications—Applications are destinations that you define using IP prefixes. Applications represent traffic destinations.
3. Setup Networks—Define the networks. Networks are the source of traffic defined through IP prefixes.
4. Create Application Policies—Application policies determine which networks or users can access which applications, and according to which traffic steering policy.
5. Create hub profiles—You assign hub profile to standalone or clustered devices to automate overlay path creation.
6. Create WAN edge templates—WAN edge templates automatically configure repetitive information such as an IP address, gateway, or VLAN when applied to sites. See Configure WAN Edge Templates.
7. Assign Spoke Templates—Each spoke template needs to be assigned to a site where you intend to launch a spoke.
8. Onboard devices—Onboard your devices so that they appear in the Juniper Mist cloud inventory.
9. Assign devices to Sites—From the inventory, each device must be assigned to its site. For spokes, after this is performed, the template assigns, and the site variables enable the configuration of the spoke to be pushed to the device from Juniper Mist cloud.

10. Complete the onboarding by attaching hub profiles. For each hub profile using the site variables.

Feel free to do additional tasks after this phase like configuring additional features, device updates or Day 2 monitoring of networks, devices and applications.

Appendix: Building a base SD-WAN Topology with Three Spokes and Two Hubs

IN THIS SECTION

We are repeating here the topology and IP address information from above for ease of use.

This lab represents the default structure where we emulate the following:

- Installation of three spoke devices
- Installation of two hub devices
- Two underlay paths with different behavior. In the lab, the underlay address range is `192.168.0.0/16`.
  - MPLS path with private IP addresses.
  - Internet path, subjected to NAT.

- An overlay network managed by the enterprise. It is implemented on the LAN side of hubs and spokes. In the lab, the overlay address range is `10.0.0.0/8`.

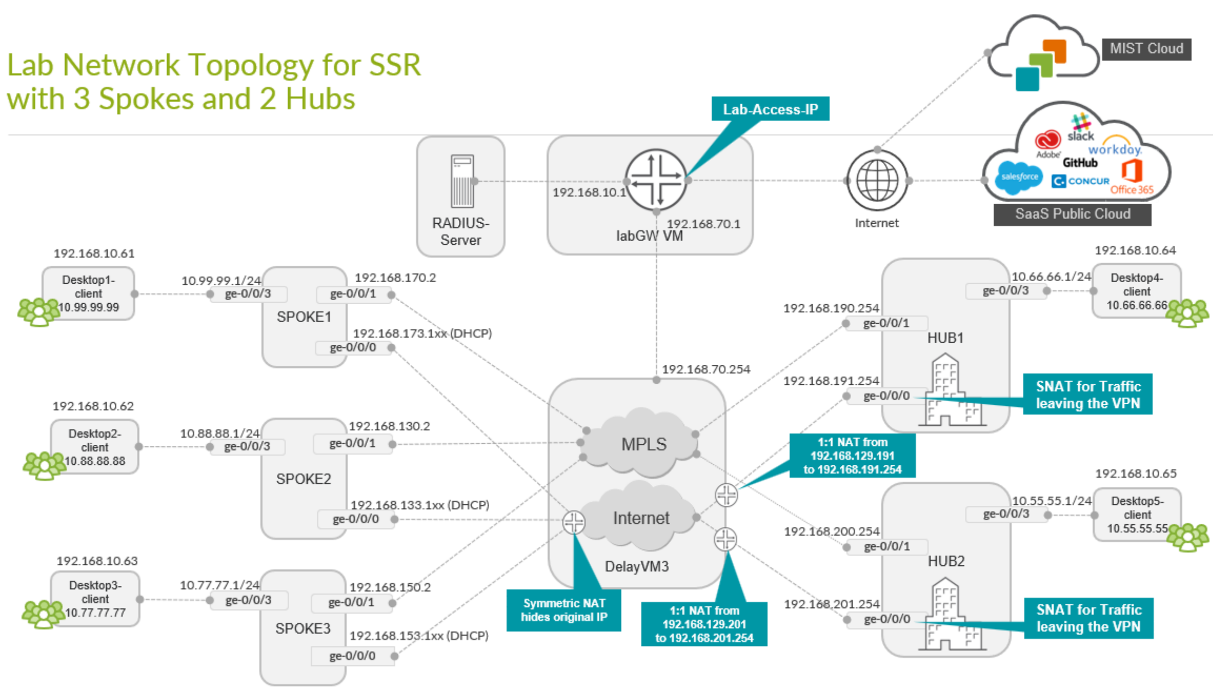
Table 2: Interfaces and IP Addresses Used in this Lab

Device	Interface	IF-Type	Path	IP Address	Assigned	NAT
Spoke1	ge-0/0/0	WAN	INET	192.168.173.1xx	DHCP	symmetric
Spoke1	ge-0/0/1	WAN	MPLS	192.168.170.2	static	none
Spoke1	ge-0/0/3	LAN	VPN	10.99.99.1/24	static	N/A
Spoke2	ge-0/0/0	WAN	INET	192.168.133.1xx	DHCP	symmetric
Spoke2	ge-0/0/1	WAN	MPLS	192.168.130.2	static	none
Spoke2	ge-0/0/3	LAN	VPN	10.88.88.1/24	static	N/A
Spoke3	ge-0/0/0	WAN	INET	192.168.153.1xx	DHCP	symmetric
Spoke3	ge-0/0/1	WAN	MPLS	192.168.150.2	static	none
Spoke3	ge-0/0/3	LAN	VPN	10.77.77.1/24	static	N/A
Hub1	ge-0/0/0	WAN	INET	192.168.191.254	static	Full Cone (1:1) 192.168.129.191
Hub1	ge-0/0/1	WAN	MPLS	192.168.190.254	static	none
Hub1	ge-0/0/3	LAN	VPN	10.66.66.1/24	static	N/A

Table 2: Interfaces and IP Addresses Used in this Lab *(Continued)*

Device	Interface	IF-Type	Path	IP Address	Assigned	NAT
Hub2	ge-0/0/0	WAN	INET	192.168.201.254	static	Full Cone (1:1) 192.168.129.201
Hub2	ge-0/0/1	WAN	MPLS	192.168.200.254	static	none
Hub2	ge-0/0/3	LAN	VPN	10.55.55.1/24	static	N/A

NOTE: In this lab, the emulated public IP addresses are 192.168.129.191 for Hub1 and 192.168.129.201 for Hub2. The spokes connect to these addresses.



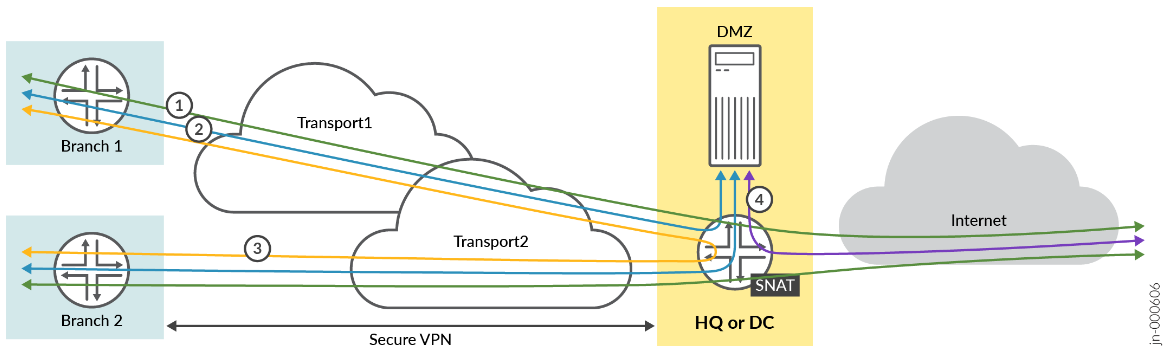
The intent of this lab is to build a VPN between the branch spoke and the hubs with the following design rules:

1. All traffic from the branch spokes that is towards the Internet must go to the hub. On the hub, we enable central breakout using source NAT towards the Internet.

2. Branch spokes must be able to access services and servers located at the hub in the DMZ. This must be bi-directional.

3. Branch spokes can send traffic to other branch spokes but this traffic must be relayed though a hub.

4. Services and servers located at the hub in the DMZ who need to send traffic toward the internet can also use the central breakout using source NAT on the hub.



Creating Sites and Site Variables

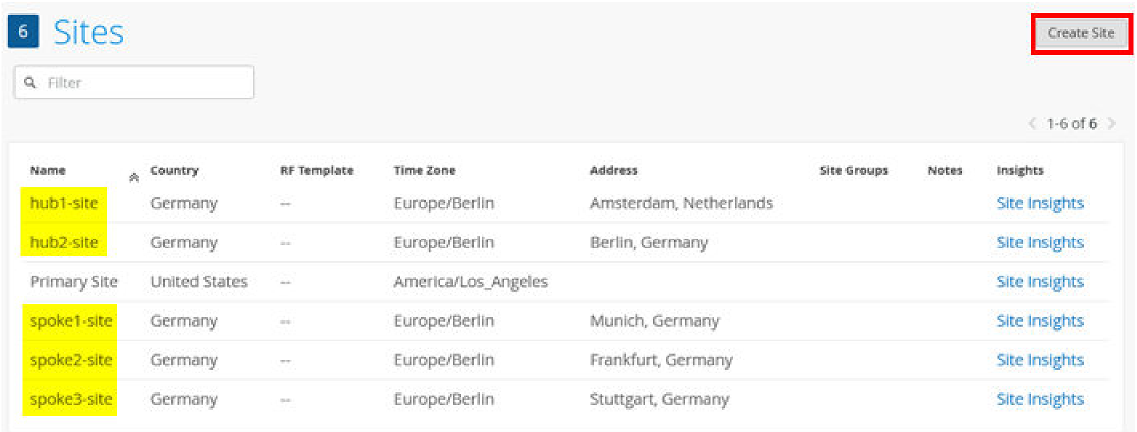
A site is a subset of your organization in the Juniper Mist cloud. You need a unique site for each physical (or logical) location in the network. Users with required privileges can configure and modify sites. The configuration changes in the sites are automatically applied to (or at least available to) all your Session Smart Routers included in the site.

Site variables provide simplicity and flexibility for deployment at a large scale.

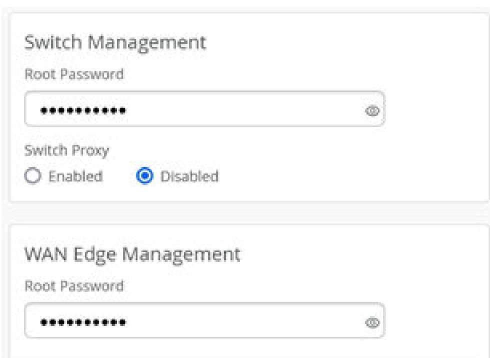
Site variables are configured on a per-site basis. When planning a network design, you can create standard templates for specific WAN edge devices and use variables in templates or the WAN edge configuration page.

Site variables allow you to use tags (for example, "WAN1_PUBIP") as placeholders for actual values (for example, `192.168.200.254`), enabling context-specific configuration. For instance, you can assign WAN1_PUBIP the value `192.168.200.254` at Site 1 and `192.168.190.254` at Site 2. These tags can then be used in Juniper Mist cloud configurations, such as within a WAN edge template. When the template is applied to different sites, the Juniper Mist cloud automatically substitutes the correct IP address for each site during configuration deployment.

First, you need to create five sites for the spokes and hubs. Go to **Organization -> Site Configuration** and add five sites like the ones you see in the below example.



Optional: In each site, we recommend configuring the switch and WAN edge device password.



Then, you need to configure the site variables for each site that are referenced within the templates and profiles.

In our case, the site variables are configured in the following way:

- The variables such as `{{SPOKE_LAN1_PFX}}`, `{{HUB1_LAN1_PFX}}`, `{{HUB2_LAN1_PFX}}`, `{{WAN0_PFX}}` and `{{WAN1_PFX}}` represent the first three octets of an IP address or a prefix.
- The variables such as `{{SPOKE_LAN1_VLAN}}`, `{{HUB1_LAN1_VLAN}}`, `{{HUB2_LAN1_VLAN}}` contain the individual VLAN IDs. In this example, use VLAN tagging to break up the broadcast domain and separate the traffic.
- The variables `{{WAN0_PUBIP}}` and `{{WAN1_PUBIP}}` defined for the WAN interfaces of hubs use the public IP address:
  - The IP address of interfaces on the Internet path is in 192.168.129.x format. You can set up NAT rules for the interface.
  - The IP address of interfaces on the MPLS path is in 192.168.x.254.
- Use the /24 subnet mask and do not create a variable for this field.

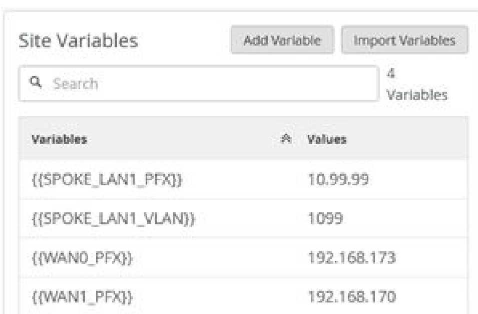
The full table for all sites matching our above topology is:

Site Name	Variable	Value	Remark
spoke1-site	{{SPOKE_LAN1_PFX}}	10.99.99	
spoke1-site	{{SPOKE_LAN1_VLAN}}	1099	
spoke1-site	{{WAN0_PFX}}	192.168.173	Not used in template yet
spoke1-site	{{WAN1_PFX}}	192.168.170	
spoke2-site	{{SPOKE_LAN1_PFX}}	10.88.88	
spoke2-site	{{SPOKE_LAN1_VLAN}}	1088	
spoke2-site	{{WAN0_PFX}}	192.168.133	Not used in template yet
spoke2-site	{{WAN1_PFX}}	192.168.130	
spoke3-site	{{SPOKE_LAN1_PFX}}	10.77.77	
spoke3-site	{{SPOKE_LAN1_VLAN}}	1077	
spoke3-site	{{WAN0_PFX}}	192.168.153	Not used in template yet
spoke3-site	{{WAN1_PFX}}	192.168.150	
hub1-site	{{HUB1_LAN1_PFX}}	10.66.66	
hub1-site	{{HUB1_LAN1_VLAN}}	1066	
hub1-site	{{WAN0_PFX}}	192.168.191	
hub1-site	{{WAN0_PUBIP}}	192.168.129.191	
hub1-site	{{WAN1_PFX}}	192.168.190	

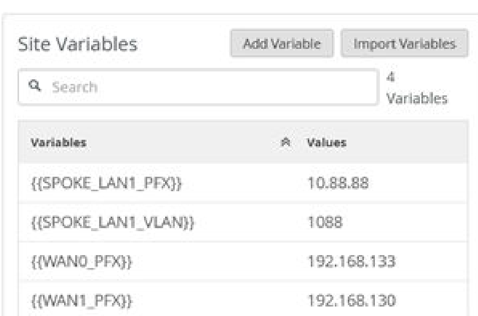
(Continued)

Site Name	Variable	Value	Remark
hub1-site	{{WAN1_PUBIP}}	192.168.190.254	Same as private interface IP
hub2-site	{{HUB1_LAN1_PFX}}	10.55.55	
hub2-site	{{HUB1_LAN1_VLAN}}	1055	
hub2-site	{{WAN0_PFX}}	192.168.201	
hub2-site	{{WAN0_PUBIP}}	192.168.129.201	
hub2-site	{{WAN1_PFX}}	192.168.200	
hub2-site	{{WAN1_PUBIP}}	192.168.200.254	Same as private interface IP

For **spoke1-site**, configure the following site variables:



For **spoke2-site**, configure the following site variables:



For **spoke3-site** configure the following site variables:

Site Variables — Add Variable | Import Variables

Search — 4 Variables

Variables	Values
{{SPOKE_LAN1_PFX}}	10.77.77
{{SPOKE_LAN1_VLAN}}	1077
{{WAN0_PFX}}	192.168.153
{{WAN1_PFX}}	192.168.150

For **hub1-site** configure the following site variables:

Site Variables — Add Variable | Import Variables

Search — 6 Variables

Variables	Values
{{HUB1_LAN1_PFX}}	10.66.66
{{HUB1_LAN1_VLAN}}	1066
{{WAN0_PFX}}	192.168.191
{{WAN0_PUBIP}}	192.168.129.191
{{WAN1_PFX}}	192.168.190
{{WAN1_PUBIP}}	192.168.190.254

For **hub2-site**, configure the following site variables:

Site Variables — Add Variable | Import Variables

Search — 6 Variables

Variables	Values
{{HUB2_LAN1_PFX}}	10.55.55
{{HUB2_LAN1_VLAN}}	1055
{{WAN0_PFX}}	192.168.201
{{WAN0_PUBIP}}	192.168.129.201
{{WAN1_PFX}}	192.168.200
{{WAN1_PUBIP}}	192.168.200.254

Configure Applications

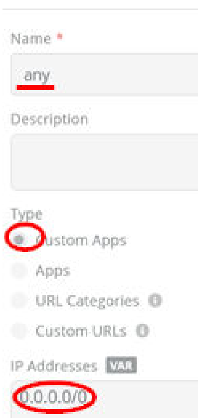
Applications represent traffic destinations. On the Session Smart Router, applications create services in the background for SVR. Applications can be ports, protocols, prefixes, custom domains, or app names from the built-in AppID library.

Applications are the services or apps that your network users will connect to in a Juniper Mist WAN Assurance design. You can define these applications manually in the Juniper Mist portal. You define applications by selecting the category (such as social media) or by selecting individual applications (such as Microsoft Teams) from a list. Another option is to use the predefined list of common traffic types. You can also create a custom application to describe anything that is not otherwise available.

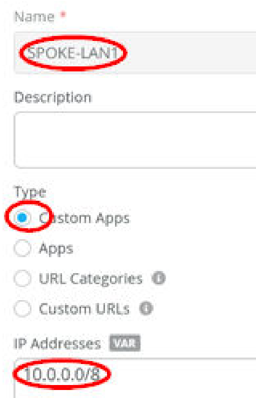
For users to access applications, you must first define the applications and then use application policies to permit or deny access. That is, you associate these applications with users and networks and then assign a traffic-steering policy and access rule (allow or deny).

All applications we are going to use here for now are only IP address destination prefix based applications. Those are the minimum required ones for building a VPN.

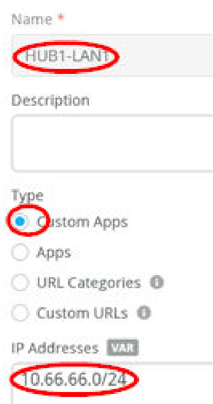
Go to **Organization -> Applications**. Then, check if there is a predefined application with the name “any” defining a custom `0.0.0.0/0` IP address range. If that is not the case yet, define it yourself.



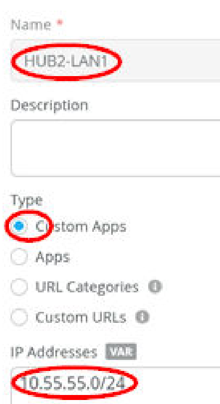
Secondly, we configure a match criterion for all IP addresses inside the corporate VPN used. Those are typically assigned directly or indirectly to all LAN interfaces of our hubs and spokes. Add an application with the name set to “SPOKE1-LAN” and under IP addresses, just configure the single IP prefix `10.0.0.0/8`. At the start, we only use the 3 IP prefixes `10.77.77.0/24`, `10.88.88.0/24`, and `10.99.99.0/24` and we could only configure those, but such a wildcard match criteria would allow easy extensions in the future with no need to change a ruleset to all devices in your environment.



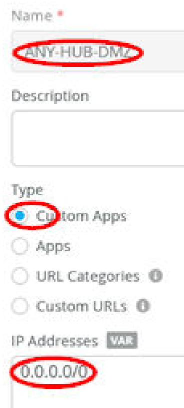
Third, we configure a match criterion for all IP addresses attached at the LAN interface of Hub1. Add an application with the name set to “HUB1-LAN1” and under IP addresses, just configure the single IP prefix `10.66.66.0/24` for now.



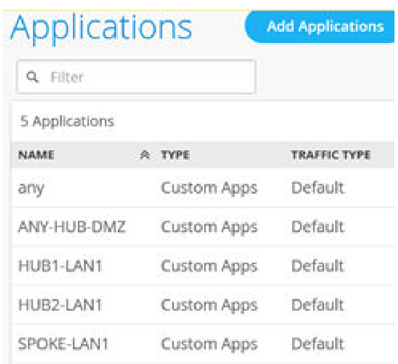
Fourth, we configure a match criterion for all IP addresses attached at the LAN interface of Hub2. Add an application with the name set to “HUB2-LAN1” and under IP addresses, just configure the single IP prefix `10.55.55.0/24` for now.



Fifth, we again configure a catch-up for all IP addresses. Add an application with the name set to "ANY-HUB-DMZ" and under IP addresses, just configure the single IP prefix `0.0.0.0/0`. You might wonder why we do this here again as we already define the same in the first rule with the name "any". This is a trick we do if you have the same traffic forwarding desire, but the origin of the traffic comes from different Interfaces into the system.



The result should look like the figure below:



Configure Networks

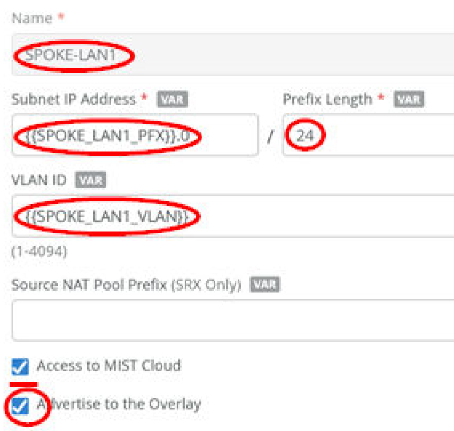
Networks are sources of the request in your Juniper WAN Assurance design. On the Session Smart Router, networks create tenants in the background for SVR and the Session Smart Router identifies tenants at the logical interface (network interface). LAN and WAN interface configurations identify your tenants.

Once you have created networks in the Juniper Mist portal, you can use networks across the entire organization in the portal. WAN Assurance design uses networks as the source in the application policy.

Go to **Organization -> Networks**. Configure the first network in the following way:

- Name=`SPOKE-LAN1`

- Subnet IP Address=`{{SPOKE_LAN1_PFX}}.0` this will substitute the value from the referenced site variable that contains the first three octets.
- Prefix Length=`24` (we only use /24 netmask in our example for ease of use).
- VLAN ID=`{{SPOKE_LAN1_VLAN}}` to automatically use the right tag via the site variable.
- Access to Mist Cloud=`Checked/Enabled`. We want possible future devices to be able to be managed by the Juniper Mist cloud and have the right policy set.
- Advertised via Overlay=`Checked/Enabled`



Then, configure the second network in the following way:

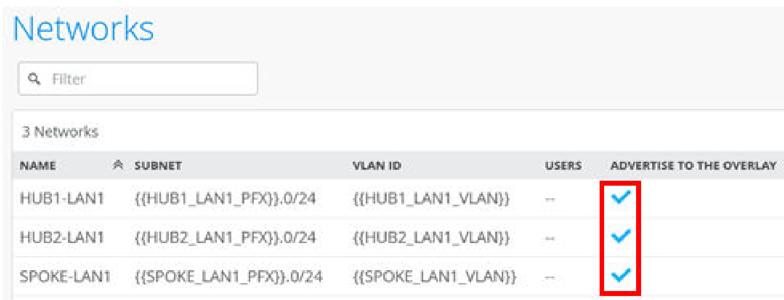
- Name=`HUB1-LAN1`
- Subnet IP Address=`{{HUB1_LAN1_PFX}}.0` this will substitute the value from the referenced site variable that contains the first three octets.
- Prefix Length=`24` (we only use /24 netmask in our example for ease of use).
- VLAN ID=`{{HUB1_LAN1_VLAN}}` to automatically use the right tag via site variable.
- Access to Mist Cloud=`Checked/Enabled`. We want possible future devices to be able to be managed by the Juniper Mist cloud and have the right policy set.
- Advertised via Overlay=`Checked/Enabled`

Then configure the third network in the following way:

- Name=`HUB2-LAN1`
- Subnet IP Address=`{{HUB2_LAN1_PFX}}.0` this will substitute the value from the referenced site variable that contains the first three octets.
- Prefix Length=`24` (we only use /24 netmask in our example for ease of use).

- VLAN ID=`{{HUB2_LAN1_VLAN}}` to automatically use the right tag via site variable.
- Access to Mist Cloud=`Checked/Enabled`. We want possible future devices to be able to be managed by the Juniper Mist cloud and have the right policy set.
- Advertised via Overlay=`Checked/Enabled`

The result should look like the figure below:



Create the Hub Profile for the First Hub

Each hub device in a Juniper Mist cloud topology must have its own profile. Hub profiles are a convenient way to create an overlay and assign a path for each WAN link on that overlay in Juniper Mist WAN Assurance.

The key difference between a hub profile and a WAN edge template lies in their scope and application. A hub profile is applied to a specific device at a hub site, while a WAN edge template is used across multiple spoke sites, each potentially with multiple devices, all sharing the same template. Each WAN interface on a hub creates an overlay endpoint for spoke connections, and the spoke WAN interfaces map to the appropriate hub interfaces, thereby defining the topology. Hub profiles control the creation and removal of overlay paths.

A hub profile comprises a set of attributes that associate with a particular hub device. Hub profiles include name, LAN, WAN, traffic steering, application policies, and routing options. You can assign the hub profile to a hub device and after a hub profile is loaded onto the site, the device assigned to the site picks up the attributes of that hub profile.

Go to **Organization -> Hub Profiles.**

Here, you have two options:

1. Create the hub profile by importing an existing JSON definition. This is the best way to repeat this example without making errors or forgetting a setting.
2. Create your own profile and do all the needed configuration in the Juniper Mist portal.

Should you choose to use the import option, click on **Import Profile** and import the below JSON as a file.


```
{
  "dhcpd_config": {
    "enabled": true
  },
  "ntpOverride": true,
  "dnsOverride": true,
  "service_policies": [
    {
      "name": "spoke-to-hub-dmz",
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "HUB1-LAN1"
      ],
      "action": "allow",
      "path_preference": "HUB-LANS",
      "idp": {
        "enabled": false
      }
    },
    {
      "name": "hub-dmz-to-spoke",
      "tenants": [
        "HUB1-LAN1"
      ],
      "services": [
        "SPOKE-LAN1"
      ],
      "action": "allow",
      "local_routing": true,
      "idp": {
        "enabled": false
      }
    },
    {
      "name": "spoke-to-spoke-hairpin",
      "tenants": [
        "SPOKE-LAN1"
      ],
```

```
      "services": [
        "SPOKE-LAN1"
      ],
      "action": "allow",
      "local_routing": true
    },
    {
      "tenants": [
        "HUB1-LAN1"
      ],
      "services": [
        "ANY-HUB-DMZ"
      ],
      "action": "allow",
      "path_preference": "CBO",
      "name": "hub-dmz-to-internet",
      "idp": {
        "enabled": false
      }
    },
    {
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "any"
      ],
      "action": "allow",
      "name": "spokes-traffic-cbo-on-hub",
      "path_preference": "CBO",
      "idp": {
        "enabled": false
      }
    }
  ],
  "ip_configs": {
    "HUB1-LAN1": {
      "type": "static",
      "ip": "{{HUB1_LAN1_PFX}}.1",
      "netmask": "/24"
    }
  },
  "dns_servers": [
```

```
    "8.8.8.8",
    "9.9.9.9"
  ],
  "port_config": {
    "ge-0/0/0": {
      "name": "INET",
      "usage": "wan",
      "wan_type": "broadband",
      "aggregated": false,
      "redundant": false,
      "traffic_shaping": {
        "enabled": false
      },
      "wan_ext_ip": "{{WAN0_PUBIP}}",
      "ip_config": {
        "type": "static",
        "ip": "{{WAN0_PFX}}.254",
        "netmask": "/24",
        "gateway": "{{WAN0_PFX}}.1"
      },
      "vpn_paths": {
        "hub1-INET.OrgOverlay": {
          "role": "hub"
        }
      }
    },
    "ge-0/0/1": {
      "name": "MPLS",
      "usage": "wan",
      "wan_type": "broadband",
      "aggregated": false,
      "redundant": false,
      "traffic_shaping": {
        "enabled": false
      },
      "wan_ext_ip": "{{WAN1_PUBIP}}",
      "ip_config": {
        "type": "static",
        "ip": "{{WAN1_PFX}}.254",
        "netmask": "/24",
        "gateway": "{{WAN1_PFX}}.1"
      },
      "vpn_paths": {
```

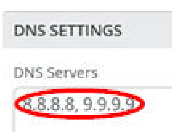
```
        "hub1-MPLS.OrgOverlay": {
          "role": "hub"
        }
      }
    },
    "ge-0/0/3": {
      "usage": "lan",
      "networks": [
        "HUB1-LAN1"
      ]
    }
  },
  "bgp_config": {},
  "routing_policies": {},
  "extra_routes": {},
  "path_preferences": {
    "HUB-LANS": {
      "strategy": "ordered",
      "paths": [
        {
          "type": "local",
          "networks": [
            "HUB1-LAN1"
          ]
        }
      ]
    },
    "CBO": {
      "strategy": "ordered",
      "paths": [
        {
          "name": "INET",
          "type": "wan"
        }
      ]
    }
  },
  "ospf_areas": {},
  "vrf_instances": {},
  "tunnel_configs": {},
  "oob_ip_config": {
    "type": "dhcp",
    "node1": {
```

```
      "type": "dhcp"
    }
  },
  "tunnel_provider_options": {
    "jse": {},
    "zscaler": {}
  },
  "ospf_config": {
    "enabled": false,
    "areas": {}
  },
  "name": "hub1",
  "type": "gateway"
}
```


Should you decide to configure everything manually in the Juniper Mist portal, then use the following steps:

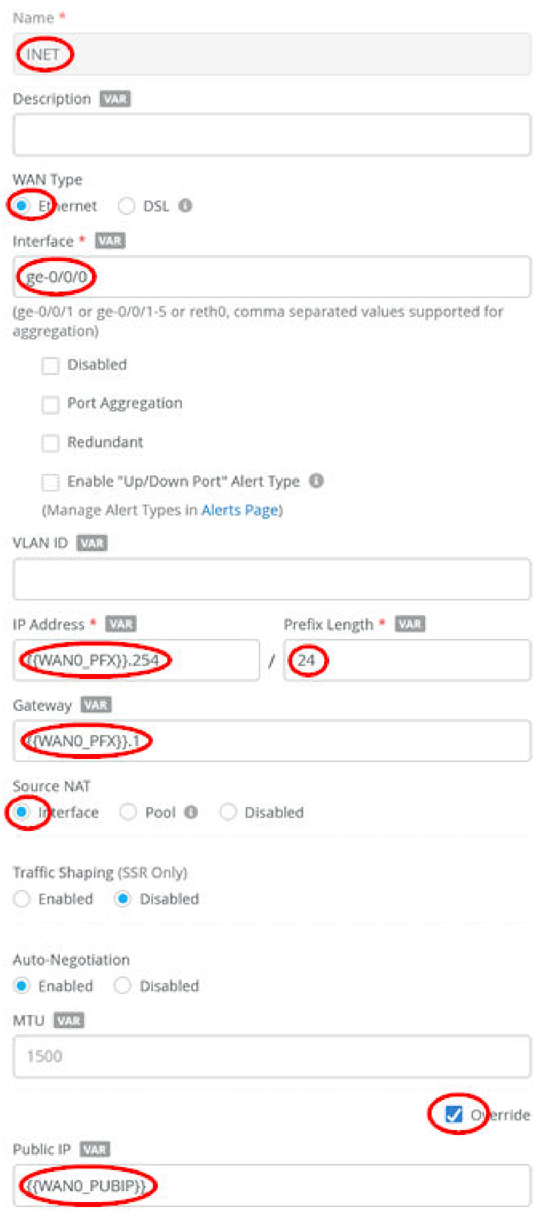
Edit the DNS Settings:

- DNS Servers=`8.8.8.8, 9.9.9.9`



Configure a first **WAN interface** as follows:

- Name=`INET` this indicates which topology it's going to use.
- WAN Type=`Ethernet`
- Interface=`ge-0/0/0`
- IP Address=`{{WAN0_PFX}}.254`
- Prefix Length=`24`
- Gateway=`{{WAN0_PFX}}.1`
- Source NAT=`Interface`
- Override for Public IP=`Checked/Enabled`
- Public IP=`{{WAN0_PUBIP}}`
- The Overlay Hub Endpoint will be automatically generated and should be "hub1-INET".

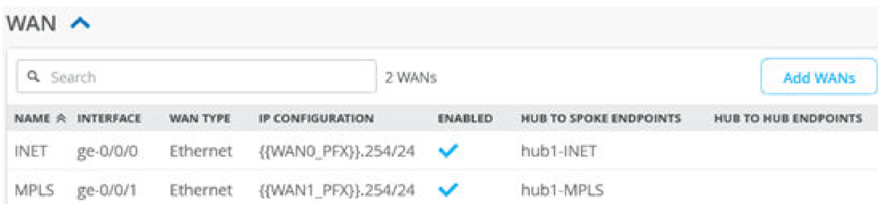


Configure a second WAN interface as follows:

- Name=`MPLS` this indicates which topology it's going to use.
- WAN Type=`Ethernet`
- Interface=`ge-0/0/1`
- IP Address=`{{WAN1_PFX}}.254`
- Prefix Length=`24`
- Gateway=`{{WAN1_PFX}}.1`

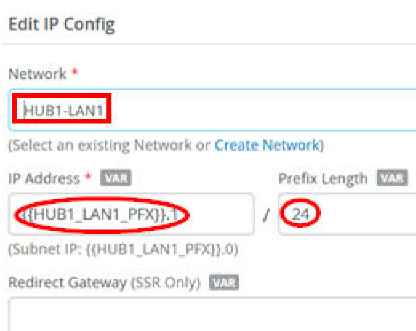
- Source NAT=`Interface`
- Override for Public IP=`Checked/Enabled`
- Public IP=`{{WAN1_PUBIP}}`
- The Overlay Hub Endpoint will be automatically generated and should be “hub1-MPLS”.

The result should look like the figure below:



Add a LAN IP config now with the following configuration:

- Network=`HUB1-LAN1`
- IP Address=`{{HUB1_LAN1_PFX}}.1`
- Prefix Length=`24`

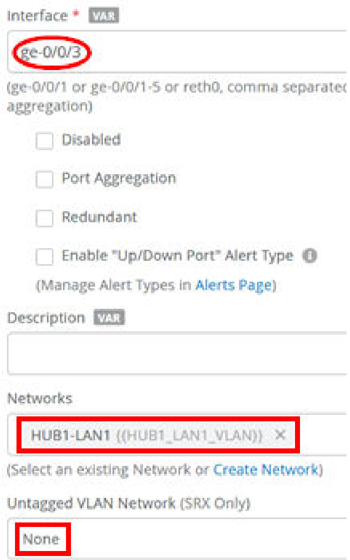


The result should look like the figure below:

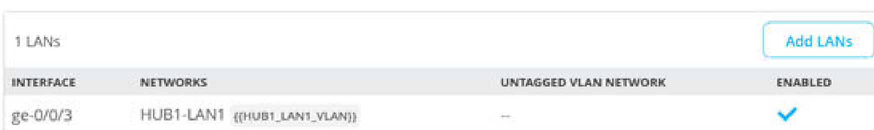
Add a LAN interface now with the following configuration:

- Interface=`ge-0/0/3`
- Networks=`HUB1-LAN1`

- Untagged VLAN=`None`



The result should look like the figure below:

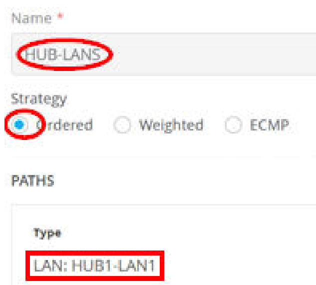


Traffic steering is where you define the different paths that application traffic can take to traverse the network. The paths that you configure within traffic steering determine the destination zone. For any traffic steering policy, you need to define the paths for traffic to traverse and strategies for utilizing those paths. Strategies include:

- Ordered—Starts with a specified path and failover to backup path(s) when needed.
- Weighted—Distributes traffic across links according to a weighted bias, as determined by a cost that you input.
- Equal-cost multipath—Load balances traffic equally across multiple paths.

Now we need to define two traffic steering rules. The first rule has the following configuration:

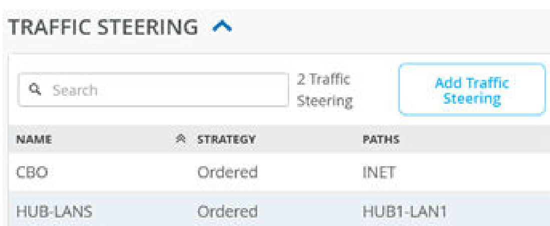
- Name=`HUB-LANS`
- Strategy=`Ordered`
- Paths
  - Path1 Type=`LAN: HUB1-LAN1`



The second rule has the following configuration:

- Name=`CBO`
- Strategy=`Ordered`
- Paths
  - Path1 Type=`WAN: INET`

The result should look like the figure below:



Application policies are where you define which network end users can access which applications, and according to which traffic-steering policy. The settings in Networks/Users determine the source zone. The Applications and Traffic Steering path settings determine the destination this traffic should be sent to. Additionally, you can assign a policy action—permit or deny—allowing or blocking traffic.. In our case, the following application policies are needed to implement the design rules of the VPN.

> **NOTE**: Some rules do not include a Traffic Steering policy, as it is not required for Session Smart Routers—unlike when managing a Juniper Networks® SRX Series Firewall. In these cases, the routing destination is determined by the automatically installed BGP routes within the overlay VPN.

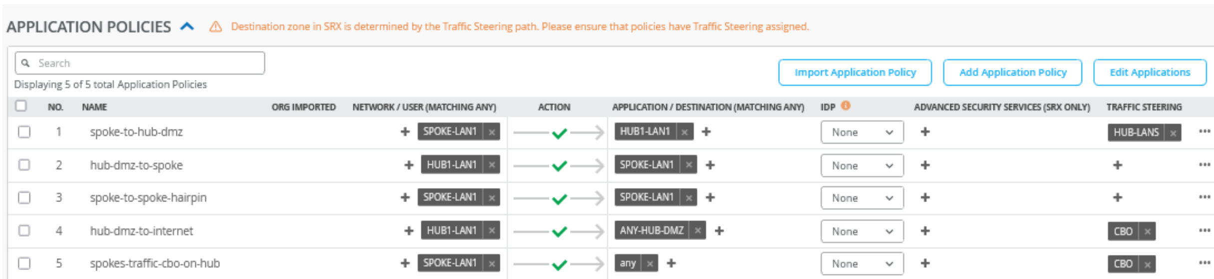
Configure or import the following application policies:

- Number=`1`
  - Name=`spoke-to-hub-dmz`
  - Network=`SPOKE-LAN1`

  - Action=Pass
  - Application=HUB1-LAN1
  - Traffic Steering=HUB-LANS
- Number=2
  - Name= hub-dmz-to-spoke
  - Network=HUB1-LAN1
  - Action=Pass
  - Application=SPOKE-LAN1
  - Traffic Steering=N/A
- Number=3
  - Name= spoke-to-spoke-hairpin
  - Network=SPOKE-LAN1
  - Action=Pass
  - Application=SPOKE-LAN1
  - Traffic Steering=N/A
- Number=4
  - Name=hub-dmz-to-internet
  - Network=HUB1-LAN1
  - Action=Pass
  - Application=ANY-HUB-DMZ
  - Traffic Steering=CBO
- Number=5
  - Name= spokes-traffic-cbo-on-hub
  - Network=SPOKE-LAN1
  - Action=Pass
  - Application=any

- Traffic Steering=`CBO`

The result should look like the figure below:



NOTE: The order of application policies has no impact on Session Smart Router configurations. However, as a best practice, it's recommended to place global rules at the end of the policy rule list. Assigning a traffic steering policy to each application rule is not mandatory for Session Smart Routers. These routers use iBGP-based route distribution to advertise all routes across LAN interfaces automatically. In Session Smart Router deployments, consistent network naming is required for traffic to flow between a hub and a spoke. The network name also functions as a security tenant for traffic isolation, so it must be identical on both sides to ensure proper connectivity.

Save your results.

Create the Hub Profile for the Second Hub

Go to **Organization -> Hub Profiles.**

Should you choose to use the import option, click on **Import Profile** and import the below JSON as a file.

```
{
  "dhcpd_config": {
    "enabled": true
  },
  "ntpOverride": true,
  "dnsOverride": true,
  "service_policies": [
    {
      "name": "spoke-to-hub-dmz",
      "tenants": [
```

```
      "SPOKE-LAN1"
    ],
    "services": [
      "HUB2-LAN1"
    ],
    "action": "allow",
    "path_preference": "HUB-LANS",
    "idp": {
      "enabled": false
    }
  },
  {
    "name": "hub-dmz-to-spoke",
    "tenants": [
      "HUB2-LAN1"
    ],
    "services": [
      "SPOKE-LAN1"
    ],
    "action": "allow",
    "local_routing": true,
    "idp": {
      "enabled": false
    }
  },
  {
    "name": "spoke-to-spoke-hairpin",
    "tenants": [
      "SPOKE-LAN1"
    ],
    "services": [
      "SPOKE-LAN1"
    ],
    "action": "allow",
    "local_routing": true
  },
  {
    "tenants": [
      "HUB2-LAN1"
    ],
    "services": [
      "ANY-HUB-DMZ"
    ],
```

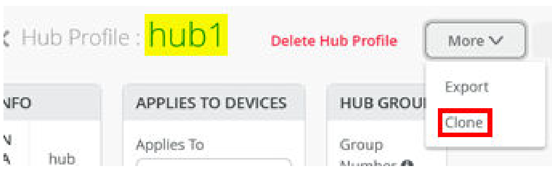
```
      "action": "allow",
      "path_preference": "CBO",
      "name": "hub-dmz-to-internet",
      "idp": {
        "enabled": false
      }
    },
    {
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "any"
      ],
      "action": "allow",
      "name": "spoke-traffic-cbo-on-hub",
      "path_preference": "CBO",
      "idp": {
        "enabled": false
      }
    }
  ],
  "ip_configs": {
    "HUB2-LAN1": {
      "type": "static",
      "ip": "{{HUB2_LAN1_PFX}}.1",
      "netmask": "/24"
    }
  },
  "dns_servers": [
    "8.8.8.8",
    "9.9.9.9"
  ],
  "port_config": {
    "ge-0/0/0": {
      "name": "INET",
      "usage": "wan",
      "wan_type": "broadband",
      "aggregated": false,
      "redundant": false,
      "traffic_shaping": {
        "enabled": false
      },
```

```
      "wan_ext_ip": "{{WAN0_PUBIP}}",
      "ip_config": {
        "type": "static",
        "ip": "{{WAN0_PFX}}.254",
        "netmask": "/24",
        "gateway": "{{WAN0_PFX}}.1"
      },
      "vpn_paths": {
        "hub2-INET.OrgOverlay": {
          "role": "hub"
        }
      }
    },
    "ge-0/0/1": {
      "name": "MPLS",
      "usage": "wan",
      "wan_type": "broadband",
      "aggregated": false,
      "redundant": false,
      "traffic_shaping": {
        "enabled": false
      },
      "wan_ext_ip": "{{WAN1_PUBIP}}",
      "ip_config": {
        "type": "static",
        "ip": "{{WAN1_PFX}}.254",
        "netmask": "/24",
        "gateway": "{{WAN1_PFX}}.1"
      },
      "vpn_paths": {
        "hub2-MPLS.OrgOverlay": {
          "role": "hub"
        }
      }
    },
    "ge-0/0/3": {
      "usage": "lan",
      "aggregated": false,
      "redundant": false,
      "networks": [
        "HUB2-LAN1"
      ]
    }
```

```
  },
  "bgp_config": {},
  "routing_policies": {},
  "extra_routes": {},
  "path_preferences": {
    "HUB-LANS": {
      "strategy": "ordered",
      "paths": [
        {
          "type": "local",
          "networks": [
            "HUB2-LAN1"
          ]
        }
      ]
    },
    "CBO": {
      "strategy": "ordered",
      "paths": [
        {
          "name": "INET",
          "type": "wan"
        }
      ]
    }
  },
  "ospf_areas": {},
  "vrf_instances": {},
  "tunnel_configs": {},
  "oob_ip_config": {
    "type": "dhcp",
    "node1": {
      "type": "dhcp"
    }
  },
  "tunnel_provider_options": {
    "jse": {},
    "zscaler": {}
  },
  "ospf_config": {
    "enabled": false,
    "areas": {}
  },
```

```
    "name": "hub2",
    "type": "gateway"
  }
```

Should you decide to configure everything manually in the Juniper Mist portal, then use the following steps. We've decided to clone the profile from hub1 and change the clone to hub2 for faster results. So, go to "hub1" profile and click on "clone".

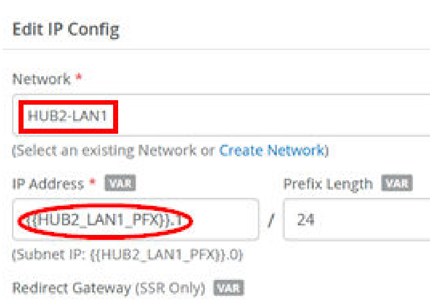


Name the clone "hub2".

After you are on the clone profile, check that the WAN Endpoints have changed as well to `hub2-INET` and `hub2-MPLS` similar to the figure below:

Change the IP address configuration to Hub2.

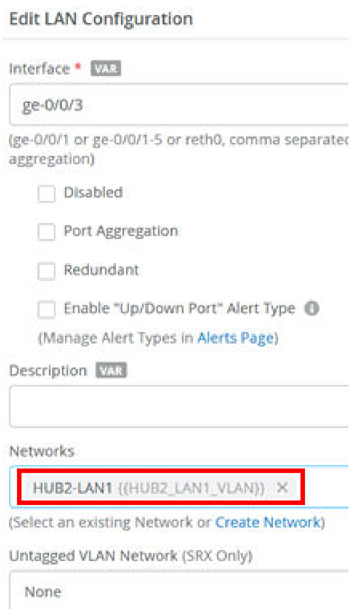
- Network=`HUB2-LAN1`
- IP Address=`{{HUB2_LAN1_PFX}}`



The result should look like the figure below:

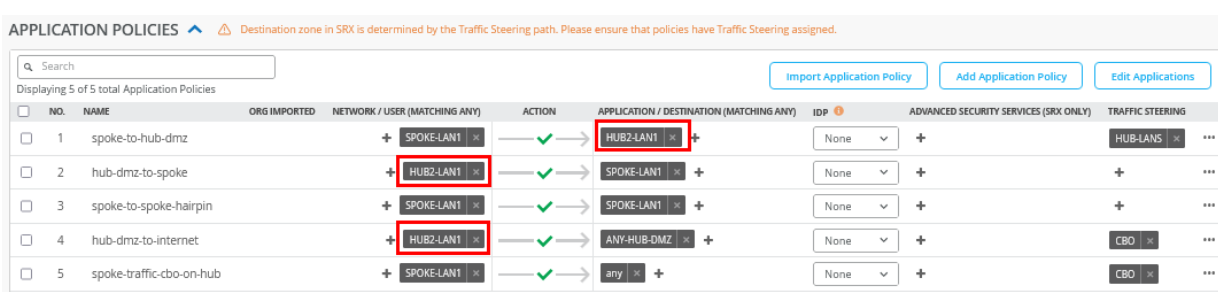
Change the LAN interface network:

- Networks=`HUB2-LAN1`



The result should look like the figure below:

Change the application policies from `HUB1-LAN1` to `HUB2-LAN1` as indicated in the figure below:



Save your results.

Create the WAN Edge Template for Spokes

Go to **Organization -> WAN Edge Templates**.

Here, you have two options:

1. Create the template by importing an existing JSON definition. This is the best way to repeat this example without making errors or forgetting a setting.

2. Create a template and make all the necessary configuration changes in the Juniper Mist portal.

Should you choose to use the import option, click on **Import Profile** and import the below JSON as a file.


```
{
  "type": "spoke",
  "dhcpd_config": {
    "enabled": true,
    "SPOKE-LAN1": {
      "type": "local",
      "ip_start": "{{SPOKE_LAN1_PFX}}.10",
      "ip_end": "{{SPOKE_LAN1_PFX}}.250",
      "gateway": "{{SPOKE_LAN1_PFX}}.1",
      "dns_servers": [
        "8.8.8.8",
        "9.9.9.9"
      ],
      "options": {},
      "lease_time": 86400,
      "fixed_bindings": {}
    }
  },
  "ntpOverride": true,
  "dnsOverride": true,
  "service_policies": [
    {
      "name": "spoke-to-hub-dmz",
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "HUB1-LAN1",
        "HUB2-LAN1"
      ],
```

```
      "action": "allow",
      "idp": {
        "enabled": false
      },
      "path_preference": "VPN"
    },
    {
      "name": "hub-dmz-to-spoke",
      "tenants": [
        "HUB1-LAN1",
        "HUB2-LAN1"
      ],
      "services": [
        "SPOKE-LAN1"
      ],
      "action": "allow",
      "path_preference": "LAN",
      "idp": {
        "enabled": false
      }
    },
    {
      "name": "spoke-to-spoke-via-hub",
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "SPOKE-LAN1"
      ],
      "action": "allow",
      "idp": {
        "enabled": false
      },
      "local_routing": true
    },
    {
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "any"
      ],
      "action": "allow",
```

```
      "name": "internet-via-hub-cbo",
      "idp": {
        "enabled": false
      },
      "path_preference": "VPN"
    }
  ],
  "ip_configs": {
    "SPOKE-LAN1": {
      "type": "static",
      "ip": "{{SPOKE_LAN1_PFX}}.1",
      "netmask": "/24"
    }
  },
  "dns_servers": [
    "8.8.8.8",
    "9.9.9.9"
  ],
  "port_config": {
    "ge-0/0/0": {
      "name": "INET",
      "usage": "wan",
      "wan_type": "broadband",
      "aggregated": false,
      "redundant": false,
      "traffic_shaping": {
        "enabled": false
      },
      "ip_config": {
        "type": "dhcp"
      },
      "vpn_paths": {
        "hub1-INET.OrgOverlay": {
          "bfd_profile": "broadband",
          "role": "spoke"
        },
        "hub2-INET.OrgOverlay": {
          "bfd_profile": "broadband",
          "role": "spoke"
        }
      }
    },
    "ge-0/0/1": {
```

```
      "name": "MPLS",
      "usage": "wan",
      "wan_type": "broadband",
      "aggregated": false,
      "redundant": false,
      "traffic_shaping": {
        "enabled": false
      },
      "ip_config": {
        "type": "static",
        "ip": "{{WAN1_PFX}}.2",
        "netmask": "/24",
        "gateway": "{{WAN1_PFX}}.1"
      },
      "vpn_paths": {
        "hub1-MPLS.OrgOverlay": {
          "bfd_profile": "broadband",
          "role": "spoke"
        },
        "hub2-MPLS.OrgOverlay": {
          "bfd_profile": "broadband",
          "role": "spoke"
        }
      }
    },
    "ge-0/0/3": {
      "usage": "lan",
      "networks": [
        "SPOKE-LAN1"
      ]
    }
  },
  "bgp_config": {},
  "routing_policies": {},
  "extra_routes": {},
  "path_preferences": {
    "VPN": {
      "strategy": "weighted",
      "paths": [
        {
          "name": "hub1-INET.OrgOverlay",
          "cost": 10,
          "type": "vpn"
```

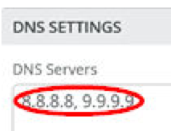
```
      },
      {
        "name": "hub2-INET.OrgOverlay",
        "cost": 20,
        "type": "vpn"
      },
      {
        "name": "hub1-MPLS.OrgOverlay",
        "cost": 30,
        "type": "vpn"
      },
      {
        "name": "hub2-MPLS.OrgOverlay",
        "cost": 40,
        "type": "vpn"
      }
    ]
  },
  "LAN": {
    "strategy": "ordered",
    "paths": [
      {
        "type": "local",
        "networks": [
          "SPOKE-LAN1"
        ]
      }
    ]
  }
},
"ospf_areas": {},
"vrf_instances": {},
"tunnel_configs": {},
"oob_ip_config": {
  "type": "dhcp",
  "node1": {
    "type": "dhcp"
  }
},
"tunnel_provider_options": {
  "jse": {},
  "zscaler": {}
},
```

```
  "ospf_config": {
    "enabled": false,
    "areas": {}
  },
  "name": "Spokes"
}
```


Should you decide to configure everything manually in the Juniper Mist portal, then use the following steps.

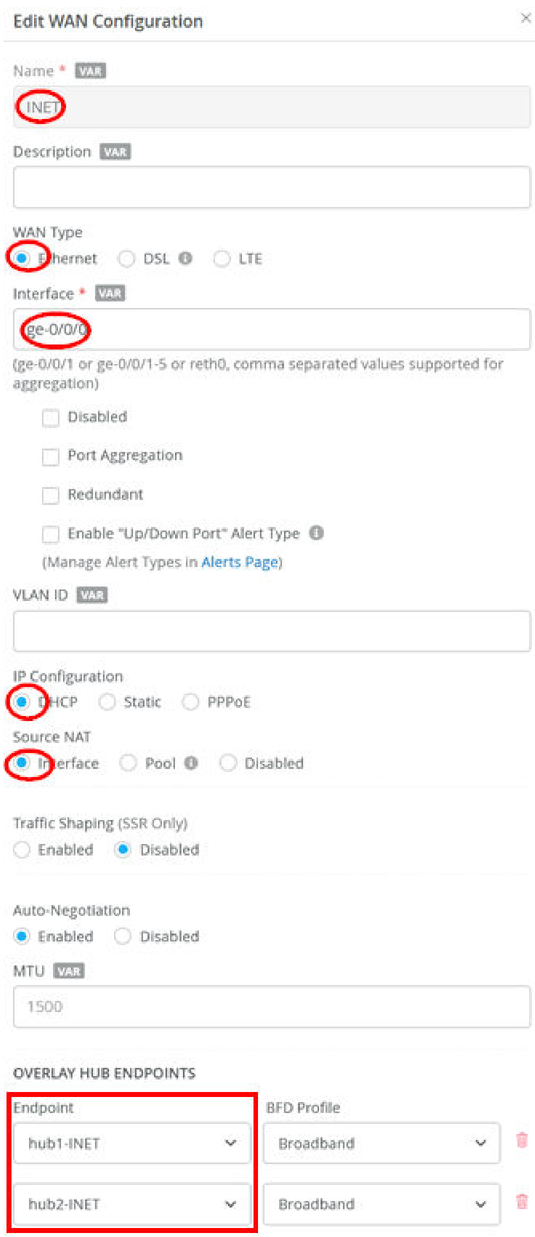
Edit the **DNS** Settings

- DNS Servers=`8.8.8.8, 9.9.9.9`



Configure a first **WAN interface** as follows

- Name=`INET` this indicates which topology it's going to use.
- WAN Type=`Ethernet`
- Interface=`ge-0/0/0`
- IP Configuration=`DHCP`
- Source NAT=`Interface`
- Overlay Hub Endpoints
  - Endpoint1=`hub1-INET`
  - BFD Profile1=`Broadband`
  - Endpoint2=`hub2-INET`
  - BFD Profile2=`Broadband`

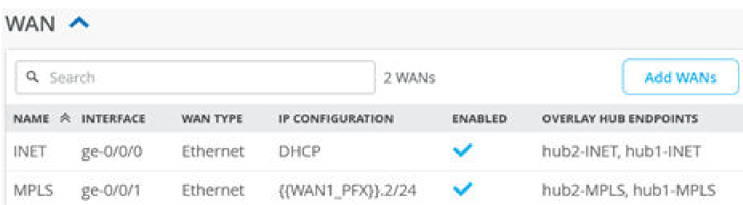


Configure the second WAN interface as follows:

- Name=`MPLS` this indicates which topology it's going to use.
- WAN Type=`Ethernet`
- Interface=`ge-0/0/1`
- IP Configuration=`Static`
- `IP Address={{WAN1_PFX}}.2`
- `Prefix Length=24`

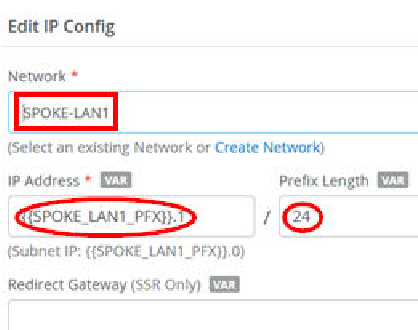
- Gateway=`{{WAN1_PFX}}.1`
- Source NAT=`Interface`
- Overlay Hub Endpoints
    - Endpoint1=`hub1-MPLS`
    - BFD Profile1=`Broadband`
    - Endpoint2=`hub2-MPLS`
    - BFD Profile2=`Broadband`

The result should look like the figure below:

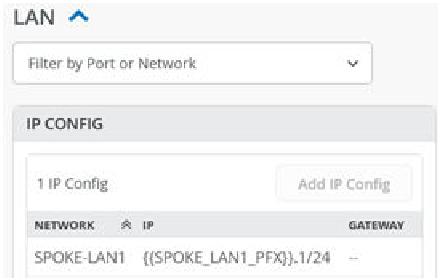


Add a LAN IP config now with the following configuration:

- Network=`SPOKE-LAN1`
- IP Address=`{{SPOKE_LAN1_PFX}}.1`
- Prefix Length=`24`

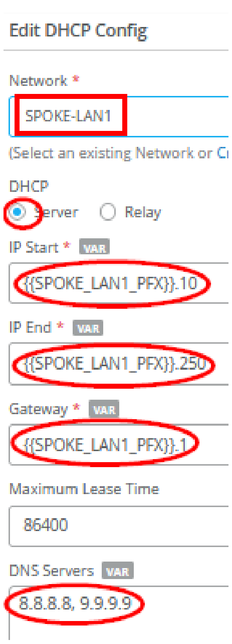


The result should look like the figure below:

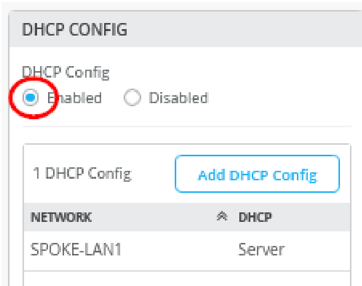


Add a DHCP server configuration like the one below:

- Network=`SPOKE-LAN1`
- DHCP=`Server`
- IP Start=`{{SPOKE_LAN1_PFX}}.10`
- IP End=`{{SPOKE_LAN1_PFX}}.250`
- Gateway=`{{SPOKE_LAN1_PFX}}.1`
- Maximum Lease Time=`86400`
- DNS Servers=`8.8.8.8, 9.9.9.9`

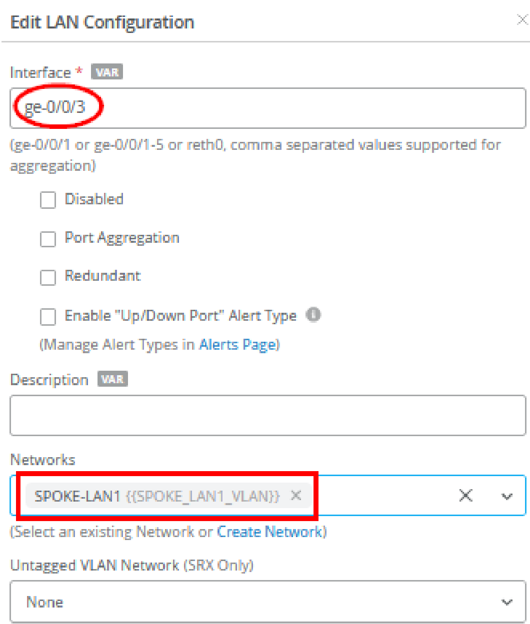


The result should look like the figure below:

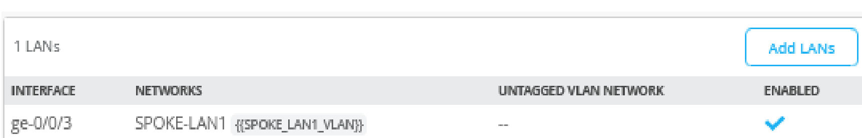


Add a LAN interface now with the following configuration:

- Interface=`ge-0/0/3`
- Networks=`SPOKE1-LAN1`
- Untagged VLAN=`None`



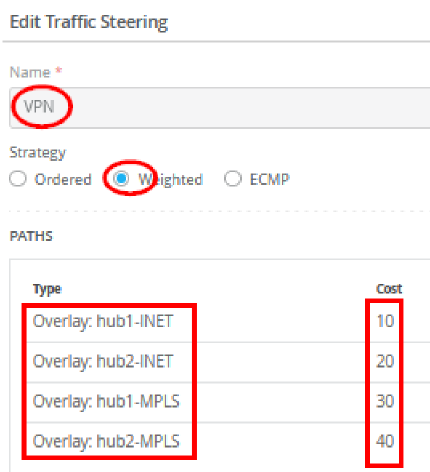
The result should look like the figure below:



Now we need to define two traffic steering rules. The first rule has the following configuration:

- Name=`VPN`
- Strategy=`Weighted`

- Paths
- Path1 Type=`Overlay: hub1-INET`
  - Path1 Cost=`10`
  - Path2 Type=`Overlay: hub2-INET`
  - Path2 Cost=`20`
  - Path3 Type=`Overlay: hub1-MPLS`
  - Path3 Cost=`30`
  - Path4 Type=`Overlay: hub2-MPLS`
  - Path4 Cost=`40`



NOTE: In typical scenarios with two different hubs, assigned weights ensure that all "any" (0.0.0.0/0) traffic destined for central Internet breakout is routed through only one active hub at a time. Avoid using Equal Cost Multi-Path (ECMP) in this setup due to the source NAT being performed at each hub for Internet-bound traffic. For consistent behavior, traffic should originate from the same public IP address to maintain application session integrity. If traffic is load-balanced across hubs, applications on the internet may observe different source IPs for each flow, potentially causing issues.

The second rule has the following configuration:

- Name=`LAN`
- Strategy=`Ordered`

- Paths
  - Path1 Type=`LAN: SPOKE-LAN1`

The result should look like the figure below:

TRAFFIC STEERING

Search 2 Traffic Steering

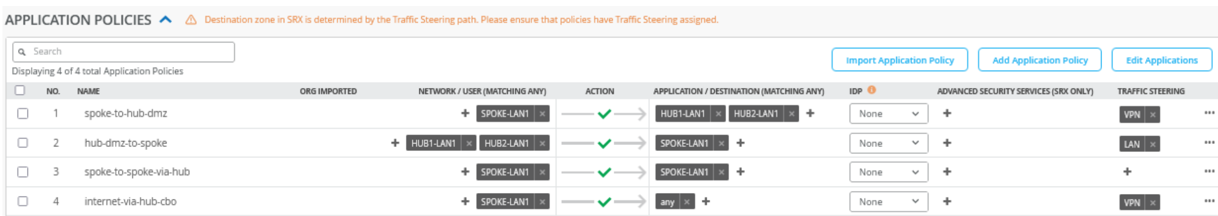
NAME	STRATEGY	PATHS
LAN	Ordered	SPOKE-LAN1
VPN	Weighted	hub1-INET[10], hub2-INET[20], hub1-MPLS[30], hub2-MPLS[40]

Configure or import the following Application Policies

- Number=`1`
  - Name=`spoke-to-hub-dmz`
  - Network=`SPOKE-LAN1`
  - Action=`Pass`
  - Application=`HUB1-LAN1 + HUB2-LAN1`
  - Traffic Steering=`VPN`
- Number=`2`
  - Name= `hub-dmz-to-spoke`
  - Network=`HUB1-LAN1 + HUB2-LAN1`
  - Action=`Pass`
  - Application=`SPOKE-LAN1`
  - Traffic Steering=`LAN`
- Number=`3`
  - Name=`spoke-to-spoke-via-hub`
  - Network=`SPOKE-LAN1`
  - Action=`Pass`
  - Application=`SPOKE-LAN1`
  - Traffic Steering=`N/A`

- Number=`4`
  - Name= `internet-via-hub-cbo`
  - Network=`SPOKE-LAN1`
  - Action=`Pass`
  - Application=`any`
  - Traffic Steering=`VPN`

The result should look like the figure below:

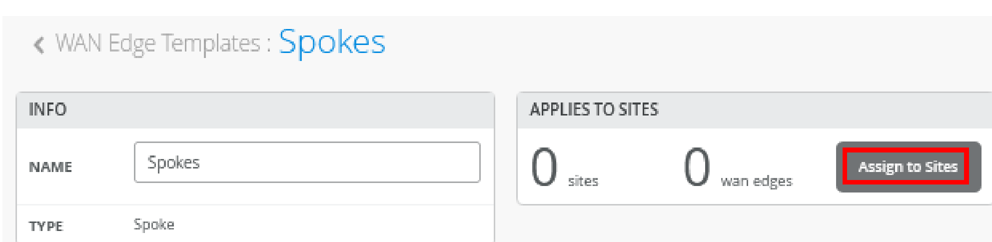


NOTE: The order of application policies has no impact on Session Smart Router configurations. However, as a best practice, it's recommended to place global rules at the end of the policy rule list. Assigning a traffic steering policy to each application rule is not mandatory for Session Smart Routers. These routers use iBGP-based route distribution to advertise all routes across LAN interfaces automatically. In Session Smart Router deployments, consistent network naming is required for traffic to flow between a hub and a spoke. The network name also functions as a security tenant for traffic isolation, so it must be identical on both sides to ensure proper connectivity.

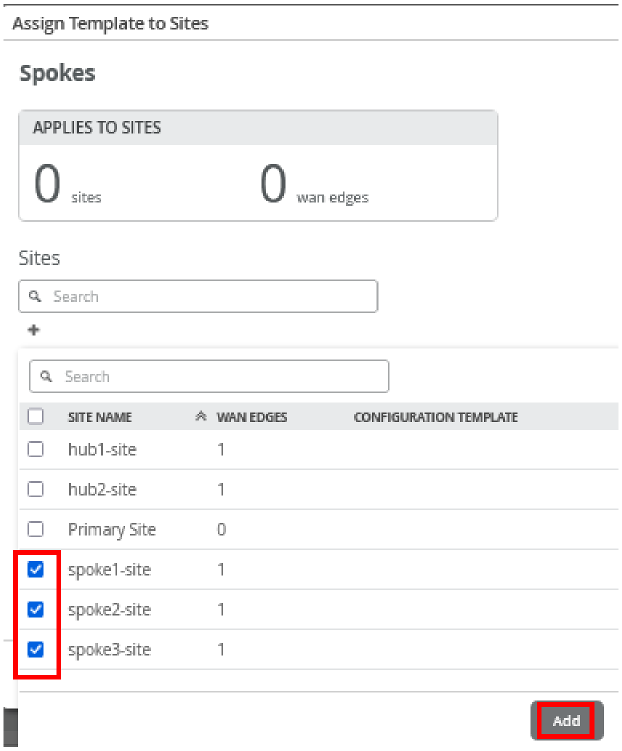
Save your results.

Assigning Spoke Templates to Sites

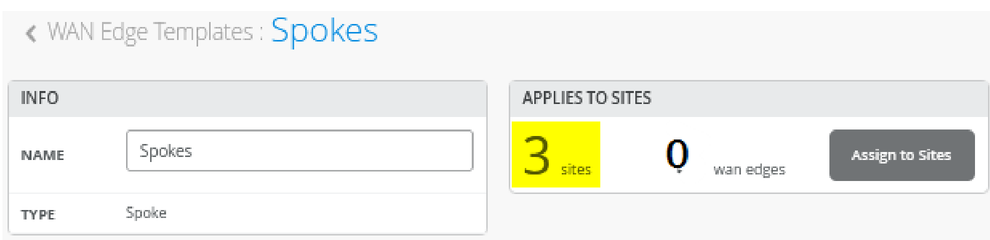
Go to **Organization -> WAN Edge Templates** and create a spoke template and click on **Assign to Sites**.



Then select only the three “spokeX-site” and “Apply”.



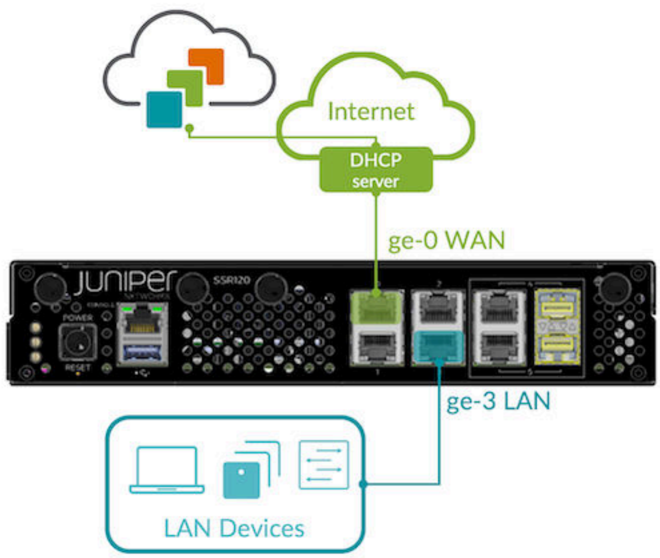
The result should indicate three sites (the WAN edges change when devices get assigned to these).



Onboard your Devices

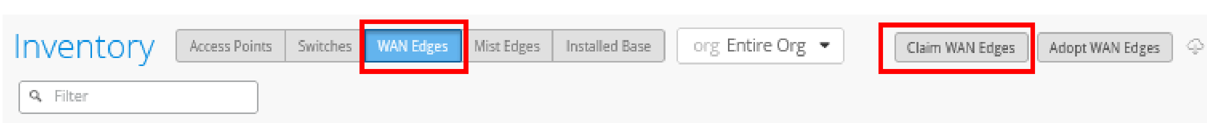
Now it’s time to use the **Claim** or **Adopt** method to onboard the devices and see them in the organization inventory. Multiple onboarding methods are supported, but the default claim and ZTP method is described here, as it is the simplest and most straightforward approach.

Connect Your Device to the Cloud. Your SSR device uses port 0 (`ge-0/0/0` in the Juniper Mist portal) as a default WAN port to contact Juniper Mist for ZTP. This interface must be able to obtain a DHCP lease to access the Internet and communicate with the Juniper Mist cloud. A static IP configuration can be applied later through a template or hub profile. Connecting the LAN interface during onboarding is only necessary when using the Adopt method.

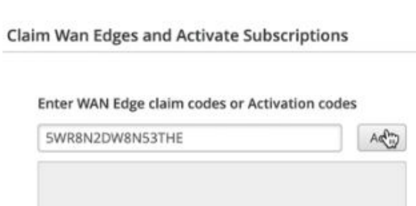


Obtain Claim code from device. On the back of the device there are two stickers with codes. It's best that you take a photo of these for later. The left sticker has the claim code to be used on the Juniper Mist portal.

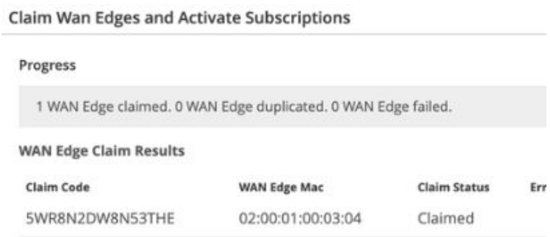
Mist Claim Code Entry. You can use the Mist mobile application to scan the QR code directly or use it on the Juniper Mist portal. Go to **Organization -> Inventory**, select **WAN Edges** and click on **Claim WAN Edges** as shown in the figure below:



Add the device claim code into the list of devices to claim.

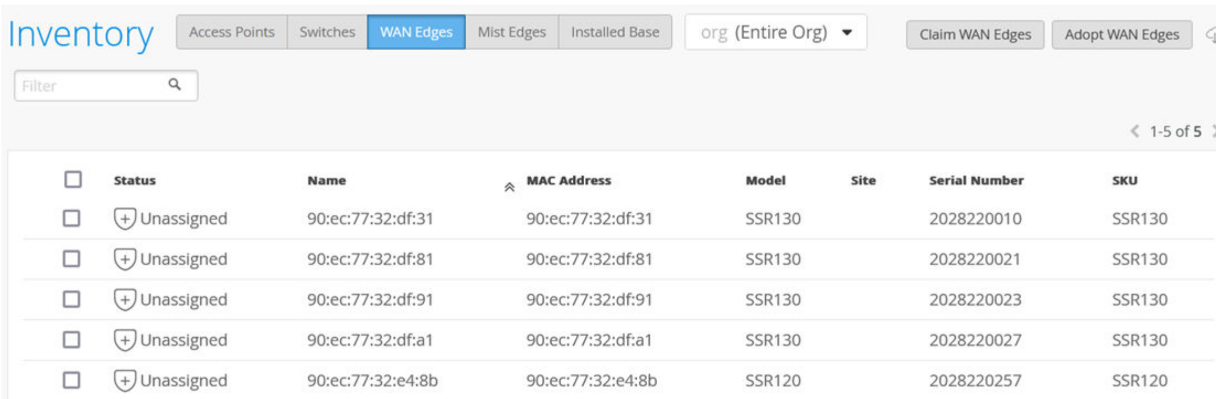


Click the **Claim** button to claim the device into your inventory.



NOTE: The MistAI app can be downloaded from mobile app stores a.) for Apple Devices and b.) for Android Devices

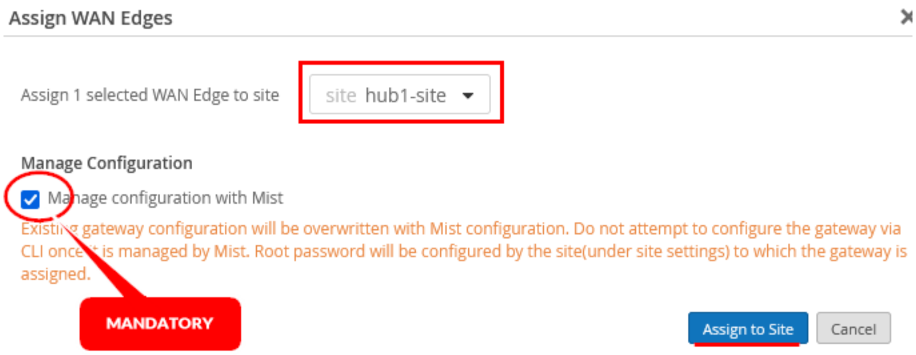
In the example below, we just claimed five devices for a lab without directly assigning them to a site. This is similar to using the adopt method.



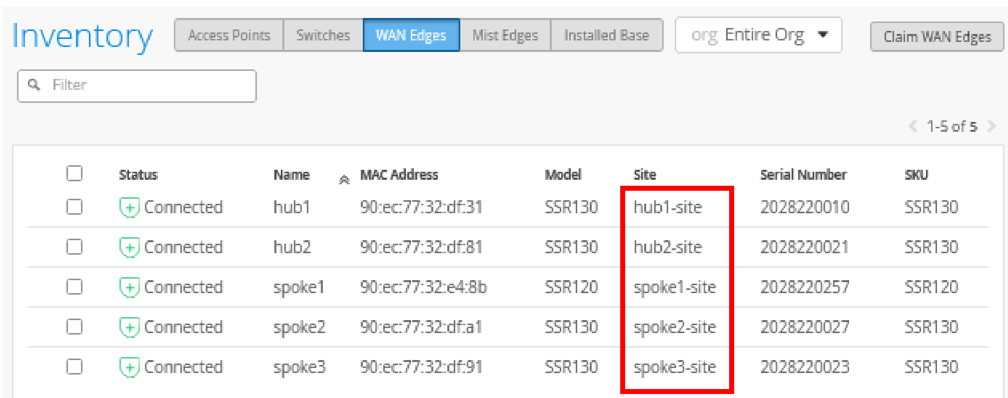
Assigning Devices to Sites

Each SSR or SRX must now be assigned one-by-one to an individual site using **Assign to Site**.

Select the site for each device and make sure to enable **Manage configuration with Mist**. The default option of not enabling device management is a better practice for SRX Firewalls.



Now assign all five devices to their individual sites until you see the below:

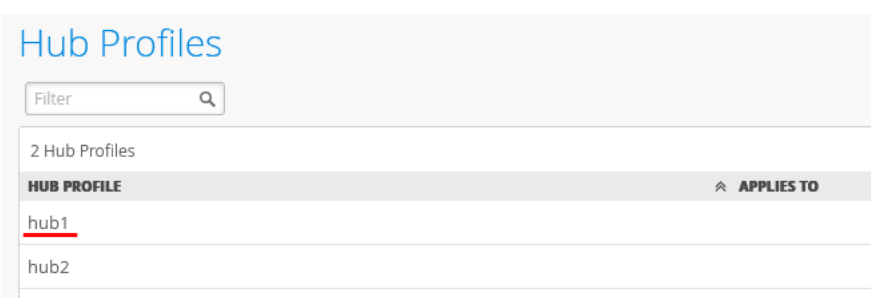


Assign Hub Profiles to Devices

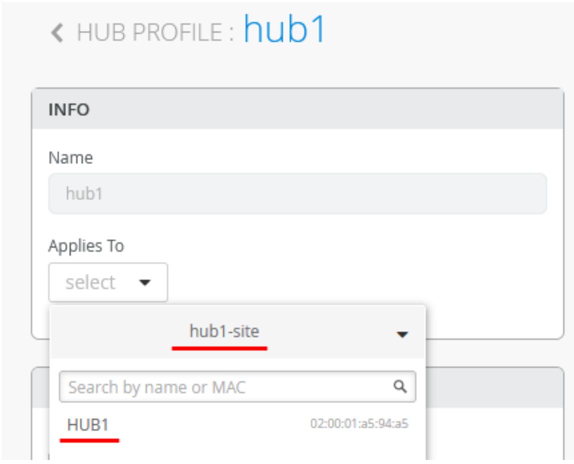
The spoke sites will automatically receive their configurations, as the templates have already been assigned. For the hub sites, however, the next step is to manually assign the appropriate hub profile.

Go to **Organization -> Hub Profiles.**

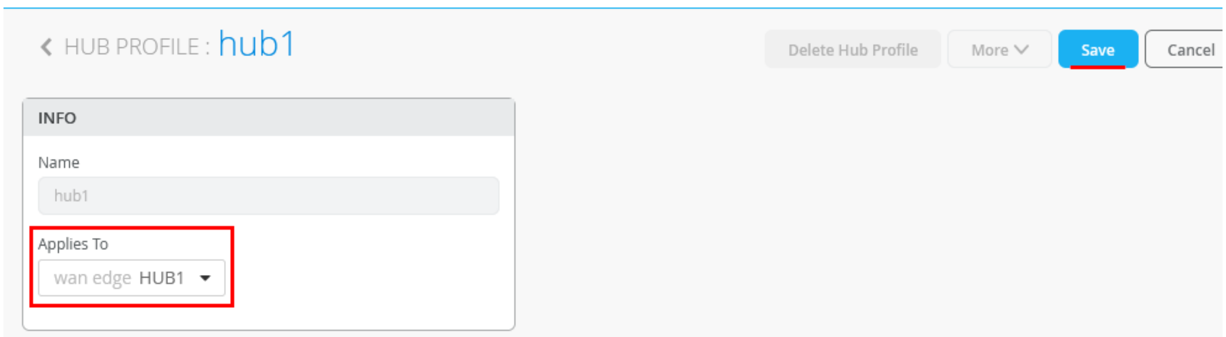
Click on the first hub profile.



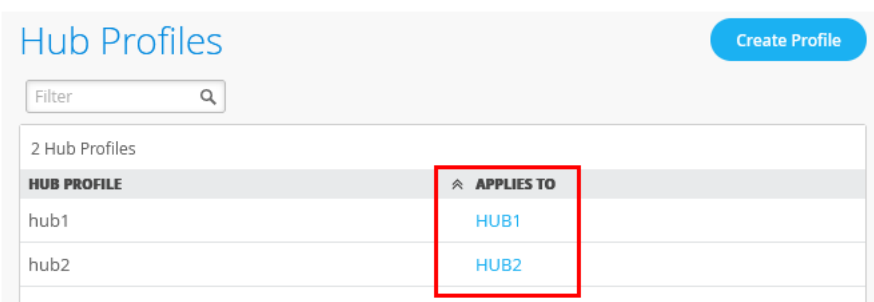
Under **Applies To** select “hub1-site” and “HUB1”.



Click on **Save**.



Repeat the process for the second hub in the second hub site so that in the end both hub profiles have their individual hub device assigned as shown below:

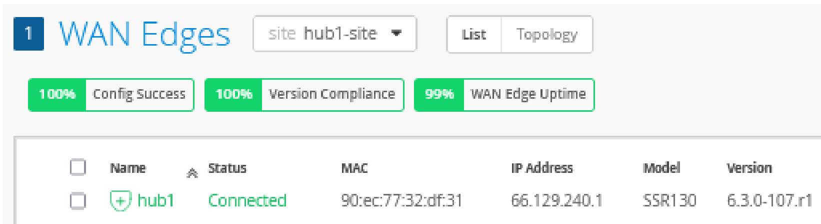


NOTE: Wait about 10 minutes until the initial configuration is brought up for the first time and all changes are made and applied.

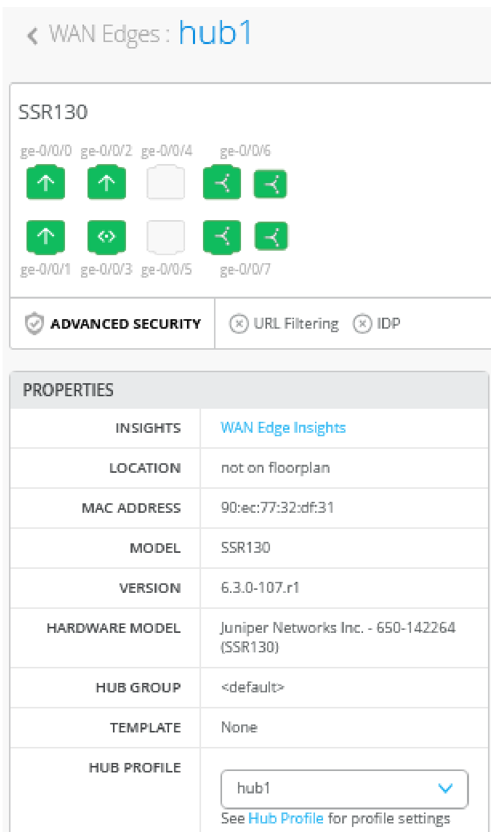
Test Your Network Configuration

We are now ready to test our configuration.

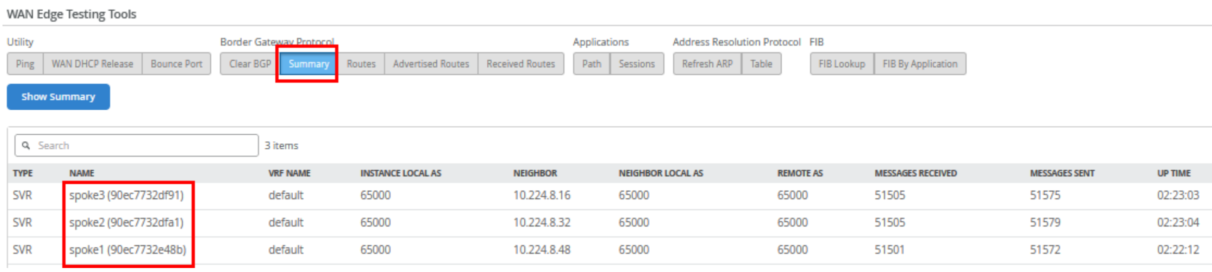
Go to **WAN Edges -> site=hub1-site** and click on “hub1”.



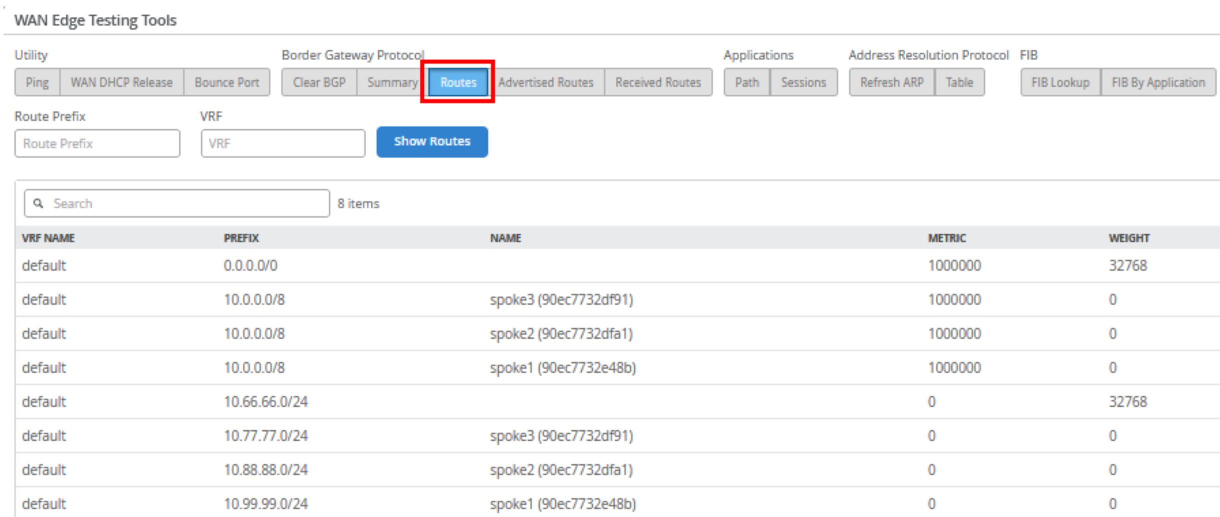
Review the device information.



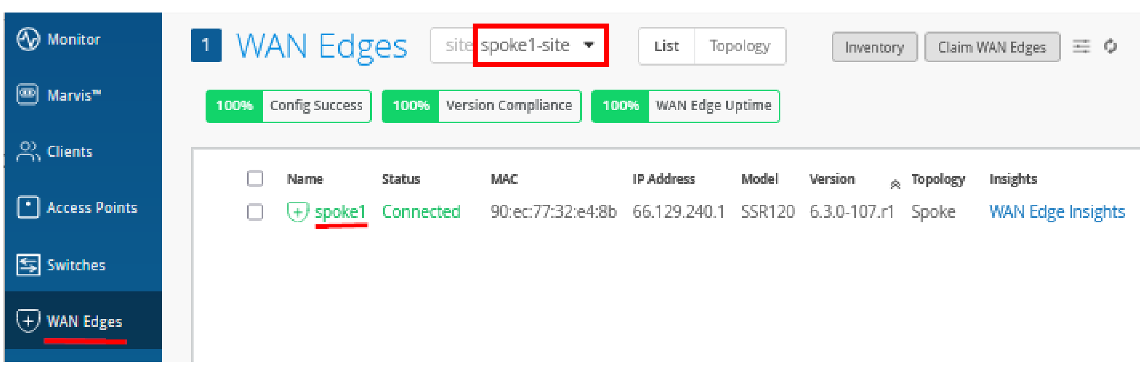
When you use **Utilities -> Testing Tools** and review the BGP neighbor summary, you will see only the three spokes connected and exchanging routes.



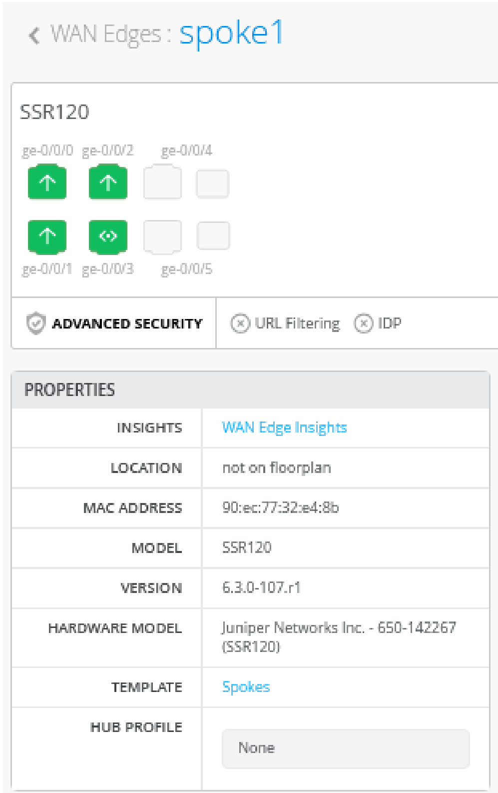
Also review the routes distributed in the VPN.



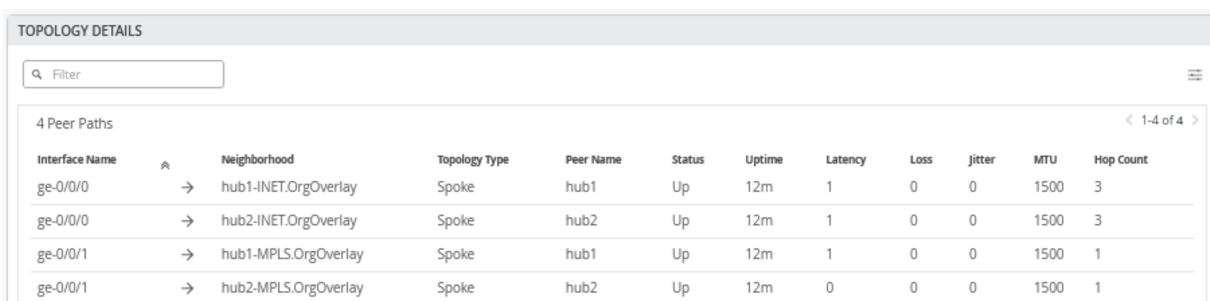
Go to **WAN Edges -> site=spoke1-site** and click on “spoke1”.



Review the device information.



Review the topology details with the four tunnels this spoke has established to the two hubs.



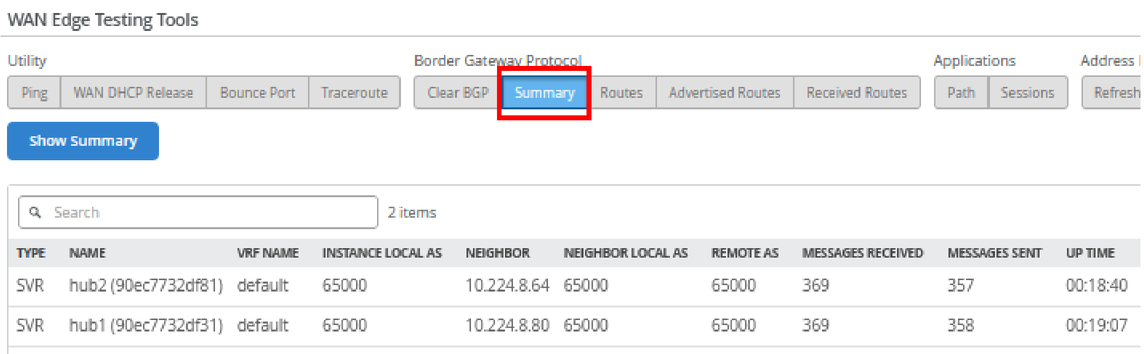
TOPOLOGY DETAILS

4 Peer Paths — 1-4 of 4

Interface Name		Neighborhood	Topology Type	Peer Name	Status	Uptime	Latency	Loss	Jitter	MTU	Hop Count
ge-0/0/0	→	hub1-INET.OrgOverlay	Spoke	hub1	Up	12m	1	0	0	1500	3
ge-0/0/0	→	hub2-INET.OrgOverlay	Spoke	hub2	Up	12m	1	0	0	1500	3
ge-0/0/1	→	hub1-MPLS.OrgOverlay	Spoke	hub1	Up	12m	1	0	0	1500	1
ge-0/0/1	→	hub2-MPLS.OrgOverlay	Spoke	hub2	Up	12m	0	0	0	1500	1

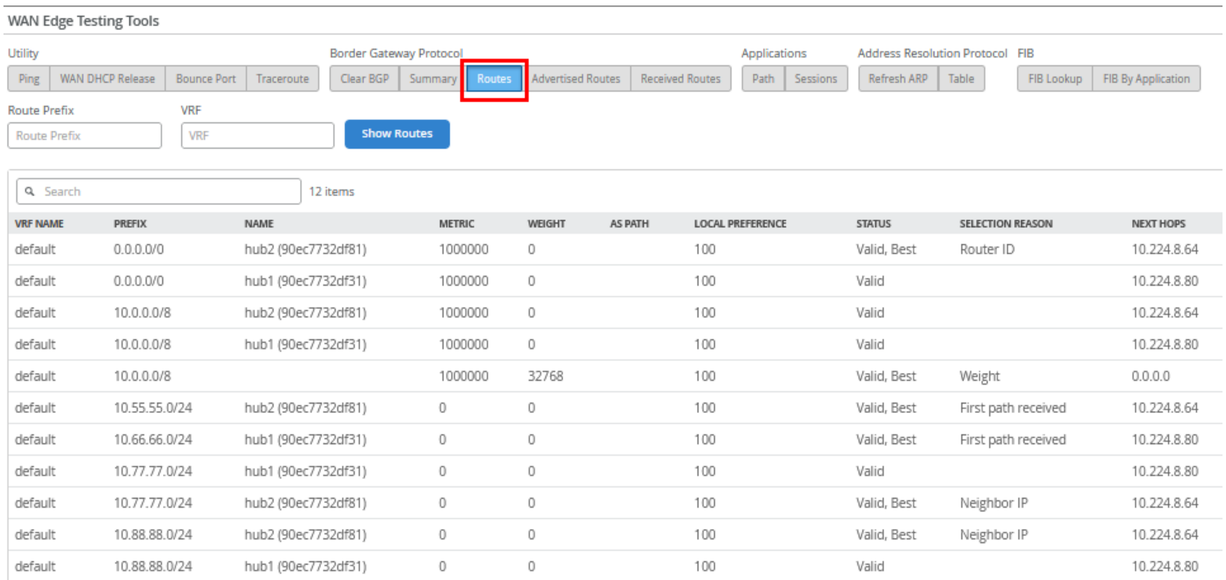
NOTE: On the hubs, only tunnels to other hubs are displayed for scale reasons. You will see that in the next topology.

When you use **Utilities -> Testing Tools** and review the BGP neighbor summary, you will see only the two hubs are connected and exchanging routes.



TYPE	NAME	VRF NAME	INSTANCE LOCAL AS	NEIGHBOR	NEIGHBOR LOCAL AS	REMOTE AS	MESSAGES RECEIVED	MESSAGES SENT	UP TIME
SVR	hub2 (90ec7732df81)	default	65000	10.224.8.64	65000	65000	369	357	00:18:40
SVR	hub1 (90ec7732df31)	default	65000	10.224.8.80	65000	65000	369	358	00:19:07

Also review the routes distributed in the VPN.



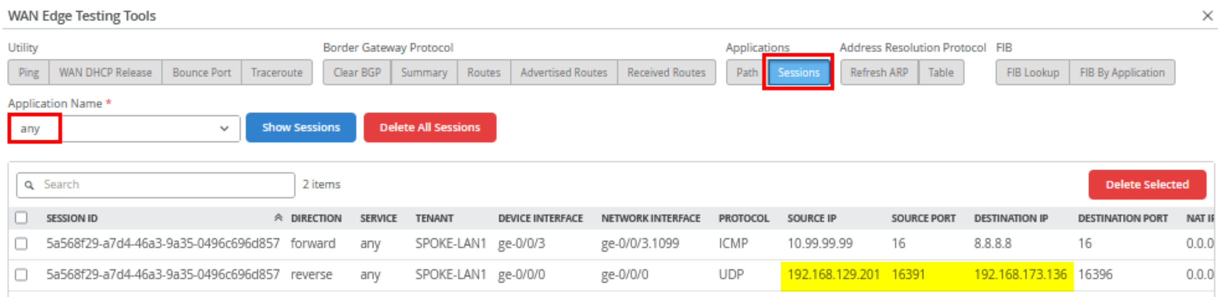
VRF NAME	PREFIX	NAME	METRIC	WEIGHT	AS PATH	LOCAL PREFERENCE	STATUS	SELECTION REASON	NEXT HOPS
default	0.0.0.0/0	hub2 (90ec7732df81)	1000000	0		100	Valid, Best	Router ID	10.224.8.64
default	0.0.0.0/0	hub1 (90ec7732df31)	1000000	0		100	Valid		10.224.8.80
default	10.0.0.0/8	hub2 (90ec7732df81)	1000000	0		100	Valid		10.224.8.64
default	10.0.0.0/8	hub1 (90ec7732df31)	1000000	0		100	Valid		10.224.8.80
default	10.0.0.0/8		1000000	32768		100	Valid, Best	Weight	0.0.0.0
default	10.55.55.0/24	hub2 (90ec7732df81)	0	0		100	Valid, Best	First path received	10.224.8.64
default	10.66.66.0/24	hub1 (90ec7732df31)	0	0		100	Valid, Best	First path received	10.224.8.80
default	10.77.77.0/24	hub1 (90ec7732df31)	0	0		100	Valid		10.224.8.80
default	10.77.77.0/24	hub2 (90ec7732df81)	0	0		100	Valid, Best	Neighbor IP	10.224.8.64
default	10.88.88.0/24	hub2 (90ec7732df81)	0	0		100	Valid, Best	Neighbor IP	10.224.8.64
default	10.88.88.0/24	hub1 (90ec7732df31)	0	0		100	Valid		10.224.8.80

We shall now continue our testing on the clients attached to the spokes. We attach to the desktop1 VM with IP address `10.99.99.99` attached to spoke1:

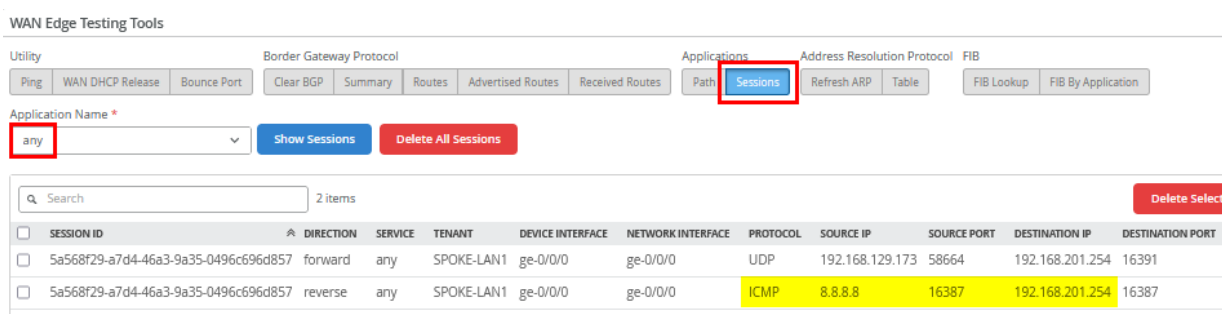
```
# try to reach the local WAN-Router interface desktop1 VM is attached to
root@desktop1:~# ping -c3 10.99.99.1
PING 10.99.99.1 (10.99.99.1) 56(84) bytes of data.
64 bytes from 10.99.99.1: icmp_seq=1 ttl=128 time=0.457 ms
64 bytes from 10.99.99.1: icmp_seq=2 ttl=128 time=0.329 ms
64 bytes from 10.99.99.1: icmp_seq=3 ttl=128 time=0.948 ms
#
# try to reach the client desktop2 VM attached to spoke2
# this causes relay on the hub for this traffic
root@desktop1:~# ping -c3 10.88.88.88
PING 10.88.88.88 (10.88.88.88) 56(84) bytes of data.
64 bytes from 10.88.88.88: icmp_seq=1 ttl=120 time=4.16 ms
```

```
64 bytes from 10.88.88.88: icmp_seq=2 ttl=120 time=1.32 ms
64 bytes from 10.88.88.88: icmp_seq=3 ttl=120 time=1.24 ms
#
# try to reach the client desktop3 VM attached to spoke3
# this causes relay on the hub for this traffic
root@desktop1:~# ping -c3 10.77.77.77
PING 10.77.77.77 (10.77.77.77) 56(84) bytes of data.
64 bytes from 10.77.77.77: icmp_seq=1 ttl=122 time=12.4 ms
64 bytes from 10.77.77.77: icmp_seq=2 ttl=122 time=1.28 ms
64 bytes from 10.77.77.77: icmp_seq=3 ttl=122 time=1.25 ms
#
# try to reach the client desktop4 VM attached to hub1
root@desktop1:~# ping -c3 10.66.66.66
PING 10.66.66.66 (10.66.66.66) 56(84) bytes of data.
64 bytes from 10.66.66.66: icmp_seq=1 ttl=59 time=4.54 ms
64 bytes from 10.66.66.66: icmp_seq=2 ttl=59 time=1.13 ms
64 bytes from 10.66.66.66: icmp_seq=3 ttl=59 time=1.13 ms
#
# try to reach the client desktop5 VM attached to hub2
root@desktop1:~# ping -c3 10.55.55.55
PING 10.55.55.55 (10.55.55.55) 56(84) bytes of data.
64 bytes from 10.55.55.55: icmp_seq=1 ttl=59 time=4.29 ms
64 bytes from 10.55.55.55: icmp_seq=2 ttl=59 time=1.14 ms
64 bytes from 10.55.55.55: icmp_seq=3 ttl=59 time=0.968 ms
#
# let a continued ping to the internet run
# in our case all traffic is sent to hub for central breakout
root@desktop1:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=47 time=8.43 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=47 time=3.83 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=47 time=3.84 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=47 time=3.98 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=47 time=3.82 ms
.
.
```

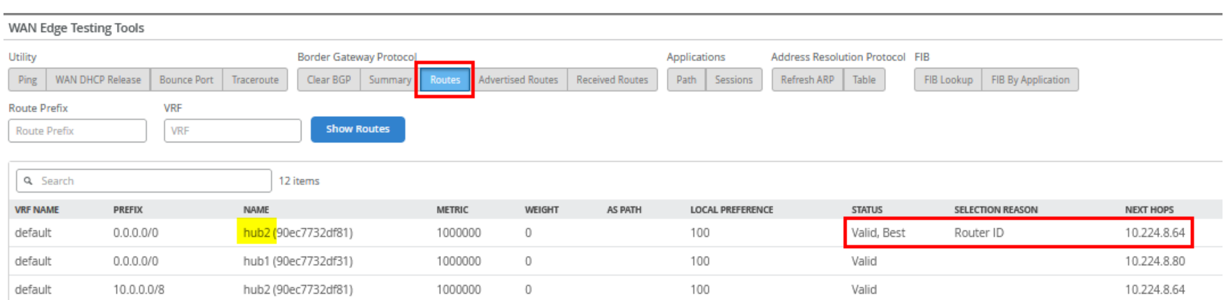
Use **Utilities -> Testing Tools** to review the application sessions with Application Name=`any`. Due to the reverse flow, we see that the traffic is received from Hub2's Internet public IP address `192.168.129.201`.



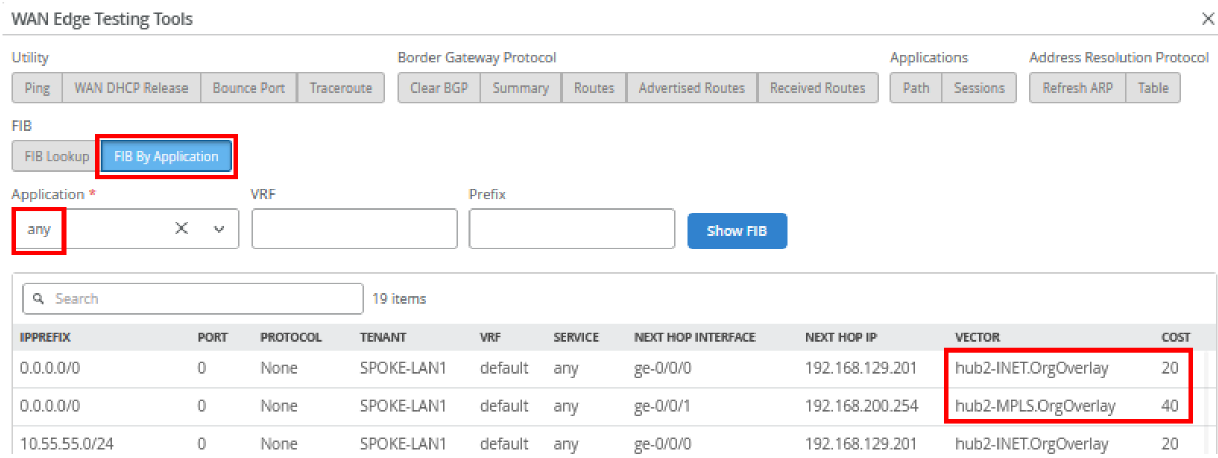
Do the same for the second hub by going to **WAN Edges -> site=hub2-site** and clicking on "hub2". Then go to **Utilities -> Testing Tools** and review the application sessions with Application Name=`any` again. Here, you can see the reverse flow ICMP responses to the source NATed Interface `ge-0/0/0` where we forwarded our traffic to.



If you're wondering why traffic wasn't routed to Hub1, check the FIB routes on Spoke1. Hub2 is preferred because, although both hubs advertised the same default route (0.0.0.0/0), Hub2 had a lower internal Router ID (or BGPoSVR loopback IP) than Hub1, making it the preferred path.



You can also verify this through the FIB created for the Application=`any` as below:



In case you do not want Hub2 as the default router, follow the instructions in "Changing the Hub Used for Central Breakout When Traffic Destination Is “Any”" on page 197 .

The remaining testing is done with the clients attached to the hubs. We connect to the desktop4 VM with IP address `10.66.66.66` attached to Hub1:

```
# try to reach the client desktop1 VM attached to spoke1
root@desktop4:~# ping -c3 10.99.99.99
PING 10.99.99.99 (10.99.99.99) 56(84) bytes of data.
64 bytes from 10.99.99.99: icmp_seq=1 ttl=59 time=4.98 ms
64 bytes from 10.99.99.99: icmp_seq=2 ttl=59 time=1.07 ms
64 bytes from 10.99.99.99: icmp_seq=3 ttl=59 time=1.03 ms
#
# try to reach the client desktop2 VM attached to spoke2
root@desktop4:~# ping -c3 10.88.88.88
PING 10.88.88.88 (10.88.88.88) 56(84) bytes of data.
64 bytes from 10.88.88.88: icmp_seq=1 ttl=59 time=5.49 ms
64 bytes from 10.88.88.88: icmp_seq=2 ttl=59 time=1.15 ms
64 bytes from 10.88.88.88: icmp_seq=3 ttl=59 time=1.06 ms
#
# try to reach the client desktop3 VM attached to spoke3
root@desktop4:~# ping -c3 10.77.77.77
PING 10.77.77.77 (10.77.77.77) 56(84) bytes of data.
64 bytes from 10.77.77.77: icmp_seq=1 ttl=59 time=7.07 ms
64 bytes from 10.77.77.77: icmp_seq=2 ttl=59 time=1.21 ms
64 bytes from 10.77.77.77: icmp_seq=3 ttl=59 time=1.04 ms
#
# try services on the internet using the local breakout on the hub
root@desktop4:~# ping -c3 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
```

```
64 bytes from 8.8.8.8: icmp_seq=1 ttl=109 time=3.10 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=109 time=2.67 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=109 time=2.68 ms
#
# try to reach the client desktop5 VM attached to hub2
# It's expected NOT to work as hub to hub traffic will be done in the next topology
root@desktop4:~# ping -c3 10.55.55.55
PING 10.55.55.55 (10.55.55.55) 56(84) bytes of data.
.
--- 10.55.55.55 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2043ms
```

For the last test, we connect to the desktop5 VM with IP address `10.55.55.55` attached to Hub2:

```
# try to reach the client desktop1 VM attached to spoke1
root@desktop5:~# ping -c3 10.99.99.99
PING 10.99.99.99 (10.99.99.99) 56(84) bytes of data.
64 bytes from 10.99.99.99: icmp_seq=1 ttl=61 time=4.21 ms
64 bytes from 10.99.99.99: icmp_seq=2 ttl=61 time=1.01 ms
64 bytes from 10.99.99.99: icmp_seq=3 ttl=61 time=1.02 ms
#
# try to reach the client desktop2 VM attached to spoke2
root@desktop5:~# ping -c3 10.88.88.88
PING 10.88.88.88 (10.88.88.88) 56(84) bytes of data.
64 bytes from 10.88.88.88: icmp_seq=1 ttl=59 time=4.95 ms
64 bytes from 10.88.88.88: icmp_seq=2 ttl=59 time=1.14 ms
64 bytes from 10.88.88.88: icmp_seq=3 ttl=59 time=1.12 ms
#
# try to reach the client desktop3 VM attached to spoke3
root@desktop5:~# ping -c3 10.77.77.77
PING 10.77.77.77 (10.77.77.77) 56(84) bytes of data.
64 bytes from 10.77.77.77: icmp_seq=1 ttl=59 time=4.83 ms
64 bytes from 10.77.77.77: icmp_seq=2 ttl=59 time=1.22 ms
64 bytes from 10.77.77.77: icmp_seq=3 ttl=59 time=1.32 ms
#
# try services on the internet using the local breakout on the hub
root@desktop5:~# ping -c3 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=51 time=3.75 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=51 time=3.48 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=51 time=3.35 ms
#
```

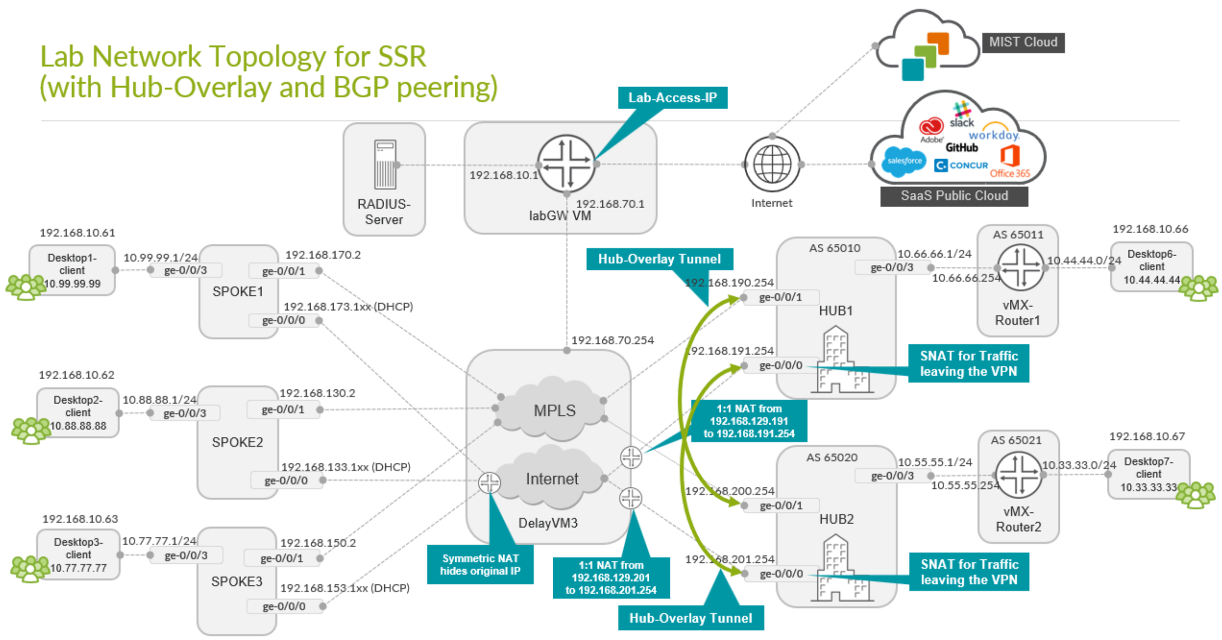
```
# try to reach the client desktop4 VM attached to hub1
# It's expected NOT to work as hub to hub traffic will be done in the next topology
root@desktop5:~# ping -c3 10.44.44.44
PING 10.44.44.44 (10.44.44.44) 56(84) bytes of data.
.
--- 10.44.44.44 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2027ms
```

Appendix: Building an Extended Topology with Hub Overlay and BGP Peering

IN THIS SECTION

This lab is an extension of the previous lab "Appendix: Building a base SD-WAN Topology with Three Spokes and Two Hubs" on page 23. The underlay connections and spoke implementation are not changed. We add two new changes to this lab:

- We build a hub overlay. This enables sending traffic between the two hubs directly by exiting the WAN infrastructure. Hence you enable DC to DC traffic using this topology.
  - For traffic steering, we utilize an ECMP-based load-balancing algorithm to have the flows distributed among the two paths between the hubs.
- We introduce data center routers that are attached to the hub LAN interfaces. Those will manage additional resources like servers attached to other interfaces. The additional IP prefixes for those resources will get announced through exterior BGP and propagated through the VPN.



The following table has the additional device information for the new topology.

Location	Direct Hub IF	Local AS	Router IF	Router AS	Route propagation	DC Name	DC IP prefix
hub1	10.66.66.1/24	65010	10.66.66.254/24	65011	eBGP	DC1	10.44.44.0/24
hub2	10.55.55.1/24	65020	10.55.55.254/24	65021	eBGP	DC2	10.33.33.0/24

NOTE: The AS numbers selected are self-defined and should be private AS and unique. Do not use AS 65000 as its in use internally already!

Extending Applications

Go to **Organization -> Applications** and add the following two new applications with the custom IP address ranges the DCs use:

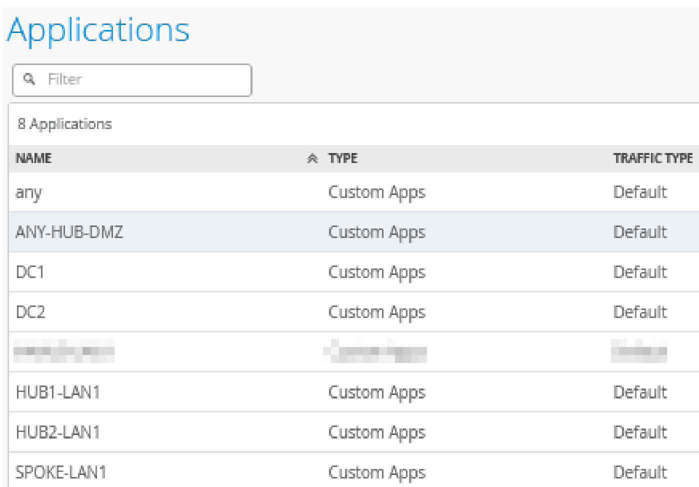
Add a new application and configure the following:

- Name=`DC1`
- Type=`Custom Apps`
- IP Addresses=`10.44.44.0/24`

Add another new application and configure the following:

- Name=`DC2`
- Type=`Custom Apps`
- IP Addresses=`10.33.33.0/24`

The result should look like the figure below:


Applications

Filter

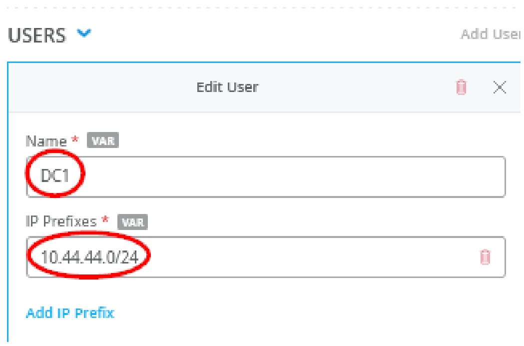
8 Applications

NAME	TYPE	TRAFFIC TYPE
any	Custom Apps	Default
ANY-HUB-DMZ	Custom Apps	Default
DC1	Custom Apps	Default
DC2	Custom Apps	Default
[illegible]	[illegible]	[illegible]
HUB1-LAN1	Custom Apps	Default
HUB2-LAN1	Custom Apps	Default
SPOKE-LAN1	Custom Apps	Default

Extending Networks

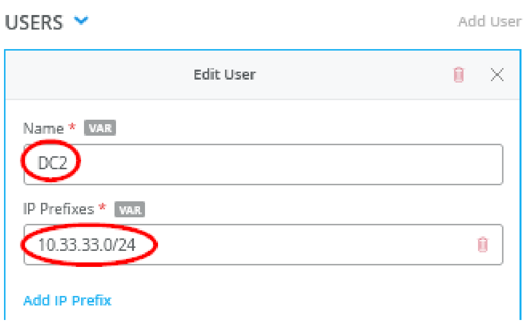
Go to **Organization -> Networks** and edit the existing Network “HUB1-LAN”. You need to add a USERS-Object:

- Name=`DC1`
- IP Prefixes=`10.44.44.0/24`

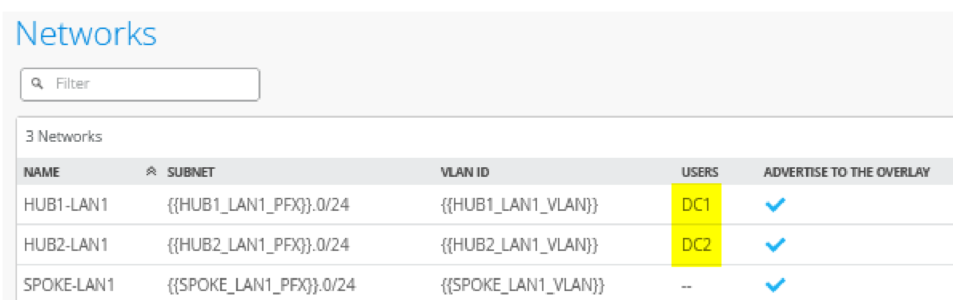


Then, edit the existing Network “HUB2-LAN”. You need to add a USERS-Object:

- Name=`DC2`
- IP Prefixes=`10.33.33.0/24`



The result should look like the figure below:



Extend the Hub1 Profile

Go to **Organization -> Hub Profiles.**

Should you choose to use the import option, click on **Import Profile** and import the below JSON as a file.


```
{
  "dhcpd_config": {
    "enabled": true
  },
  "ntpOverride": true,
  "dnsOverride": true,
  "service_policies": [
    {
      "name": "spoke-to-hub-dmz",
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "HUB1-LAN1",
        "DC1"
      ],
      "action": "allow",
      "path_preference": "HUB-LANS",
      "idp": {
        "enabled": false
      }
    },
    {
      "name": "hub-dmz-to-spoke",
      "tenants": [
        "HUB1-LAN1",
        "DC1.HUB1-LAN1"
      ],
      "services": [
        "SPOKE-LAN1"
      ],
      "action": "allow",
      "local_routing": true,
      "idp": {
        "enabled": false
      }
    },
    {
      "name": "spoke-to-spoke-hairpin",
      "tenants": [
```

```
      "SPOKE-LAN1"
    ],
    "services": [
      "SPOKE-LAN1"
    ],
    "action": "allow",
    "local_routing": true,
    "idp": {
      "enabled": false
    }
  },
  {
    "tenants": [
      "HUB1-LAN1",
      "DC1.HUB1-LAN1"
    ],
    "services": [
      "ANY-HUB-DMZ"
    ],
    "action": "allow",
    "path_preference": "CBO",
    "name": "hub-dmz-to-internet",
    "idp": {
      "enabled": false
    }
  },
  {
    "tenants": [
      "SPOKE-LAN1"
    ],
    "services": [
      "any"
    ],
    "action": "allow",
    "name": "spokes-traffic-cbo-on-hub",
    "path_preference": "CBO",
    "idp": {
      "enabled": false
    }
  },
  {
    "name": "remotehub-to-myhub",
    "tenants": [
```

```
        "HUB2-LAN1",
        "DC2.HUB2-LAN1"
      ],
      "services": [
        "HUB1-LAN1",
        "DC1"
      ],
      "action": "allow",
      "idp": {
        "enabled": false
      },
      "path_preference": "HUB-LANS"
    },
    {
      "name": "myhub-to-remotehub",
      "tenants": [
        "HUB1-LAN1",
        "DC1.HUB1-LAN1"
      ],
      "services": [
        "HUB2-LAN1",
        "DC2"
      ],
      "action": "allow",
      "idp": {
        "enabled": false
      },
      "path_preference": "REMOTEHUB"
    }
  ],
  "ip_configs": {
    "HUB1-LAN1": {
      "type": "static",
      "ip": "{{HUB1_LAN1_PFX}}.1",
      "netmask": "/24"
    }
  },
  "dns_servers": [
    "8.8.8.8",
    "9.9.9.9"
  ],
  "port_config": {
    "ge-0/0/0": {
```

```
      "name": "INET",
      "usage": "wan",
      "aggregated": false,
      "redundant": false,
      "critical": false,
      "disabled": false,
      "wan_type": "broadband",
      "ip_config": {
        "type": "static",
        "ip": "{{WAN0_PFX}}.254",
        "netmask": "/24",
        "gateway": "{{WAN0_PFX}}.1"
      },
      "wan_ext_ip": "{{WAN0_PUBIP}}",
      "disable_autoneg": false,
      "wan_source_nat": {
        "disabled": false
      },
      "vpn_paths": {
        "hub2-INET.OrgOverlay": {
          "role": "spoke"
        },
        "hub1-INET.OrgOverlay": {
          "role": "hub"
        }
      }
    },
    "ge-0/0/1": {
      "name": "MPLS",
      "usage": "wan",
      "aggregated": false,
      "redundant": false,
      "critical": false,
      "disabled": false,
      "wan_type": "broadband",
      "ip_config": {
        "type": "static",
        "ip": "{{WAN1_PFX}}.254",
        "netmask": "/24",
        "gateway": "{{WAN1_PFX}}.1"
      },
      "wan_ext_ip": "{{WAN1_PUBIP}}",
      "disable_autoneg": false,
```

```
      "wan_source_nat": {
        "disabled": false
      },
      "vpn_paths": {
        "hub2-MPLS.OrgOverlay": {
          "role": "spoke"
        },
        "hub1-MPLS.OrgOverlay": {
          "role": "hub"
        }
      }
    },
    "ge-0/0/3": {
      "usage": "lan",
      "networks": [
        "HUB1-LAN1"
      ]
    }
  },
  "bgp_config": {
    "DC1": {
      "networks": [
        "HUB1-LAN1"
      ],
      "via": "lan",
      "type": "external",
      "no_readvertise_to_overlay": false,
      "local_as": 65010,
      "hold_time": 90,
      "graceful_restart_time": 120,
      "neighbors": {
        "10.66.66.254": {
          "disabled": false,
          "neighbor_as": 65011
        }
      },
      "disable_bfd": false
    }
  },
  "routing_policies": {},
  "extra_routes": {},
  "path_preferences": {
    "HUB-LANS": {
```

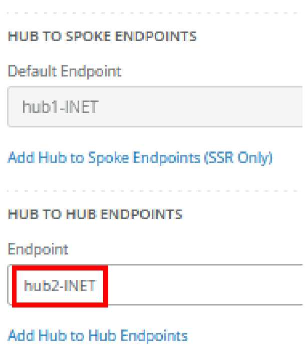
```
      "strategy": "ordered",
      "paths": [
        {
          "type": "local",
          "networks": [
            "HUB1-LAN1"
          ]
        }
      ]
    },
    "CBO": {
      "strategy": "ordered",
      "paths": [
        {
          "name": "INET",
          "type": "wan"
        }
      ]
    },
    "REMOTEHUB": {
      "strategy": "ecmp",
      "paths": [
        {
          "name": "hub2-INET.OrgOverlay",
          "type": "vpn"
        },
        {
          "name": "hub2-MPLS.OrgOverlay",
          "type": "vpn"
        }
      ]
    }
  },
  "ospf_areas": {},
  "vrf_instances": {},
  "tunnel_configs": {},
  "oob_ip_config": {
    "type": "dhcp",
    "node1": {
      "type": "dhcp"
    }
  },
  "tunnel_provider_options": {
```

```
    "jse": {},
    "zscaler": {}
  },
  "ospf_config": {
    "enabled": false,
    "areas": {}
  },
  "name": "hub1",
  "type": "gateway"
}
```

Should you decide to configure everything manually in the Juniper Mist portal, then use the following steps.

Edit the existing WAN “INET” and add:

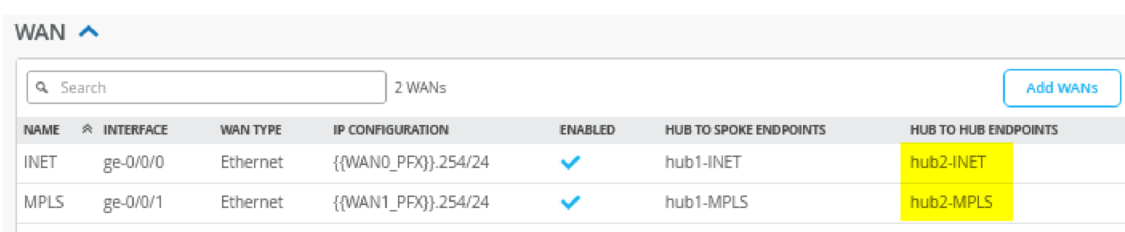
- Hub to Hub Endpoint=`hub2-INET`



Edit the existing WAN “MPLS” and add:

- Hub to Hub Endpoint=`hub2-MPLS`

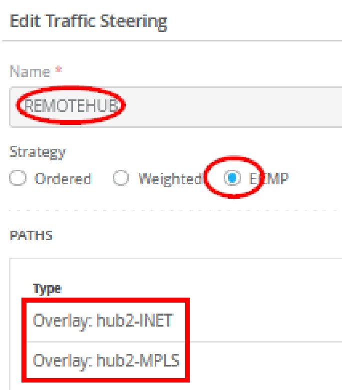
The result should look like the figure below:



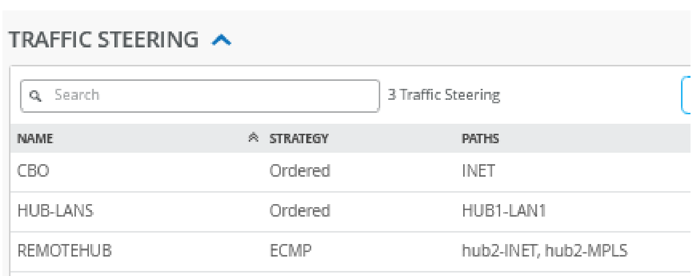
Add the following new traffic steering policy:

- Name=`REMOTEHUB`
- Strategy=`ECMP`

- Paths
  - Path Type1=`Overlay: hub2-INET`
  - Path Type2=`Overlay: hub2-MPLS`



The result should look like the figure below:



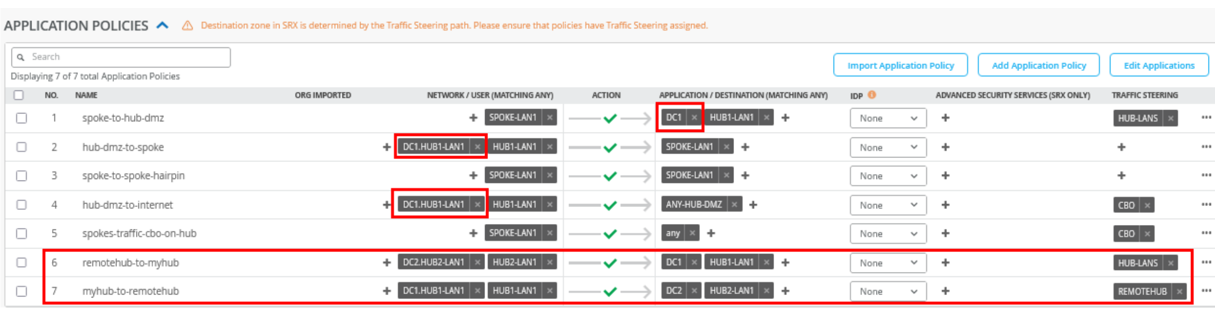
Edit the existing application policies to include the following:

- Number=`1`
  - Name= `spoke-to-hub-dmz`
  - Application=`HUB1-LAN1 + DC1`
- Number=`2`
  - Name= `hub-dmz-to-spoke`
  - Network=`HUB1-LAN1 + DC1.HUB1-LAN1`
- Number=`4`
  - Name=`hub-dmz-to-internet`
  - Network=`HUB1-LAN1 + DC1.HUB1-LAN1`

Add the following two application policies:

- Number=6
  - Name=remoterhub-to-myhub
  - Network=HUB2-LAN1 + DC2.HUB2-LAN1
  - Action=Pass
  - Application=HUB1-LAN + DC1
  - Traffic Steering=HUB-LANS
- Number=7
  - Name= myhub-to-remotehub
  - Network=HUB1-LAN1 + DC1.HUB1-LAN1
  - Action=Pass
  - Application=HUB2-LAN1 + DC2
  - Traffic Steering=REMOTEHUB

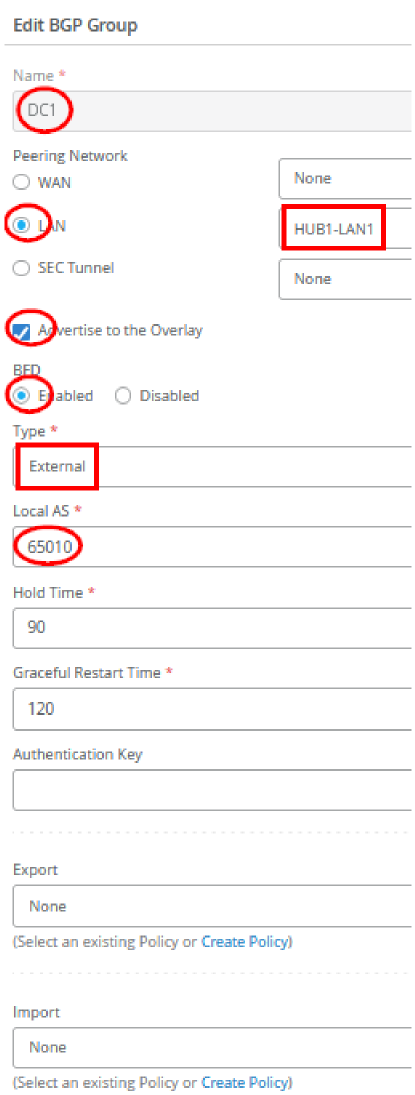
The result should look like the figure below:

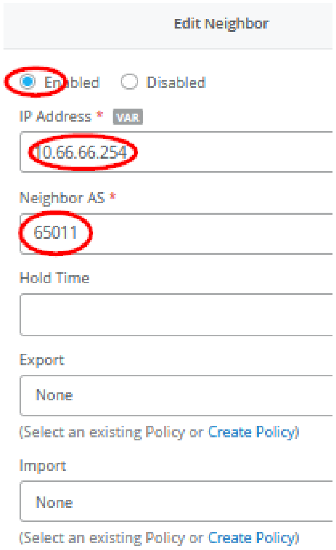


Configure the BGP peering with the data center router as follows:

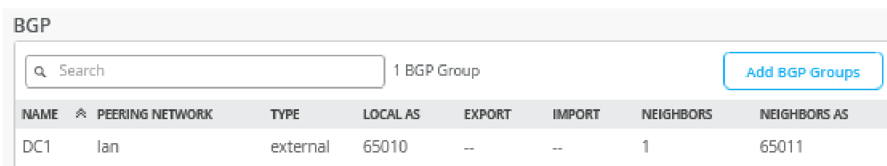
- Name=DC1
- Peering Network LAN=HUB1-LAN1
- Advertise to Overlay=Enabled/Checked
- BFD=Enabled
- Type=External
- Local AS=65010
- Hold Time=90

- Graceful Restart Time=`120`
- Export=`None`
- Import=`None`
- BGP Neighbor
    - Neighbor=`Enabled`
    - IP Address=`10.66.66.254`
    - Neighbor AS=`65011`
    - Export=`None`
    - Import=`None`





The result should look like the figure below:



Save your results.

Extend the Hub2 Profile

Go to **Organization -> Hub Profiles.**

Should you choose to use the import option, click on **Import Profile** and import the below JSON as a file.


```
{
  "dhcpd_config": {
    "enabled": true
  },
  "ntpOverride": true,
  "dnsOverride": true,
  "service_policies": [
    {
      "name": "spoke-to-hub-dmz",
      "tenants": [
```

```
      "SPOKE-LAN1"
    ],
    "services": [
      "HUB2-LAN1",
      "DC2"
    ],
    "action": "allow",
    "path_preference": "HUB-LANS",
    "idp": {
      "enabled": false
    }
  },
  {
    "name": "hub-dmz-to-spoke",
    "tenants": [
      "HUB2-LAN1",
      "DC2.HUB2-LAN1"
    ],
    "services": [
      "SPOKE-LAN1"
    ],
    "action": "allow",
    "local_routing": true,
    "idp": {
      "enabled": false
    }
  },
  {
    "name": "spoke-to-spoke-hairpin",
    "tenants": [
      "SPOKE-LAN1"
    ],
    "services": [
      "SPOKE-LAN1"
    ],
    "action": "allow",
    "local_routing": true,
    "idp": {
      "enabled": false
    }
  },
  {
    "tenants": [
```

```
      "HUB2-LAN1",
      "DC2.HUB2-LAN1"
    ],
    "services": [
      "ANY-HUB-DMZ"
    ],
    "action": "allow",
    "path_preference": "CBO",
    "name": "hub-dmz-to-internet",
    "idp": {
      "enabled": false
    }
  },
  {
    "tenants": [
      "SPOKE-LAN1"
    ],
    "services": [
      "any"
    ],
    "action": "allow",
    "name": "spoke-traffic-cbo-on-hub",
    "path_preference": "CBO",
    "idp": {
      "enabled": false
    }
  },
  {
    "name": "remotehub-to-myhub",
    "tenants": [
      "HUB1-LAN1",
      "DC1.HUB1-LAN1"
    ],
    "services": [
      "HUB2-LAN1",
      "DC2"
    ],
    "action": "allow",
    "idp": {
      "enabled": false
    },
    "path_preference": "HUB-LANS"
  },
```

```
    {
      "name": "myhub-to-remotehub",
      "tenants": [
        "HUB2-LAN1",
        "DC2.HUB2-LAN1"
      ],
      "services": [
        "HUB1-LAN1",
        "DC1"
      ],
      "action": "allow",
      "idp": {
        "enabled": false
      },
      "path_preference": "REMOTEHUB"
    }
  ],
  "ip_configs": {
    "HUB2-LAN1": {
      "type": "static",
      "ip": "{{HUB2_LAN1_PFX}}.1",
      "netmask": "/24"
    }
  },
  "dns_servers": [
    "8.8.8.8",
    "9.9.9.9"
  ],
  "port_config": {
    "ge-0/0/0": {
      "name": "INET",
      "usage": "wan",
      "aggregated": false,
      "redundant": false,
      "critical": false,
      "disabled": false,
      "wan_type": "broadband",
      "ip_config": {
        "type": "static",
        "ip": "{{WAN0_PFX}}.254",
        "netmask": "/24",
        "gateway": "{{WAN0_PFX}}.1"
      },
```

```
      "wan_ext_ip": "{{WAN0_PUBIP}}",
      "disable_autoneg": false,
      "wan_source_nat": {
        "disabled": false
      },
      "vpn_paths": {
        "hub1-INET.OrgOverlay": {
          "role": "spoke"
        },
        "hub2-INET.OrgOverlay": {
          "role": "hub"
        }
      }
    },
    "ge-0/0/1": {
      "name": "MPLS",
      "usage": "wan",
      "aggregated": false,
      "redundant": false,
      "critical": false,
      "disabled": false,
      "wan_type": "broadband",
      "ip_config": {
        "type": "static",
        "ip": "{{WAN1_PFX}}.254",
        "netmask": "/24",
        "gateway": "{{WAN1_PFX}}.1"
      },
      "wan_ext_ip": "{{WAN1_PUBIP}}",
      "disable_autoneg": false,
      "wan_source_nat": {
        "disabled": false
      },
      "vpn_paths": {
        "hub1-MPLS.OrgOverlay": {
          "role": "spoke"
        },
        "hub2-MPLS.OrgOverlay": {
          "role": "hub"
        }
      }
    },
    "ge-0/0/3": {
```

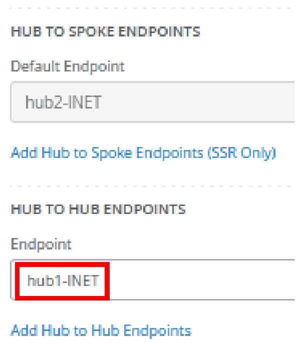
```
      "usage": "lan",
      "aggregated": false,
      "redundant": false,
      "networks": [
        "HUB2-LAN1"
      ]
    }
  },
  "bgp_config": {
    "DC2": {
      "networks": [
        "HUB2-LAN1"
      ],
      "via": "lan",
      "type": "external",
      "no_readvertise_to_overlay": false,
      "local_as": 65020,
      "hold_time": 90,
      "graceful_restart_time": 120,
      "neighbors": {
        "10.55.55.254": {
          "disabled": false,
          "neighbor_as": 65021
        }
      },
      "disable_bfd": false
    }
  },
  "routing_policies": {},
  "extra_routes": {},
  "path_preferences": {
    "HUB-LANS": {
      "strategy": "ordered",
      "paths": [
        {
          "type": "local",
          "networks": [
            "HUB2-LAN1"
          ]
        }
      ]
    },
    "CBO": {
```

```
      "strategy": "ordered",
      "paths": [
        {
          "name": "INET",
          "type": "wan"
        }
      ]
    },
    "REMOTEHUB": {
      "strategy": "ecmp",
      "paths": [
        {
          "name": "hub1-INET.OrgOverlay",
          "type": "vpn"
        },
        {
          "name": "hub1-MPLS.OrgOverlay",
          "type": "vpn"
        }
      ]
    }
  },
  "ospf_areas": {},
  "vrf_instances": {},
  "tunnel_configs": {},
  "oob_ip_config": {
    "type": "dhcp",
    "node1": {
      "type": "dhcp"
    }
  },
  "tunnel_provider_options": {
    "jse": {},
    "zscaler": {}
  },
  "ospf_config": {
    "enabled": false,
    "areas": {}
  },
  "name": "hub2",
  "type": "gateway"
}
```

Should you decide to configure everything manually in the Juniper Mist portal, then use the following steps.

Edit the existing WAN “INET” and add:

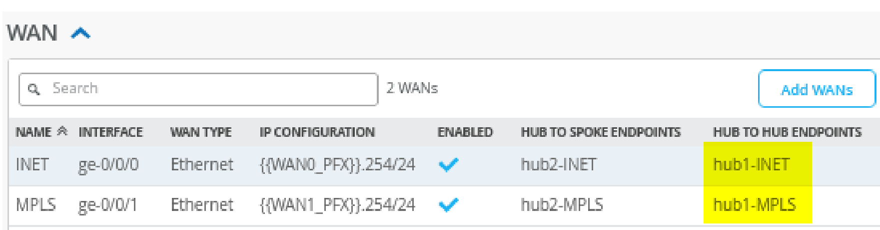
- Hub to Hub Endpoint=`hub1-INET`



Edit the existing WAN “MPLS” and add:

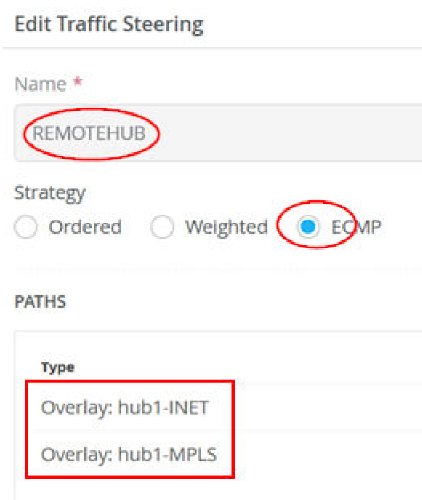
- Hub to Hub Endpoint=`hub1-MPLS`

The result should look like the figure below:

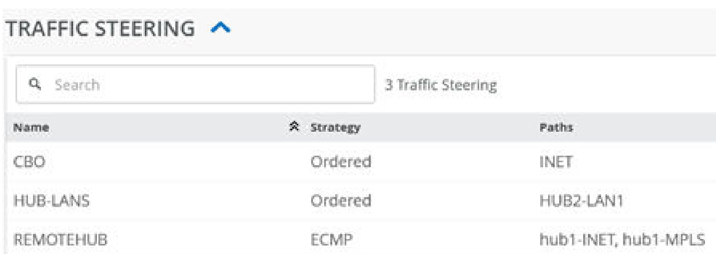


Add the following new traffic steering policy:

- Name=`REMOTEHUB`
- Strategy=`ECMP`
- Paths
  - Path Type1=`Overlay: hub1-INET`
  - Path Type2=`Overlay: hub1-MPLS`



The result should look like the figure below:



Name	Strategy	Paths
CBO	Ordered	INET
HUB-LANS	Ordered	HUB2-LAN1
REMOTEHUB	ECMP	hub1-INET, hub1-MPLS

Edit the existing application policies to include the following:

- Number=`1`
  - Name= `spoke-to-hub-dmz`
  - Application=`HUB2-LAN1` + `DC2`
- Number=`2`
  - Name= `hub-dmz-to-spoke`
  - Network=`HUB2-LAN1` + `DC2.HUB2-LAN1`
- Number=`4`
  - Name=`hub-dmz-to-internet`
  - Network=`HUB2-LAN1` + `DC2.HUB2-LAN1`

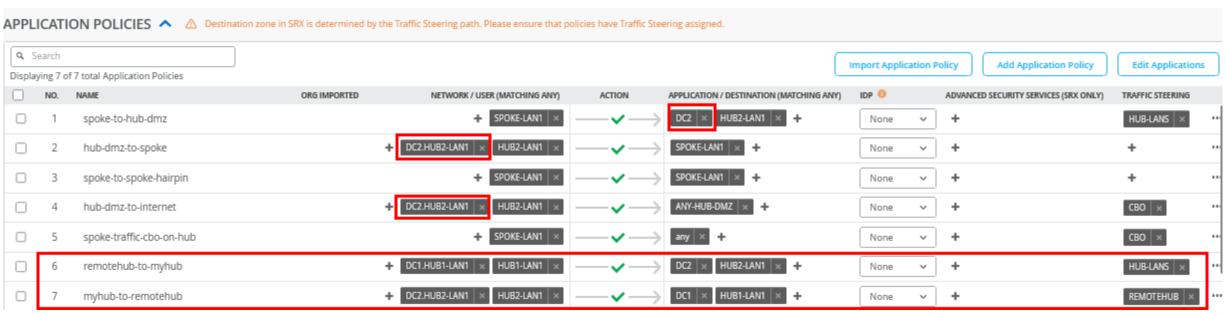
Add the following two application policies:

- Number=`6`
  - Name=`remoterhub-to-myhub`

- Network=`HUB1-LAN1 + DC1.HUB1-LAN1`
- Action=`Pass`
- Application=`HUB2-LAN + DC2`
- Traffic Steering=`HUB-LANS`

- Number=`7`
  - Name= `myhub-to-remotehub`
  - Network=`HUB2-LAN1 + DC2.HUB1-LAN1`
  - Action=`Pass`
  - Application=`HUB1-LAN1 + DC1`
  - Traffic Steering=`REMOTEHUB`

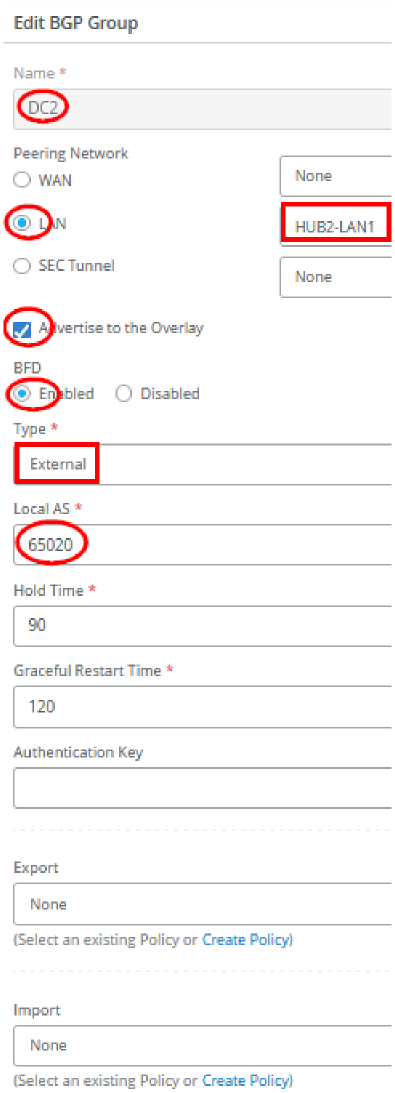
The result should look like the figure below:

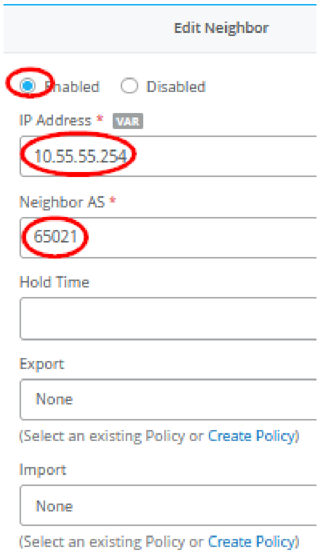


Configure the BGP peering with the data center router as follows:

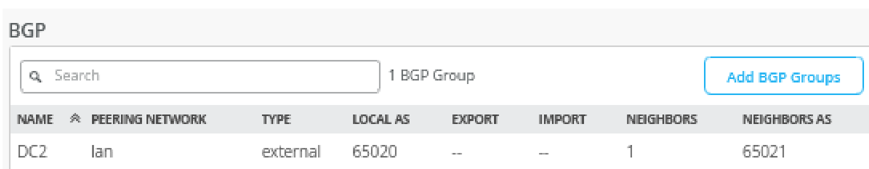
- Name=`DC2`
- Peering Network LAN=`HUB2-LAN1`
- Advertise to Overlay=`Enabled/Checked`
- BFD=`Enabled`
- Type=`External`
- Local AS=`65020`
- Hold Time=`90`
- Graceful Restart Time=`120`
- Export=`None`

- Import=`None`
- BGP Neighbor
  - Neighbor=`Enabled`
  - IP Address=`10.55.55.254`
  - Neighbor AS=`65021`
  - Export=`None`
  - Import=`None`





The result should look like the figure below:



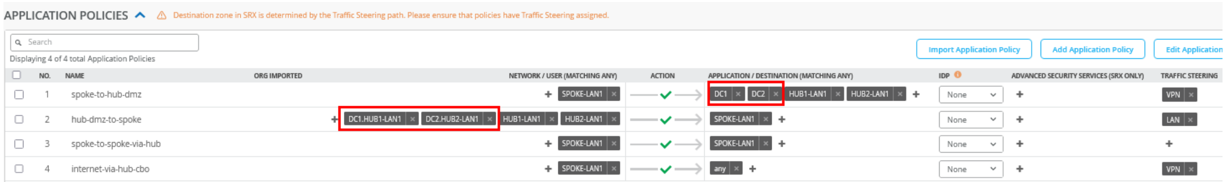
Save your results.

Extend the Spokes Template

The new DC1 and DC2 subnets are added here to the existing rules for visibility.

Edit the following Application Policies:

- Number=`1`
  - Name=`spoke-to-hub-dmz`
  - Application=`HUB1-LAN1 + HUB2-LAN1 + DC1 + DC2`
- Number=`2`
  - Name=`hub-dmz-to-spoke`
  - Network=`HUB1-LAN1 + HUB2-LAN1 + DC1.HUB1-LAN1 + DC2.HUB1-LAN1`



Configuring DC Routers

There are many ways your data center routers can be configured to share routes using eBGP with the hub they are attached to. In our example below, we use an Ubuntu Linux-based VM with the BIRD Internet Routing Daemon for this exchange. Feel free to reuse or utilize other frameworks.

The below example shares the network and BIRD configuration used on DC-Router1 of this topology:

```
              cat /etc/netplan/01-netcfg.yaml
network:
  version: 2
  renderer: networkd
  ethernets:
    ens3:
      addresses:
        - 192.168.10.71/24
      dhcp4: false
    ens4:
      dhcp4: false
    ens5:
      dhcp4: false
  vlans:
    vlan1066:
      id: 1066
      link: ens5
      addresses: [10.66.66.254/24]
      gateway4: 10.66.66.1
      nameservers:
        addresses: [8.8.8.8, 9.9.9.9]
    vlan1044:
      id: 1044
      link: ens4
      addresses: [10.44.44.1/24]
```

```
#
# enable forwarding between interfaces
echo 'net.ipv4.ip_forward=1' >>/etc/sysctl.conf
              sudo sysctl -p
#
# install bgp daemon
apt-get install -y bird
#
cp /etc/bird/bird.conf /etc/bird/bird.conf.orig
#
# configure bgp daemon
cat<<EOF >/etc/bird/bird.conf
# Configure logging
log syslog all;
# Override router ID
router id 10.66.66.254;
# This pseudo-protocol performs synchronization between BIRD's routing
# tables and the kernel. If your kernel supports multiple routing tables
# (as Linux 2.2.x does), you can run multiple instances of the kernel
# protocol and synchronize different kernel tables with different BIRD tables.
protocol kernel {
# learn;          # Learn all alien routes from the kernel
  persist;        # Don't remove routes on bird shutdown
  scan time 20;       # Scan kernel routing table every 20 seconds
# import none;        # Default is import all
  export all;     # Default is export none
# kernel table 5;     # Kernel table to synchronize with (default: main)
}
# This pseudo-protocol watches all interface up/down events.
protocol device {
  scan time 10;       # Scan interfaces every 10 seconds
}
# add out local IF towards desktop6 VM to the table
protocol direct direct1 {
    interface "vlan1044";
}
#BGP Configuration
protocol bgp Spoke1 {
        import all;
        export where proto = "direct1";
        local as 65011;
                     neighbor 10.66.66.1 as 65010;
}
```

```
EOF
#
# disable and restart our bgp-daemon with the new config
systemctl disable bird
              systemctl restart bird
```

The below example shares the network and BIRD configuration used on DC-Router2 of this topology:

```
              cat /etc/netplan/01-netcfg.yaml
network:
  version: 2
  renderer: networkd
  ethernets:
    ens3:
      addresses:
        - 192.168.10.72/24
      dhcp4: false
    ens4:
      dhcp4: false
    ens5:
      dhcp4: false
  vlans:
    vlan1055:
      id: 1055
      link: ens5
      addresses: [10.55.55.254/24]
      gateway4: 10.55.55.1
      nameservers:
        addresses: [8.8.8.8, 9.9.9.9]
    vlan1033:
      id: 1033
      link: ens4
      addresses: [10.33.33.1/24]
.
# enable forwarding between interfaces
echo 'net.ipv4.ip_forward=1' >>/etc/sysctl.conf
              sudo sysctl -p
#
# install bgp daemon
apt-get install -y bird
```

```
#
cp /etc/bird/bird.conf /etc/bird/bird.conf.orig
#
# configure bgp daemon
cat<<EOF >/etc/bird/bird.conf
# Configure logging
log syslog all;
# Override router ID
router id 10.55.55.254;
# This pseudo-protocol performs synchronization between BIRD's routing
# tables and the kernel. If your kernel supports multiple routing tables
# (as Linux 2.2.x does), you can run multiple instances of the kernel
# protocol and synchronize different kernel tables with different BIRD tables.
protocol kernel {
# learn;          # Learn all alien routes from the kernel
  persist;        # Don't remove routes on bird shutdown
  scan time 20;       # Scan kernel routing table every 20 seconds
# import none;        # Default is import all
  export all;     # Default is export none
# kernel table 5;     # Kernel table to synchronize with (default: main)
}
# This pseudo-protocol watches all interface up/down events.
protocol device {
  scan time 10;       # Scan interfaces every 10 seconds
}
# add out local IF towards desktop6 VM to the table
protocol direct direct1 {
    interface "vlan1033";
}
#BGP Configuration
protocol bgp Spoke1 {
        import all;
        export where proto = "direct1";
        local as 65021;
                     neighbor 10.55.55.1 as 65020;
}
EOF
#
# disable and restart our bgp-daemon with the new config
systemctl disable bird
              systemctl restart bird
```

Test your network configuration

After the configuration is done, we can now test the new network configuration and verify the traffic between the two data centers via the two hubs.

The configuration on the Router1 VM now displays the exchanged routes both locally and within the BIRD process. Among these, key routes include the direct interface route to Hub2 (`10.55.55.0/24`) and the propagated data center route from DC2 (`10.33.33.0/24`).

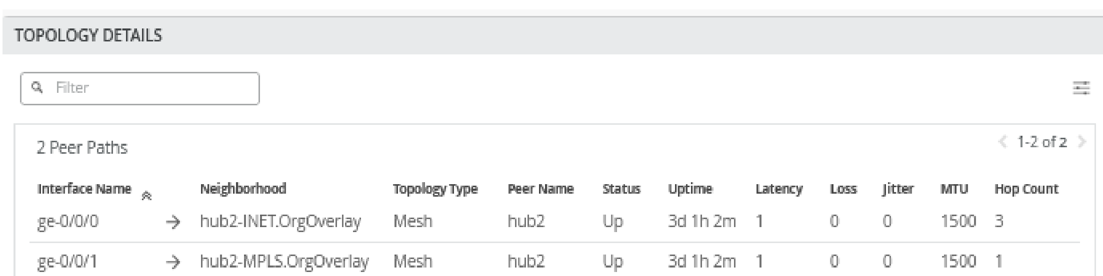
```
root@router1:~# ip route
default via 10.66.66.1 dev vlan1066 proto static
10.0.0.0/8 via 10.66.66.1 dev vlan1066 proto bird
10.33.33.0/24 via 10.66.66.1 dev vlan1066 proto bird
10.44.44.0/24 dev vlan1044 proto kernel scope link src 10.44.44.1
10.55.55.0/24 via 10.66.66.1 dev vlan1066 proto bird
10.66.66.0/24 dev vlan1066 proto kernel scope link src 10.66.66.254
10.77.77.0/24 via 10.66.66.1 dev vlan1066 proto bird
10.88.88.0/24 via 10.66.66.1 dev vlan1066 proto bird
10.99.99.0/24 via 10.66.66.1 dev vlan1066 proto bird
192.168.10.0/24 dev ens3 proto kernel scope link src 192.168.10.71
#
# check the BGP daemon
root@router1:~# birdc
BIRD 1.6.8 ready.
bird> show route
0.0.0.0/0          via 10.66.66.1 on vlan1066 [Spoke1 12:29:03] ! (100) [AS65000?]
10.0.0.0/8         via 10.66.66.1 on vlan1066 [Spoke1 12:29:03] * (100) [AS65000?]
10.88.88.0/24      via 10.66.66.1 on vlan1066 [Spoke1 12:29:03] * (100) [AS65000?]
10.66.66.0/24      via 10.66.66.1 on vlan1066 [Spoke1 12:29:03] ! (100) [AS65000?]
10.77.77.0/24      via 10.66.66.1 on vlan1066 [Spoke1 12:29:03] * (100) [AS65000?]
10.44.44.0/24      dev vlan1044 [direct1 12:24:39] * (240)
10.33.33.0/24      via 10.66.66.1 on vlan1066 [Spoke1 12:35:45] * (100) [AS65021i]
10.55.55.0/24      via 10.66.66.1 on vlan1066 [Spoke1 12:29:03] * (100) [AS65000?]
10.99.99.0/24      via 10.66.66.1 on vlan1066 [Spoke1 12:29:03] * (100) [AS65000?]
```

The configuration on the Router2 VM now displays the exchanged routes both locally and within the BIRD process. Among these, key routes include the direct interface route to Hub2 (`10.66.66.0/24`) and the propagated data center route from DC1 (`10.44.44.0/24`).

```
root@router2:~# ip route
default via 10.55.55.1 dev vlan1055 proto static
```

```
10.0.0.0/8 via 10.55.55.1 dev vlan1055 proto bird
10.33.33.0/24 dev vlan1033 proto kernel scope link src 10.33.33.1
10.44.44.0/24 via 10.55.55.1 dev vlan1055 proto bird
10.55.55.0/24 dev vlan1055 proto kernel scope link src 10.55.55.254
10.66.66.0/24 via 10.55.55.1 dev vlan1055 proto bird
10.77.77.0/24 via 10.55.55.1 dev vlan1055 proto bird
10.88.88.0/24 via 10.55.55.1 dev vlan1055 proto bird
10.99.99.0/24 via 10.55.55.1 dev vlan1055 proto bird
192.168.10.0/24 dev ens3 proto kernel scope link src 192.168.10.72
#
# check the BGP daemon
root@router2:~# birdc
BIRD 1.6.8 ready.
bird> show route
0.0.0.0/0          via 10.55.55.1 on vlan1055 [Spoke1 12:35:46] ! (100) [AS65000?]
10.0.0.0/8         via 10.55.55.1 on vlan1055 [Spoke1 12:35:46] * (100) [AS65000?]
10.88.88.0/24      via 10.55.55.1 on vlan1055 [Spoke1 12:35:46] * (100) [AS65000?]
10.66.66.0/24      via 10.55.55.1 on vlan1055 [Spoke1 12:35:46] * (100) [AS65000?]
10.77.77.0/24      via 10.55.55.1 on vlan1055 [Spoke1 12:35:46] * (100) [AS65000?]
10.44.44.0/24      via 10.55.55.1 on vlan1055 [Spoke1 12:35:46] * (100) [AS65011i]
10.33.33.0/24      dev vlan1033 [direct1 12:35:44] * (240)
10.55.55.0/24      via 10.55.55.1 on vlan1055 [Spoke1 12:35:46] ! (100) [AS65000?]
10.99.99.0/24      via 10.55.55.1 on vlan1055 [Spoke1 12:35:46] * (100) [AS65000?]
```

When you go to **WAN Edges -> hub1-site -> hub1** you can see the additional overlay tunnels:



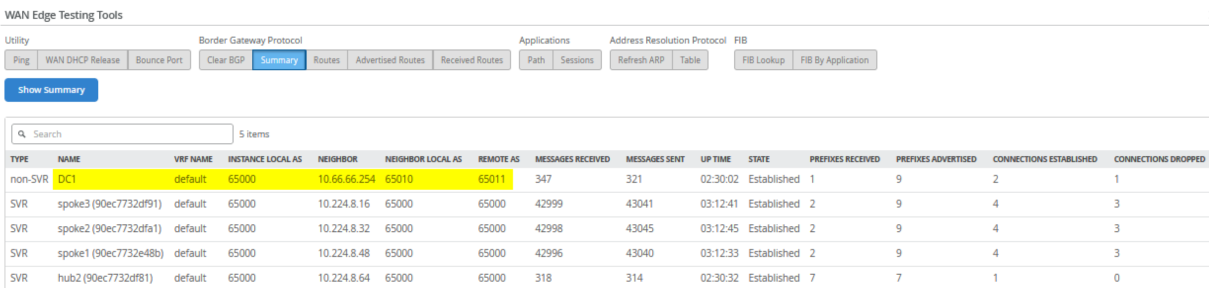
TOPOLOGY DETAILS

Filter

2 Peer Paths — 1-2 of 2

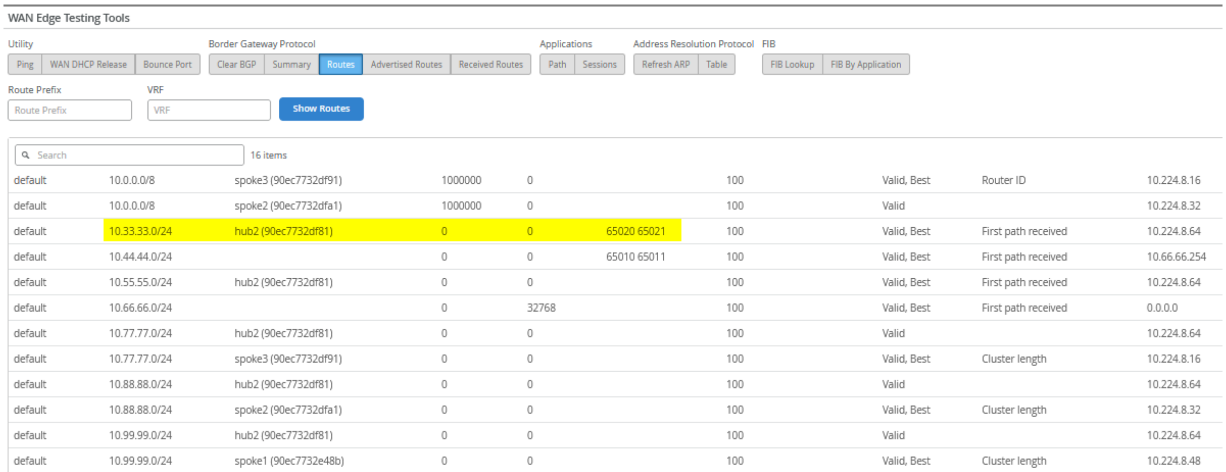
Interface Name		Neighborhood	Topology Type	Peer Name	Status	Uptime	Latency	Loss	Jitter	MTU	Hop Count
ge-0/0/0	→	hub2-INET.OrgOverlay	Mesh	hub2	Up	3d 1h 2m	1	0	0	1500	3
ge-0/0/1	→	hub2-MPLS.OrgOverlay	Mesh	hub2	Up	3d 1h 2m	1	0	0	1500	1

Go further to **Utilities -> Testing Tools** and click on **BGP – Summary** you can see the BGP neighbor summary on Hub1:



TYPE	NAME	VRF NAME	INSTANCE LOCAL AS	NEIGHBOR	NEIGHBOR LOCAL AS	REMOTE AS	MESSAGES RECEIVED	MESSAGES SENT	UP TIME	STATE	PREFIXES RECEIVED	PREFIXES ADVERTISED	CONNECTIONS ESTABLISHED	CONNECTIONS DROPPED
non-SVR	DC1	default	65000	10.66.66.254	65010	65011	347	321	02:30:02	Established	1	9	2	1
SVR	spoke3 (90ec7732df91)	default	65000	10.224.8.16	65000	65000	42999	43041	03:12:41	Established	2	9	4	3
SVR	spoke2 (90ec7732dfa1)	default	65000	10.224.8.32	65000	65000	42998	43045	03:12:45	Established	2	9	4	3
SVR	spoke1 (90ec7732e48b)	default	65000	10.224.8.48	65000	65000	42996	43040	03:12:33	Established	2	9	4	3
SVR	hub2 (90ec7732df81)	default	65000	10.224.8.64	65000	65000	318	314	02:30:32	Established	7	7	1	0

Then, check the routes in the system. Here, it’s important to receive the remote DC2 route `10.33.33.0/24` .



default	10.0.0.0/8	spoke3 (90ec7732df91)	1000000	0		100	Valid, Best	Router ID	10.224.8.16
default	10.0.0.0/8	spoke2 (90ec7732dfa1)	1000000	0		100	Valid		10.224.8.32
default	10.33.33.0/24	hub2 (90ec7732df81)	0	0	65020 65021	100	Valid, Best	First path received	10.224.8.64
default	10.44.44.0/24		0	0	65010 65011	100	Valid, Best	First path received	10.66.66.254
default	10.55.55.0/24	hub2 (90ec7732df81)	0	0		100	Valid, Best	First path received	10.224.8.64
default	10.66.66.0/24		0	32768		100	Valid, Best	First path received	0.0.0.0
default	10.77.77.0/24	hub2 (90ec7732df81)	0	0		100	Valid		10.224.8.64
default	10.77.77.0/24	spoke3 (90ec7732df91)	0	0		100	Valid, Best	Cluster length	10.224.8.16
default	10.88.88.0/24	hub2 (90ec7732df81)	0	0		100	Valid		10.224.8.64
default	10.88.88.0/24	spoke2 (90ec7732dfa1)	0	0		100	Valid, Best	Cluster length	10.224.8.32
default	10.99.99.0/24	hub2 (90ec7732df81)	0	0		100	Valid		10.224.8.64
default	10.99.99.0/24	spoke1 (90ec7732e48b)	0	0		100	Valid, Best	Cluster length	10.224.8.48

Now check the traffic utilizing the desktop6 VM which acts as a service in DC1 with the IP address `10.44.44.44`:

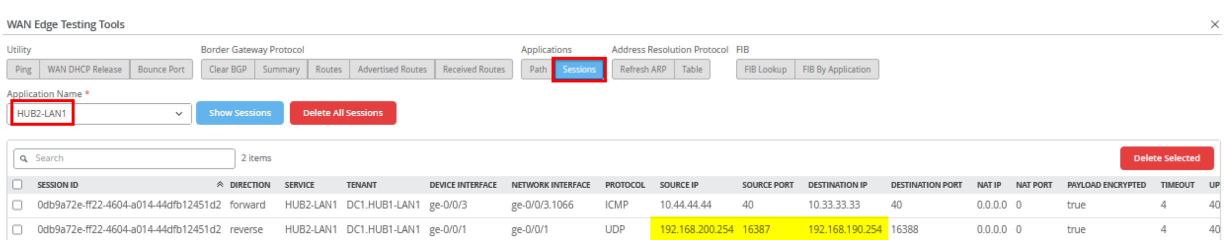
```
# ping hub1 local interface
root@desktop6:~# ping -c3 10.66.66.1
PING 10.66.66.1 (10.66.66.1) 56(84) bytes of data.
64 bytes from 10.66.66.1: icmp_seq=1 ttl=127 time=0.757 ms
64 bytes from 10.66.66.1: icmp_seq=2 ttl=127 time=0.769 ms
64 bytes from 10.66.66.1: icmp_seq=3 ttl=127 time=0.776 ms
#
# check connection to internet via hub CBO
root@desktop6:~# ping -c3 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=50 time=6.34 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=50 time=3.78 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=50 time=3.81 ms
#
# check connection to spoke1
root@desktop6:~# ping -c3 10.99.99.99
```

```
PING 10.99.99.99 (10.99.99.99) 56(84) bytes of data.
64 bytes from 10.99.99.99: icmp_seq=1 ttl=60 time=4.08 ms
64 bytes from 10.99.99.99: icmp_seq=2 ttl=60 time=1.65 ms
64 bytes from 10.99.99.99: icmp_seq=3 ttl=60 time=1.65 ms
#
# verify connection to DC2 local hub LAN
root@desktop6:~# ping -c3 10.55.55.55
PING 10.55.55.55 (10.55.55.55) 56(84) bytes of data.
64 bytes from 10.55.55.55: icmp_seq=1 ttl=58 time=2.98 ms
64 bytes from 10.55.55.55: icmp_seq=2 ttl=58 time=1.52 ms
64 bytes from 10.55.55.55: icmp_seq=3 ttl=58 time=1.45 ms
```

The most important check is to reach the desktop7 VM IP address `10.33.33.33` which acts as a service in the remote DC2. This verifies the hub-to-hub overlay is working as expected.

```
root@desktop6:~# ping 10.33.33.33
PING 10.33.33.33 (10.33.33.33) 56(84) bytes of data.
64 bytes from 10.33.33.33: icmp_seq=1 ttl=59 time=3.53 ms
64 bytes from 10.33.33.33: icmp_seq=2 ttl=59 time=1.88 ms
64 bytes from 10.33.33.33: icmp_seq=3 ttl=59 time=1.84 ms
64 bytes from 10.33.33.33: icmp_seq=4 ttl=59 time=1.78 ms
64 bytes from 10.33.33.33: icmp_seq=5 ttl=59 time=1.83 ms
.
.
```

Let the `ping 10.33.33.33` on desktop6 VM continuously run and then check on Hub1 **Applications – Session** with the application name `HUB2-LAN1` as the destination. The reverse traffic source and destination IP will indirectly determine which traffic path is used for this traffic which is MPLS as seen in the figure below:

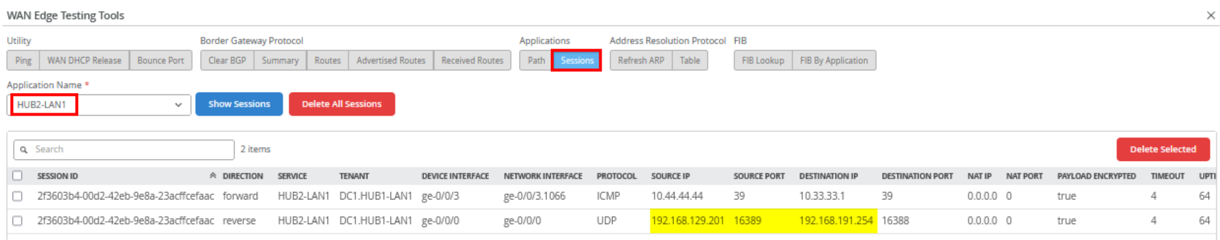


Stop the ping and start a new continuous ping to `10.33.33.1` on the desktop6 VM as we need a different destination IP address now.

```
root@desktop6:~# ping 10.33.33.1
PING 10.33.33.1 (10.33.33.1) 56(84) bytes of data.
```

```
64 bytes from 10.33.33.1: icmp_seq=1 ttl=58 time=2.80 ms
64 bytes from 10.33.33.1: icmp_seq=2 ttl=58 time=1.46 ms
64 bytes from 10.33.33.1: icmp_seq=3 ttl=58 time=1.55 ms
64 bytes from 10.33.33.1: icmp_seq=4 ttl=58 time=1.52 ms
.
.
```

Then, check on Hub1 **Applications – Session** with the application name `HUB2-LAN1` as the destination. The reverse traffic source and destination IP will indirectly determine which traffic path is used for this traffic which is INET as seen in the figure below. This verifies that the ECMP-based traffic steering between the two Hubs is working as expected.



The final test involves verifying that a VM connected to a spoke can access resources in both DC1 and DC2. This is demonstrated below using the desktop1 VM:

```
# check connection to DC1
root@desktop1:~# ping -c3 10.44.44.44
PING 10.44.44.44 (10.44.44.44) 56(84) bytes of data.
64 bytes from 10.44.44.44: icmp_seq=1 ttl=58 time=2.84 ms
64 bytes from 10.44.44.44: icmp_seq=2 ttl=58 time=1.51 ms
64 bytes from 10.44.44.44: icmp_seq=3 ttl=58 time=1.52 ms
#
# check connection to DC2
root@desktop1:~# ping -c3 10.33.33.33
PING 10.33.33.33 (10.33.33.33) 56(84) bytes of data.
64 bytes from 10.33.33.33: icmp_seq=1 ttl=58 time=3.82 ms
64 bytes from 10.33.33.33: icmp_seq=2 ttl=58 time=1.37 ms
64 bytes from 10.33.33.33: icmp_seq=3 ttl=58 time=1.41 ms
```

Appendix: Building a High Availability Hub-and-Spoke Using SSR Chassis Cluster Pairs Topology

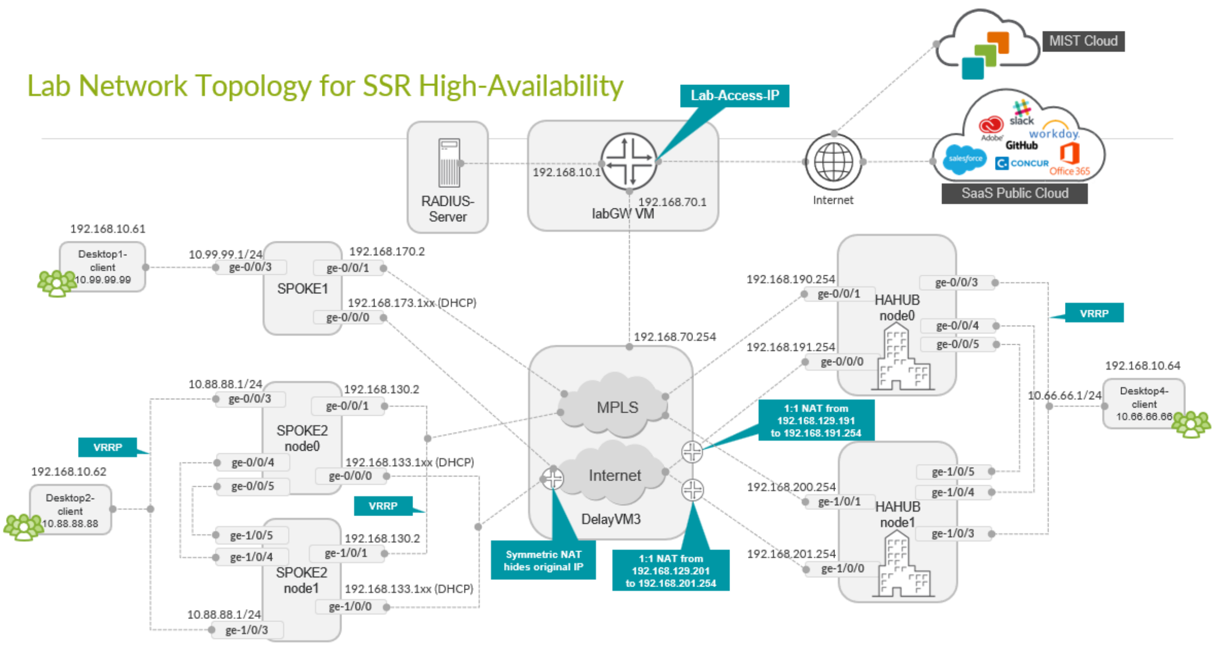
IN THIS SECTION

This lab builds on the previous lab "Appendix: Building a base SD-WAN Topology with Three Spokes and Two Hubs" on page 23. The underlay connections change a bit here.

- Each pair of devices configured for high availability must have two direct links between them. This is a mandatory requirement, just like ensuring both devices are running the same software version to maintain consistent technical specifications and seamless failover operation.
- To keep the lab setup simple, we did not modify the IP addressing on the clustered hub's WAN interfaces. Avoid using redundant interface configurations on these links, as only one of the two would be active at a time. In the recommended configuration, all four links remain active simultaneously.
- On the clustered hub's LAN interface, link redundancy is configured, allowing only one active link at a time. VRRP is used to manage failover between the two cluster nodes.
- On the clustered spoke WAN interface, we've configured the following:
  - The Internet path uses DHCP leases, hence you cannot configure link redundancy here. You can attach them to the same broadband router as indicated in the topology and each will receive its own IP address from the same subnet.
  - The MPLS path uses a single static IP address and is configured with link redundancy. As shown in the topology, both links can connect to the same PE router, but only one will be active at a time. VRRP handles failover between the two cluster nodes.

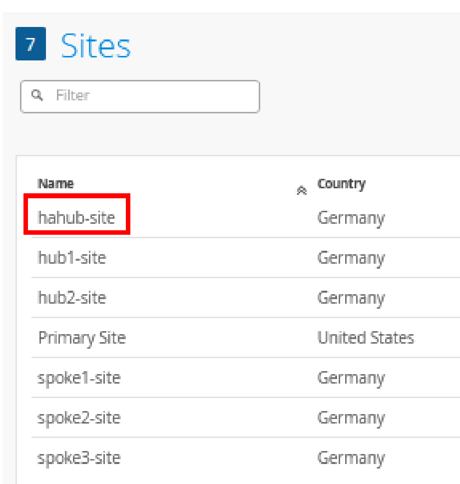
- Link redundancy is configured on the clustered spoke's LAN interface, allowing only one link to be active at a time. VRRP is used to enable failover between the two cluster nodes.

With this topology, you can test all possible failover scenarios for the JVD, but you may need to adapt this for an individual deployment environment.



Create a Site and Extend Applications and Networks

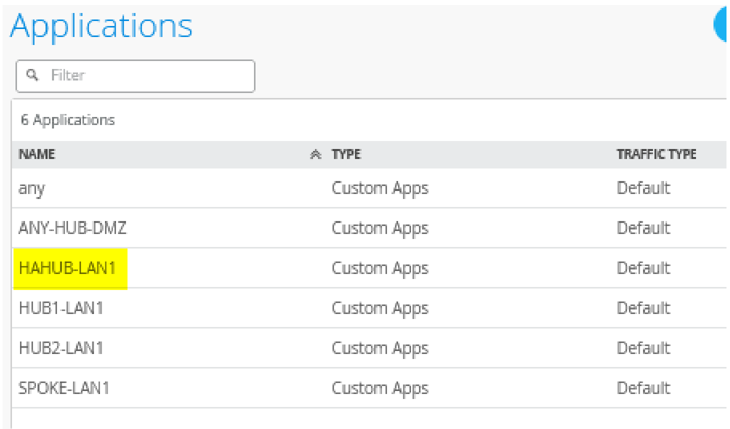
Go to **Organization -> Site Configuration** and create a new site called "hahub-site". This time there is no need to configure site variables as the hub profile will be unique anyway. The result is displayed below:



Then, go **Organization -> Applications** and create a new custom application like the following:

- Name=`HAHUB-LAN1`
- Type=`Custom Apps`
- IP Address=`10.66.66.0/24`

The result should look like the figure below:


Applications

Filter

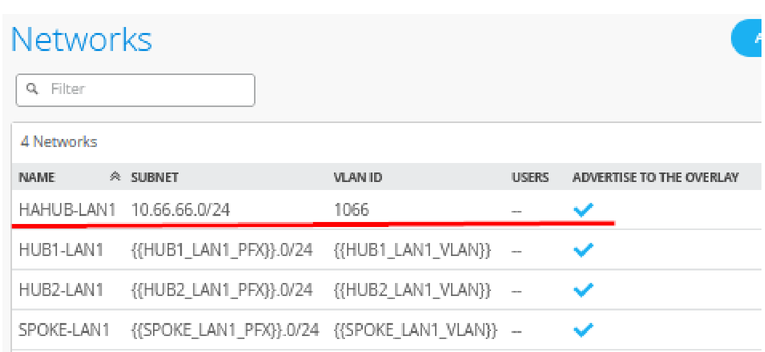
6 Applications

NAME	TYPE	TRAFFIC TYPE
any	Custom Apps	Default
ANY-HUB-DMZ	Custom Apps	Default
HAHUB-LAN1	Custom Apps	Default
HUB1-LAN1	Custom Apps	Default
HUB2-LAN1	Custom Apps	Default
SPOKE-LAN1	Custom Apps	Default

Then go **Organization -> Networks** and create a new network like:

- Name=`HAHUB-LAN1`
- Subnet IP Address=`10.66.66.0`
- Prefix Length=`24` (we only use a /24 netmask in our example for ease of use)
- VLAN ID=`1066`
- Access to Mist Cloud=`Checked/Enabled`
- Advertised via Overlay=`Checked/Enabled`

The result should look like the figure below:


Networks

Filter

4 Networks

NAME	SUBNET	VLAN ID	USERS	ADVERTISE TO THE OVERLAY
HAHUB-LAN1	10.66.66.0/24	1066	–	✓
HUB1-LAN1	{{HUB1_LAN1_PFX}}.0/24	{{HUB1_LAN1_VLAN}}	–	✓
HUB2-LAN1	{{HUB2_LAN1_PFX}}.0/24	{{HUB2_LAN1_VLAN}}	–	✓
SPOKE-LAN1	{{SPOKE_LAN1_PFX}}.0/24	{{SPOKE_LAN1_VLAN}}	–	✓

Create a High-Availability Hub Profile

Go to **Organization -> Hub Profiles.**

Should you choose to use the import option, click on **Import Profile** and import the below JSON as a file.


```
{
  "dhcpd_config": {
    "enabled": true
  },
  "ntpOverride": true,
  "dnsOverride": true,
  "service_policies": [
    {
      "name": "spoke-to-hub-dmz",
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "HAHUB-LAN1"
      ],
      "action": "allow",
      "path_preference": "HUB-LANS",
      "idp": {
        "enabled": false
      }
    },
    {
      "name": "hub-dmz-to-spoke",
      "tenants": [
        "HAHUB-LAN1"
      ],
      "services": [
        "SPOKE-LAN1"
      ],
      "action": "allow",
      "local_routing": true,
      "idp": {
        "enabled": false
      }
    },
    {
```

```
      "name": "spoke-to-spoke-hairpin",
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "SPOKE-LAN1"
      ],
      "action": "allow",
      "local_routing": true,
      "idp": {
        "enabled": false
      }
    },
    {
      "tenants": [
        "HAHUB-LAN1"
      ],
      "services": [
        "ANY-HUB-DMZ"
      ],
      "action": "allow",
      "name": "hub-dmz-to-internet",
      "idp": {
        "enabled": false
      },
      "path_preference": "CBO"
    },
    {
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "any"
      ],
      "action": "allow",
      "name": "spokes-traffic-cbo-on-hub",
      "idp": {
        "enabled": false
      },
      "path_preference": "CBO"
    }
  ],
  "ip_configs": {
```

```
    "HAHUB-LAN1": {
      "type": "static",
      "ip": "10.66.66.1",
      "netmask": "/24"
    }
  },
  "dns_servers": [
    "8.8.8.8",
    "9.9.9.9"
  ],
  "port_config": {
    "ge-0/0/0": {
      "name": "N0-INET",
      "usage": "wan",
      "aggregated": false,
      "redundant": false,
      "critical": false,
      "disabled": false,
      "wan_type": "broadband",
      "ip_config": {
        "type": "static",
        "ip": "192.168.191.254",
        "netmask": "/24",
        "gateway": "192.168.191.1"
      },
      "wan_ext_ip": "192.168.129.191",
      "disable_autoneg": false,
      "wan_source_nat": {
        "disabled": false
      },
      "vpn_paths": {
        "hahub-N0-INET.OrgOverlay": {
          "role": "hub"
        }
      }
    },
    "ge-0/0/1": {
      "name": "N0-MPLS",
      "usage": "wan",
      "aggregated": false,
      "redundant": false,
      "critical": false,
      "disabled": false,
```

```
    "wan_type": "broadband",
    "ip_config": {
      "type": "static",
      "ip": "192.168.190.254",
      "netmask": "/24",
      "gateway": "192.168.190.1"
    },
    "disable_autoneg": false,
    "wan_source_nat": {
      "disabled": false
    },
    "vpn_paths": {
      "hahub-N0-MPLS.OrgOverlay": {
        "role": "hub"
      }
    }
  },
  "ge-1/0/0": {
    "name": "N1-INET",
    "usage": "wan",
    "aggregated": false,
    "redundant": false,
    "critical": false,
    "disabled": false,
    "wan_type": "broadband",
    "ip_config": {
      "type": "static",
      "ip": "192.168.201.254",
      "netmask": "/24",
      "gateway": "192.168.201.1"
    },
    "wan_ext_ip": "192.168.129.201",
    "disable_autoneg": false,
    "wan_source_nat": {
      "disabled": false
    },
    "vpn_paths": {
      "hahub-N1-INET.OrgOverlay": {
        "role": "hub"
      }
    }
  },
  "ge-1/0/1": {
```

```
      "name": "N1-MPLS",
      "usage": "wan",
      "aggregated": false,
      "redundant": false,
      "critical": false,
      "disabled": false,
      "wan_type": "broadband",
      "ip_config": {
        "type": "static",
        "ip": "192.168.200.254",
        "netmask": "/24",
        "gateway": "192.168.200.1"
      },
      "disable_autoneg": false,
      "wan_source_nat": {
        "disabled": false
      },
      "vpn_paths": {
        "hahub-N1-MPLS.OrgOverlay": {
          "role": "hub"
        }
      }
    },
    "ge-0/0/3,ge-1/0/3": {
      "networks": [
        "HAHUB-LAN1"
      ],
      "usage": "lan",
      "aggregated": false,
      "redundant": true,
      "reth_idx": 3,
      "reth_node": "node0",
      "critical": false,
      "disabled": false
    }
  },
  "bgp_config": {},
  "routing_policies": {},
  "extra_routes": {},
  "path_preferences": {
    "HUB-LANS": {
      "strategy": "ordered",
      "paths": [
```

```
        {
          "type": "local",
          "networks": [
            "HAHUB-LAN1"
          ]
        }
      ]
    },
    "CBO": {
      "strategy": "ordered",
      "paths": [
        {
          "name": "N0-INET",
          "type": "wan"
        },
        {
          "name": "N1-INET",
          "type": "wan"
        }
      ]
    }
  },
  "ospf_areas": {},
  "vrf_instances": {},
  "tunnel_configs": {},
  "oob_ip_config": {
    "type": "dhcp",
    "node1": {
      "type": "dhcp"
    }
  },
  "tunnel_provider_options": {
    "jse": {},
    "zscaler": {}
  },
  "ospf_config": {
    "enabled": false,
    "areas": {}
  },
  "name": "hahub",
  "type": "gateway"
}
```

Should you decide to configure everything manually in the Juniper Mist portal, then use the following steps.

Create a new hub profile with the name “hahub”. Do not change this name as it has dependencies later.

Configure a first **WAN interface** for Node0 as follows:

- Name=`N0-INET` this indicates which topology and node it’s going to use.
- WAN Type=`Ethernet`
- Interface=`ge-0/0/0` as all interfaces starting with `ge-0` are on node0.
- IP Address=`192.168.191.254`
- Prefix Length=`24`
- Gateway=`192.168.191.1`
- Source NAT=`Interface`
- Override for Public IP=`Checked/Enabled`
- Public IP=`192.168.129.191`
- The Overlay Hub Endpoint will be automatically generated and should be “hahub-N0-INET”.

Configure a second **WAN interface** for Node0 as follows:

- Name=`N0-MPLS` this indicates which topology and node it’s going to use.
- WAN Type=`Ethernet`
- Interface=`ge-0/0/1` as all interfaces starting with `ge-0` are on node0.
- IP Address=`192.168.190.254`
- Prefix Length=`24`
- Gateway=`192.168.190.1`
- Source NAT=`Interface`
- Public IP=`192.168.190.254` (auto inserted)
- The Overlay Hub Endpoint will be automatically generated and should be “hahub-N0-MPLS”.

Configure a first **WAN interface** for Node1 as follows:

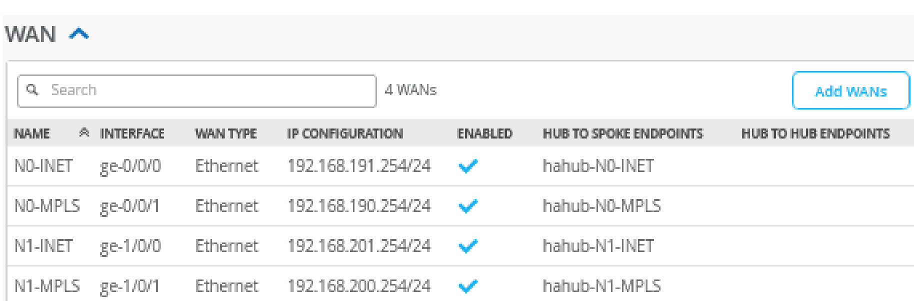
- Name=`N1-INET` this indicates which topology and node it’s going to use.

- WAN Type=`Ethernet`
- Interface=`ge-1/0/0` as all interfaces starting with `ge-1` are on node1.
- IP Address=`192.168.201.254`
- Prefix Length=`24`
- Gateway=`192.168.201.1`
- Source NAT=`Interface`
- Override for Public IP=`Checked/Enabled`
- Public IP=`192.168.129.201`
- The Overlay Hub Endpoint will be automatically generated and should be “hahub-N1-INET”.

Configure a second **WAN interface** for Node1 as follows:

- Name=`N1-MPLS` this indicates which Topology and Node it’s going to use.
- WAN Type=`Ethernet`
- Interface=`ge-1/0/1` as all interfaces starting with `ge-1` are on node1.
- IP Address=`192.168.200.254`
- Prefix Length=`24`
- Gateway=`192.168.200.1`
- Source NAT=`Interface`
- Public IP=`192.168.200.254` (auto inserted)
- The Overlay Hub Endpoint will be automatically generated and should be “hahub-N1-MPLS”.

The result should look like the figure below:

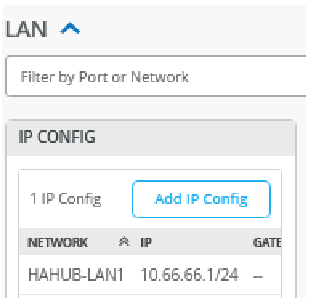


NAME	INTERFACE	WAN TYPE	IP CONFIGURATION	ENABLED	HUB TO SPOKE ENDPOINTS	HUB TO HUB ENDPOINTS
N0-INET	ge-0/0/0	Ethernet	192.168.191.254/24	✓	hahub-N0-INET	
N0-MPLS	ge-0/0/1	Ethernet	192.168.190.254/24	✓	hahub-N0-MPLS	
N1-INET	ge-1/0/0	Ethernet	192.168.201.254/24	✓	hahub-N1-INET	
N1-MPLS	ge-1/0/1	Ethernet	192.168.200.254/24	✓	hahub-N1-MPLS	

Add a LAN IP config now with the following configuration:

- Network=`HAHUB-LAN1`
- IP Address=`10.66.66.1`
- Prefix Length=`24`

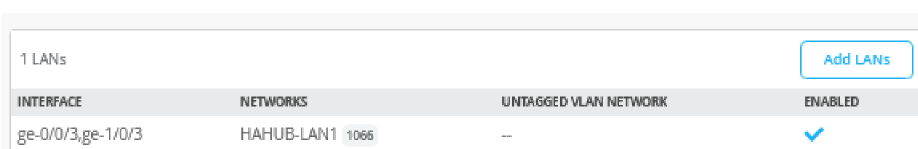
The result should look like the figure below:



Add a LAN interface now with the following configuration:

- Interface=`ge-0/0/3`, `ge-1/0/3`
- Redundant=`Checked/Enabled`
  - Redundant Index=`3` (this is not required for an SSR, but we add it for compatibility)
  - Primary Node=`node0`
- Networks=`HAHUB-LAN1`
- Untagged VLAN=`None`

The result should look like the figure below:



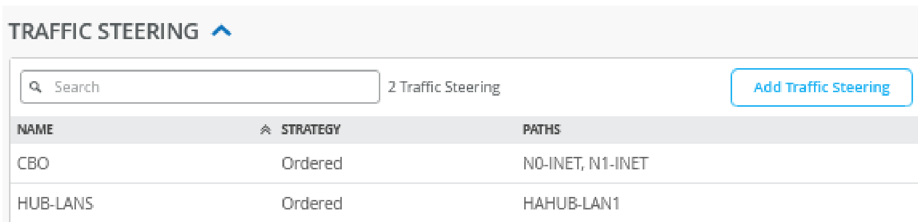
Now we need to define two traffic steering rules. The first rule has the following configuration:

- Name=`HUB-LANS`
- Strategy=`Ordered`
- Paths
  - Path1 Type=`LAN: HAHUB1-LAN1`

The second rule has the following configuration:

- Name=`CBO`
- Strategy=`Ordered`
- Paths
    - Path1 Type=`WAN: N0-INET`
    - Path2 Type=`WAN: N1-INET`

The result should look like the figure below:

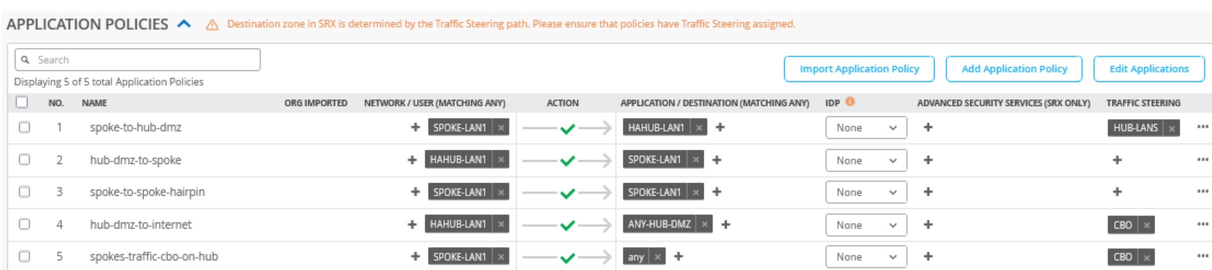


Configure or import the following application policies:

- Number=`1`
    - Name=`spoke-to-hub-dmz`
    - Network=`SPOKE-LAN1`
    - Action=`Pass`
    - Application=`HAHUB-LAN1`
    - Traffic Steering=`HUB-LANS`
- Number=`2`
    - Name= `hub-dmz-to-spoke`
    - Network=`HAHUB-LAN1`
    - Action=`Pass`
    - Application=`SPOKE-LAN1`
    - Traffic Steering=`N/A`
- Number=`3`
    - Name= `spoke-to-spoke-hairpin`
    - Network=`SPOKE-LAN1`

    - Action=`Pass`
    - Application=`SPOKE-LAN1`
    - Traffic Steering=`N/A`
- Number=`4`
    - Name=`hub-dmz-to-internet`
    - Network=`HAHUB-LAN1`
    - Action=`Pass`
    - Application=`ANY-HUB-DMZ`
    - Traffic Steering=`CBO`
- Number=`4`
    - Name= `spokes-traffic-cbo-on-hub`
    - Network=`SPOKE-LAN1`
    - Action=`Pass`
    - Application=`any`
    - Traffic Steering=`CBO`

The result should look like the figure below:



Save your results.

Create a WAN Edge Template for a Single Spoke

Go to **Organization -> WAN Edge Templates.**

Should you choose to use the import option, click on **Import Profile** and import the below JSON as a file.


```
{
  "type": "spoke",
  "dhcpd_config": {
    "enabled": true,
    "SPOKE-LAN1": {
      "type": "local",
      "ip_start": "{{SPOKE_LAN1_PFX}}.10",
      "ip_end": "{{SPOKE_LAN1_PFX}}.250",
      "gateway": "{{SPOKE_LAN1_PFX}}.1",
      "dns_servers": [
        "8.8.8.8",
        "9.9.9.9"
      ],
      "options": {},
      "lease_time": 86400,
      "fixed_bindings": {}
    }
  },
  "ntpOverride": true,
  "dnsOverride": true,
  "service_policies": [
    {
      "name": "spoke-to-hub-dmz",
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "HAHUB-LAN1"
      ],
      "action": "allow",
      "idp": {
        "enabled": false
      },
      "path_preference": "VPN"
    },
    {
      "name": "hub-dmz-to-spoke",
      "tenants": [
        "HAHUB-LAN1"
      ],
```

```
      "services": [
        "SPOKE-LAN1"
      ],
      "action": "allow",
      "path_preference": "LAN",
      "idp": {
        "enabled": false
      }
    },
    {
      "name": "spoke-to-spoke-via-hub",
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "SPOKE-LAN1"
      ],
      "action": "allow",
      "idp": {
        "enabled": false
      },
      "local_routing": true
    },
    {
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "any"
      ],
      "action": "allow",
      "name": "internet-via-hub-cbo",
      "idp": {
        "enabled": false
      },
      "path_preference": "VPN"
    }
  ],
  "ip_configs": {
    "SPOKE-LAN1": {
      "type": "static",
      "ip": "{{SPOKE_LAN1_PFX}}.1",
      "netmask": "/24"
```

```
    }
  },
  "dns_servers": [
    "8.8.8.8",
    "9.9.9.9"
  ],
  "port_config": {
    "ge-0/0/0": {
      "name": "INET",
      "usage": "wan",
      "aggregated": false,
      "redundant": false,
      "critical": false,
      "disabled": false,
      "wan_type": "broadband",
      "ip_config": {
        "type": "dhcp"
      },
      "disable_autoneg": false,
      "wan_source_nat": {
        "disabled": false
      },
      "vpn_paths": {
        "hahub-N0-INET.OrgOverlay": {
          "role": "spoke",
          "bfd_profile": "broadband"
        },
        "hahub-N1-INET.OrgOverlay": {
          "role": "spoke",
          "bfd_profile": "broadband"
        }
      }
    },
    "ge-0/0/1": {
      "name": "MPLS",
      "usage": "wan",
      "aggregated": false,
      "redundant": false,
      "critical": false,
      "disabled": false,
      "wan_type": "broadband",
      "ip_config": {
        "type": "static",
```

```
        "ip": "{{WAN1_PFX}}.2",
        "netmask": "/24",
        "gateway": "{{WAN1_PFX}}.1"
      },
      "disable_autoneg": false,
      "wan_source_nat": {
        "disabled": false
      },
      "vpn_paths": {
        "hahub-N0-MPLS.OrgOverlay": {
          "bfd_profile": "broadband",
          "role": "spoke",
          "key": 0
        },
        "hahub-N1-MPLS.OrgOverlay": {
          "bfd_profile": "broadband",
          "role": "spoke",
          "key": 1
        }
      }
    },
    "ge-0/0/3": {
      "usage": "lan",
      "networks": [
        "SPOKE-LAN1"
      ]
    }
  },
  "bgp_config": {},
  "routing_policies": {},
  "extra_routes": {},
  "path_preferences": {
    "LAN": {
      "strategy": "ordered",
      "paths": [
        {
          "type": "local",
          "networks": [
            "SPOKE-LAN1"
          ]
        }
      ]
    },
```

```
    "VPN": {
      "strategy": "weighted",
      "paths": [
        {
          "name": "hahub-N0-INET.OrgOverlay",
          "cost": 10,
          "type": "vpn"
        },
        {
          "name": "hahub-N1-INET.OrgOverlay",
          "cost": 20,
          "type": "vpn"
        },
        {
          "name": "hahub-N0-MPLS.OrgOverlay",
          "cost": 30,
          "type": "vpn"
        },
        {
          "name": "hahub-N1-MPLS.OrgOverlay",
          "cost": 40,
          "type": "vpn"
        }
      ]
    }
  },
  "ospf_areas": {},
  "vrf_instances": {},
  "tunnel_configs": {},
  "oob_ip_config": {
    "type": "dhcp",
    "node1": {
      "type": "dhcp"
    }
  },
  "tunnel_provider_options": {
    "jse": {},
    "zscaler": {}
  },
  "ospf_config": {
    "enabled": false,
    "areas": {}
  },
```

```
  "name": "single-spoke"
}
```

Should you decide to configure everything manually in the Juniper Mist portal, then use the following steps.

We recommend you clone the existing “Spokes” template and name the new template “single-spoke”. Then make the following changes as the endpoint definitions and hub networks change with the new hub profile:

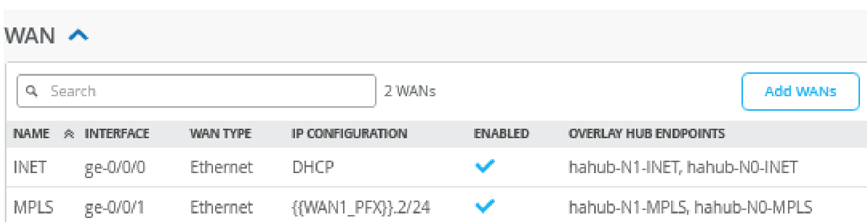
Edit a first **WAN interface** with the name “INET” as follows:

- Name=`INET` this indicates which Topology it’s going to use.
  - Endpoint1=`hahub-N0-INET`
  - Endpoint2=`hahub-N1-INET`

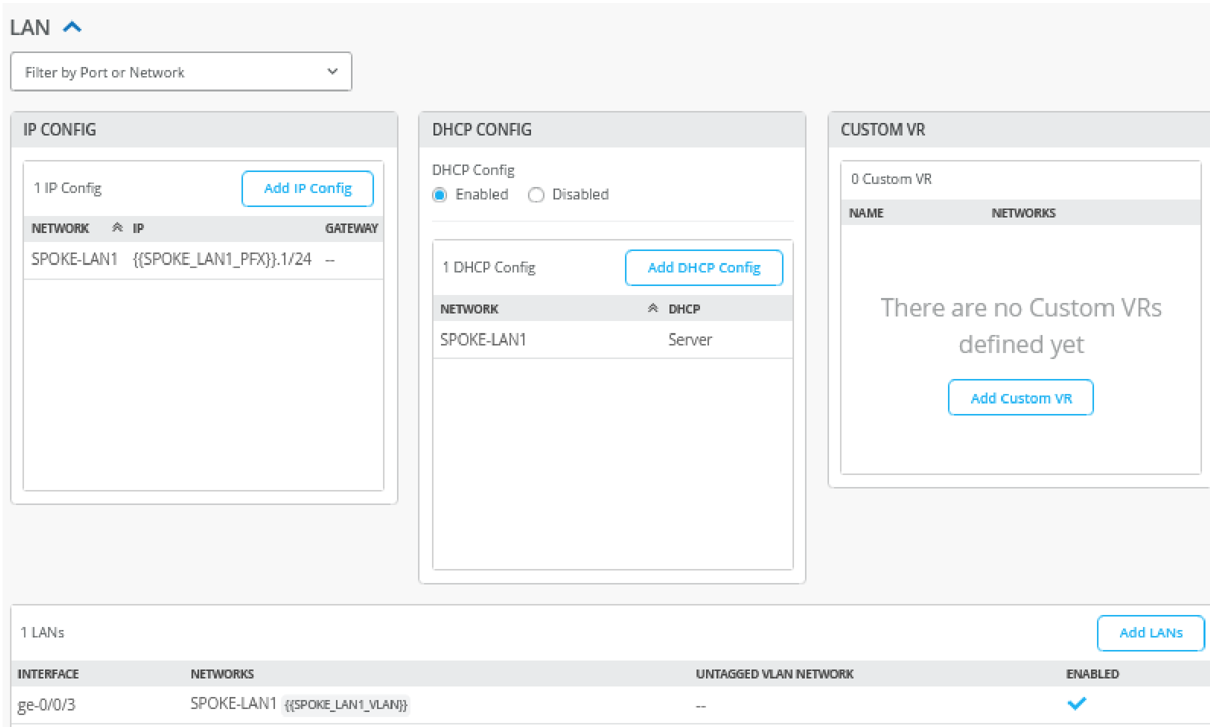
Edit a second **WAN interface** with the name “MPLS” as follows:

- Name=`INET` this indicates which Topology it’s going to use.
  - Endpoint1=`hahub-N0-MPLS`
  - Endpoint2=`hahub-N1-MPLS`

The result should look like the figure below:



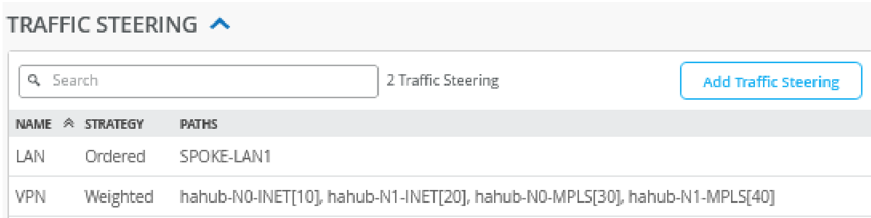
The LAN sections do not need to be changed and should still look like the figure below:



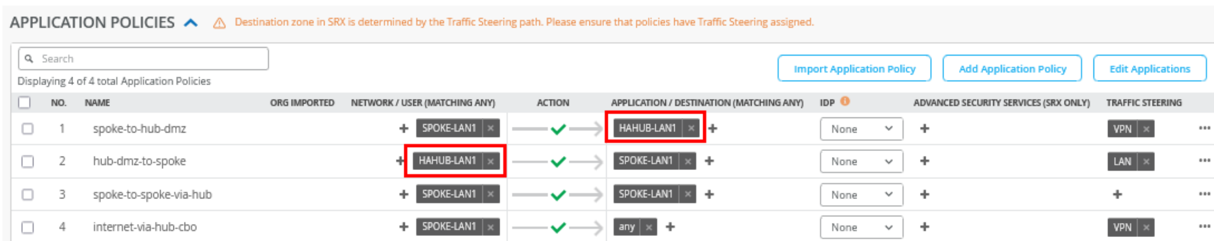
In the traffic steering section, we need to edit the end points of the “VPN” profile. Please apply the following configuration:

- Name=`VPN`
- Strategy=`Weighted`
- Paths
  - Path1 Type=`Overlay: hahub-N0-INET`
  - Path1 Cost=`10`
  - Path2 Type=`Overlay: hahub-N1-INET`
  - Path2 Cost=`20`
  - Path3 Type=`Overlay: hahub-N0-MPLS`
  - Path3 Cost=`30`
  - Path4 Type=`Overlay: hahub-N1-MPLS`
  - Path4 Cost=`40`

The result should look like the figure below:



In application policies, we need to change the prior `HUB1-LAN1` and `HUB2-LAN1` definitions to a single `HAHUB-LAN1` as indicated in the figure below:



Save your results.

Create a WAN Edge Template for a High-Availability Spoke

Go to **Organization -> WAN Edge Templates.**

Should you choose to use the import option, click on **Import Profile** and import the below JSON as a file.


```
{
  "type": "spoke",
  "dhcpd_config": {
    "enabled": true,
    "SPOKE-LAN1": {
      "type": "local",
      "ip_start": "{{SPOKE_LAN1_PFX}}.10",
      "ip_end": "{{SPOKE_LAN1_PFX}}.250",
      "gateway": "{{SPOKE_LAN1_PFX}}.1",
      "dns_servers": [
        "8.8.8.8",
        "9.9.9.9"
      ],
      "options": {},
      "lease_time": 86400,
      "fixed_bindings": {}
```

```
    }
  },
  "ntpOverride": true,
  "dnsOverride": true,
  "service_policies": [
    {
      "name": "spoke-to-hub-dmz",
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "HAHUB-LAN1"
      ],
      "action": "allow",
      "idp": {
        "enabled": false
      },
      "path_preference": "VPN"
    },
    {
      "name": "hub-dmz-to-spoke",
      "tenants": [
        "HAHUB-LAN1"
      ],
      "services": [
        "SPOKE-LAN1"
      ],
      "action": "allow",
      "path_preference": "LAN",
      "idp": {
        "enabled": false
      }
    },
    {
      "name": "spoke-to-spoke-via-hub",
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "SPOKE-LAN1"
      ],
      "action": "allow",
      "idp": {
```

```
        "enabled": false
      },
      "local_routing": true
    },
    {
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "any"
      ],
      "action": "allow",
      "name": "internet-via-hub-cbo",
      "idp": {
        "enabled": false
      },
      "path_preference": "VPN"
    }
  ],
  "ip_configs": {
    "SPOKE-LAN1": {
      "type": "static",
      "ip": "{{SPOKE_LAN1_PFX}}.1",
      "netmask": "/24"
    }
  },
  "dns_servers": [
    "8.8.8.8",
    "9.9.9.9"
  ],
  "port_config": {
    "ge-0/0/0": {
      "name": "N0-INET",
      "usage": "wan",
      "aggregated": false,
      "redundant": false,
      "critical": false,
      "disabled": false,
      "wan_type": "broadband",
      "ip_config": {
        "type": "dhcp"
      },
      "disable_autoneg": false,
```

```
      "wan_source_nat": {
        "disabled": false
      },
      "vpn_paths": {
        "hahub-N0-INET.OrgOverlay": {
          "role": "spoke",
          "bfd_profile": "broadband"
        },
        "hahub-N1-INET.OrgOverlay": {
          "role": "spoke",
          "bfd_profile": "broadband"
        }
      }
    },
    "ge-1/0/0": {
      "name": "N1-INET",
      "usage": "wan",
      "aggregated": false,
      "redundant": false,
      "critical": false,
      "disabled": false,
      "wan_type": "broadband",
      "ip_config": {
        "type": "dhcp"
      },
      "disable_autoneg": false,
      "wan_source_nat": {
        "disabled": false
      },
      "vpn_paths": {
        "hahub-N0-INET.OrgOverlay": {
          "role": "spoke",
          "bfd_profile": "broadband",
          "key": 0
        },
        "hahub-N1-INET.OrgOverlay": {
          "role": "spoke",
          "bfd_profile": "broadband",
          "key": 1
        }
      }
    },
    "ge-0/0/1,ge-1/0/1": {
```

```
      "name": "HA-MPLS",
      "usage": "wan",
      "aggregated": false,
      "redundant": true,
      "reth_idx": 1,
      "reth_node": "node0",
      "critical": false,
      "disabled": false,
      "wan_type": "broadband",
      "ip_config": {
        "type": "static",
        "ip": "{{WAN1_PFX}}.2",
        "netmask": "/24",
        "gateway": "{{WAN1_PFX}}.1"
      },
      "disable_autoneg": false,
      "wan_source_nat": {
        "disabled": false
      },
      "vpn_paths": {
        "hahub-N0-MPLS.OrgOverlay": {
          "role": "spoke",
          "bfd_profile": "broadband",
          "key": 0
        },
        "hahub-N1-MPLS.OrgOverlay": {
          "role": "spoke",
          "bfd_profile": "broadband",
          "key": 1
        }
      }
    },
    "ge-0/0/3,ge-1/0/3": {
      "networks": [
        "SPOKE-LAN1"
      ],
      "usage": "lan",
      "aggregated": false,
      "redundant": true,
      "reth_idx": 3,
      "reth_node": "node0",
      "critical": false,
      "disabled": false
```

```
    }
  },
  "bgp_config": {},
  "routing_policies": {},
  "extra_routes": {},
  "path_preferences": {
    "LAN": {
      "strategy": "ordered",
      "paths": [
        {
          "type": "local",
          "networks": [
            "SPOKE-LAN1"
          ]
        }
      ]
    },
    "VPN": {
      "strategy": "weighted",
      "paths": [
        {
          "name": "hahub-N0-INET.OrgOverlay",
          "cost": 10,
          "type": "vpn"
        },
        {
          "name": "hahub-N1-INET.OrgOverlay",
          "cost": 20,
          "type": "vpn"
        },
        {
          "name": "hahub-N0-MPLS.OrgOverlay",
          "cost": 30,
          "type": "vpn"
        },
        {
          "name": "hahub-N1-MPLS.OrgOverlay",
          "cost": 40,
          "type": "vpn"
        }
      ]
    }
  },
```

```
  "ospf_areas": {},
  "vrf_instances": {},
  "tunnel_configs": {},
  "oob_ip_config": {
    "type": "dhcp",
    "node1": {
      "type": "dhcp"
    }
  },
  "tunnel_provider_options": {
    "jse": {},
    "zscaler": {}
  },
  "ospf_config": {
    "enabled": false,
    "areas": {}
  },
  "name": "ha-spoke"
}
```


Should you decide to configure everything manually in the Juniper Mist portal, then use the following steps.

We recommend you clone the existing "single-spoke" template and name the new template "ha-spoke". Then make the following changes as the WAN and LAN interfaces change with the high-availability configuration.

Delete all prior WAN interfaces and add the following three new WAN interfaces.

Configure a first WAN interface for Node0 as follows:

- Name=`N0-INET` this indicates which topology and node it's going to use.
- WAN Type=`Ethernet`
- Interface=`ge-0/0/0` as all interfaces starting with `ge-0` are on node0.
- IP Configuration=`DHCP`
- Source NAT=`Interface`
- Overlay Hub Endpoints
  - Endpoint1=`hahub-N0-INET`
  - BFD Profile1=`Broadband`

    - Endpoint2=`hahub-N1-INET`
    - BFD Profile2=`Broadband`

Then configure a first WAN interface for Node1 as follows:

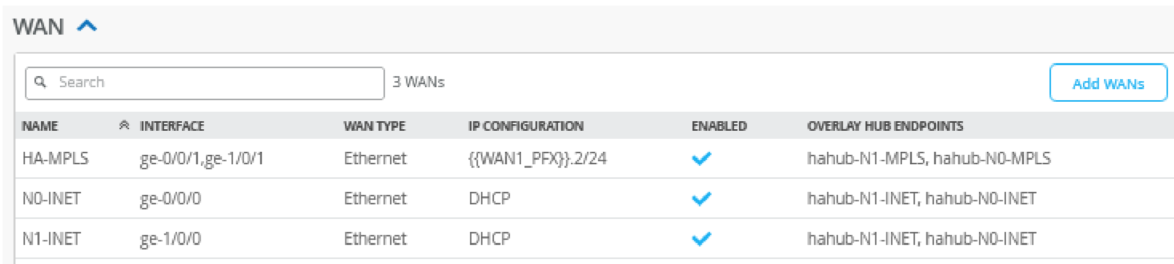
- Name=`N1-INET` this indicates which Topology and Node it's going to use.
- WAN Type=`Ethernet`
- Interface=`ge-1/0/0` as all interfaces starting with `ge-1` are on node1.
- IP Configuration=`DHCP`
- Source NAT=`Interface`
- Overlay Hub Endpoints
    - Endpoint1=`hahub-N0-INET`
    - BFD Profile1=`Broadband`
    - Endpoint2=`hahub-N1-INET`
    - BFD Profile2=`Broadband`

Then configure the third and redundant WAN interface for the MPLS path for Node0 and Node1 as follows:

- Name=`HA-MPLS` this indicates which topology and node it's going to use.
- WAN Type=`Ethernet`
- Interface=`ge-0/0/1,ge-1/0/1`
    - Redundant=`Checked/Enabled`
    - Redundant Index=1 (this is not required for an SSR, but we add it for compatibility)
    - Primary Node=`node0`
- IP Configuration=`Static`
- `IP Address={{WAN1_PFX}}.2`
- `Prefix Length=24`
- `Gateway={{WAN1_PFX}}.1`
- Source NAT=`Interface`

- Overlay Hub Endpoints
  - Endpoint1=`hahub-N0-MPLS`
  - BFD Profile1=`Broadband`
  - Endpoint2=`hahub-N1-MPLS`
  - BFD Profile2=`Broadband`

The result should look like the figure below:

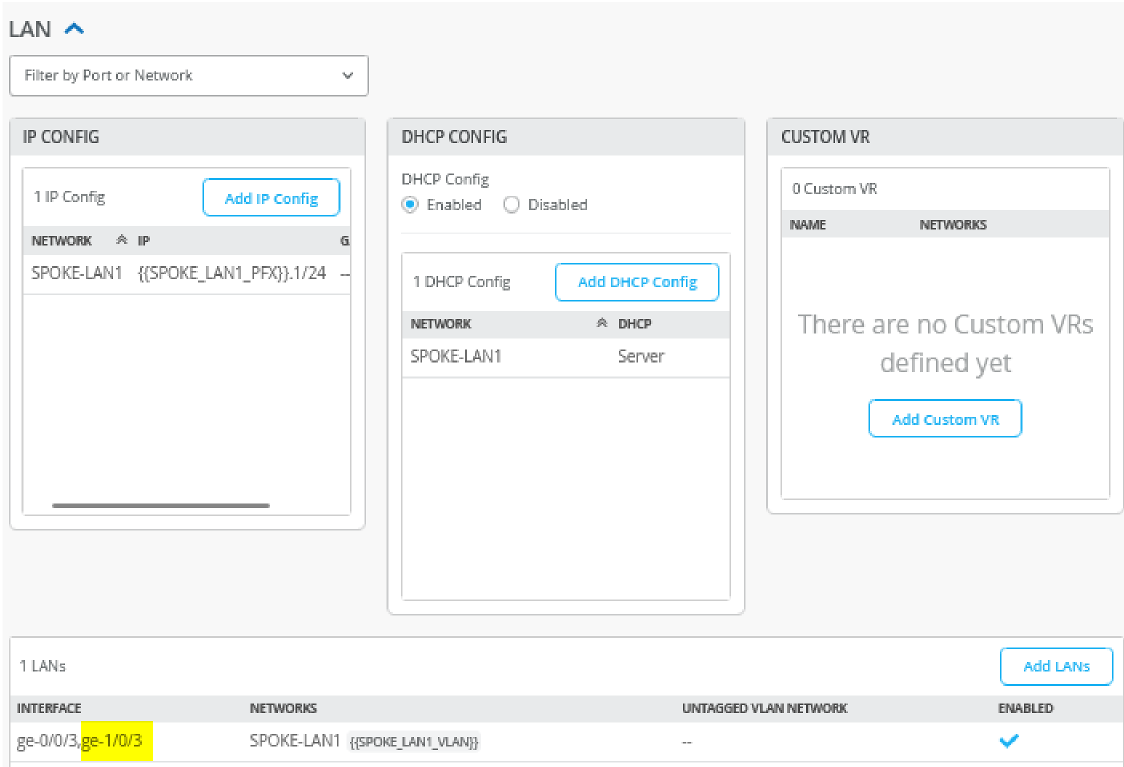


NAME	INTERFACE	WAN TYPE	IP CONFIGURATION	ENABLED	OVERLAY HUB ENDPOINTS
HA-MPLS	ge-0/0/1,ge-1/0/1	Ethernet	{{WAN1_PFX}}.2/24	✓	hahub-N1-MPLS, hahub-N0-MPLS
N0-INET	ge-0/0/0	Ethernet	DHCP	✓	hahub-N1-INET, hahub-N0-INET
N1-INET	ge-1/0/0	Ethernet	DHCP	✓	hahub-N1-INET, hahub-N0-INET

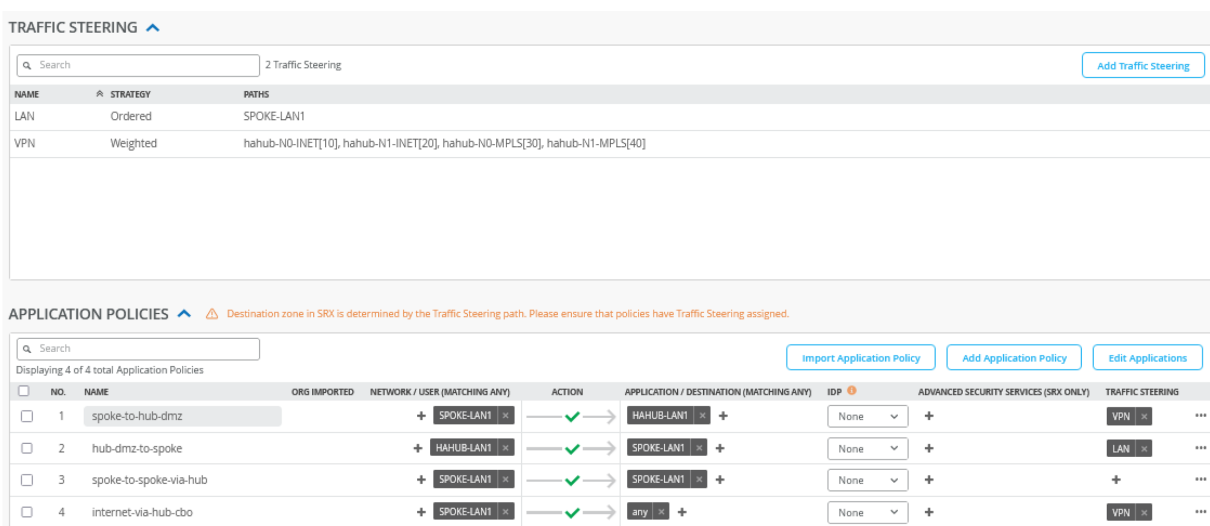
In the LAN section, the IP config and DHCP config stay the same inherited from the previous templates. The LAN interface itself needs to be edited to support the redundant configuration. Please change the existing configuration so that you have the following:

- Interface=`ge-0/0/3`, `ge-1/0/3`
- Redundant=Checked/Enabled
  - Redundant Index=3 (this is not required for an SSR, but we add it for compatibility)
  - Primary Node=node0
- Networks=`SPOKE-LAN1`
- Untagged VLAN=`None`

The result should look like the figure below:



Traffic steering and application profiles should be inherited from the already existing “single-spoke” template, so they do not need to be changed. They should look like the figure below:



Save your results.

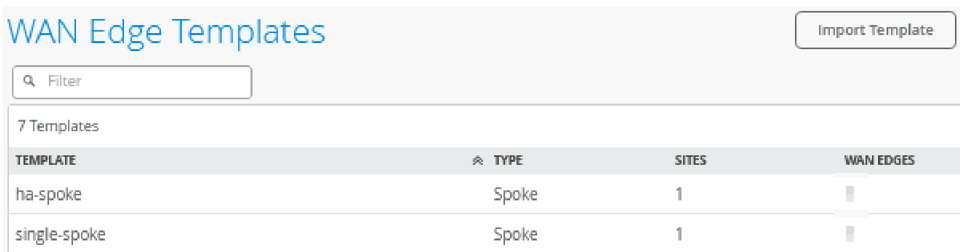
Remaining Tasks for This Lab

First you need to assign the new spoke templates to their sites.

Go to **Organization -> WAN Edge Templates -> "single-spoke"** template and click on **Assign to Sites**. Then, assign spoke1-site to this template.

Go to **Organization -> WAN Edge Templates -> "ha-spoke"** template and click on **Assign to Sites**. Then, assign spoke2-site to this template.

The result should look like the figure below:

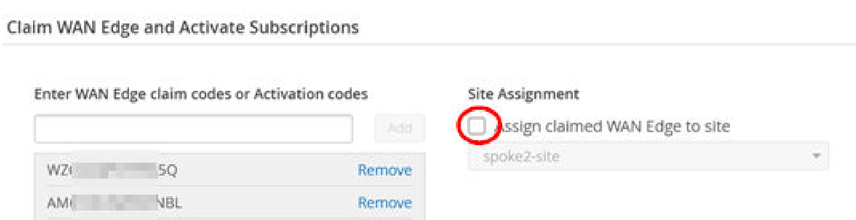


If you have the hub and spoke devices in use from previous labs, go to the inventory and release them. This will bring them back into factory state.

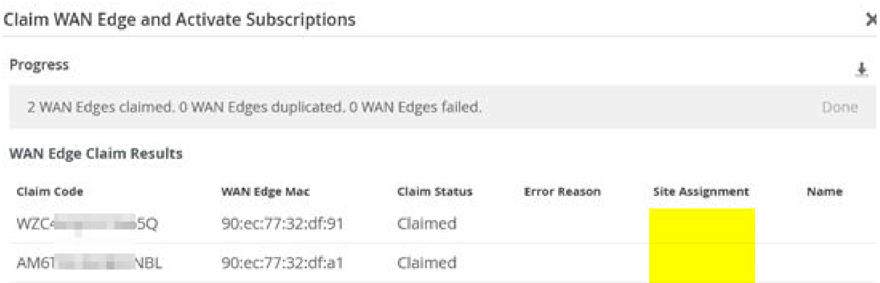
Here, we describe how to build a cluster during the onboarding—assuming all devices are in a factory default state.

Go to **Organization -> Inventory** and select **WAN Edges** and click **Claim WAN Edges**. Then, do the following:

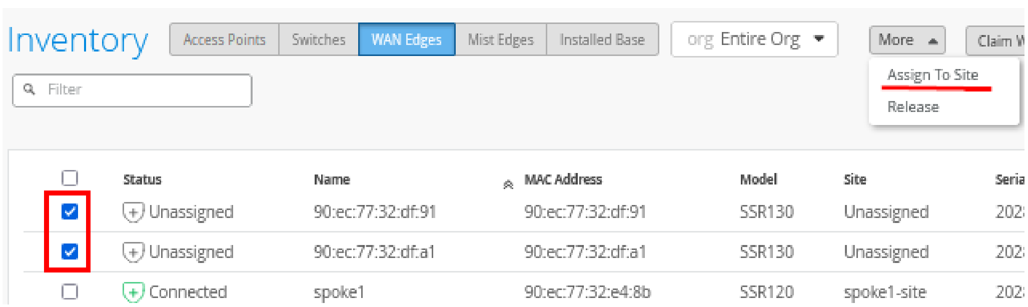
- Assign claimed WAN Edge to site=`Unchecked/Disabled`.
- Enter the claim codes for the two devices.



This will claim the two devices without assigning them to a site.



Select the two devices and click on **Assign To Site** from the **More** menu.



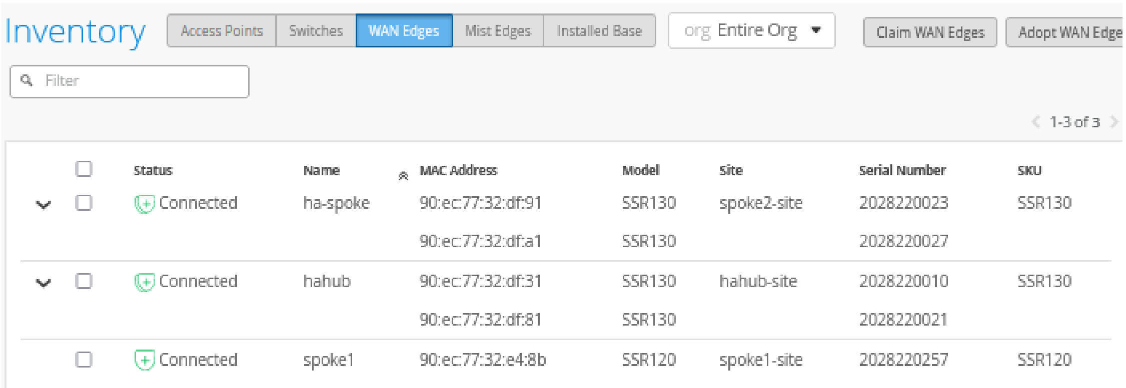
Now configure the following:

- Assign the two selected WAN edges to site=`spoke2-site`
- Create Cluster=`Checked/Enabled`
- Select a device to act as node0 = select that as required
- Manage Configuration with Mist=`Enabled` (automatically)

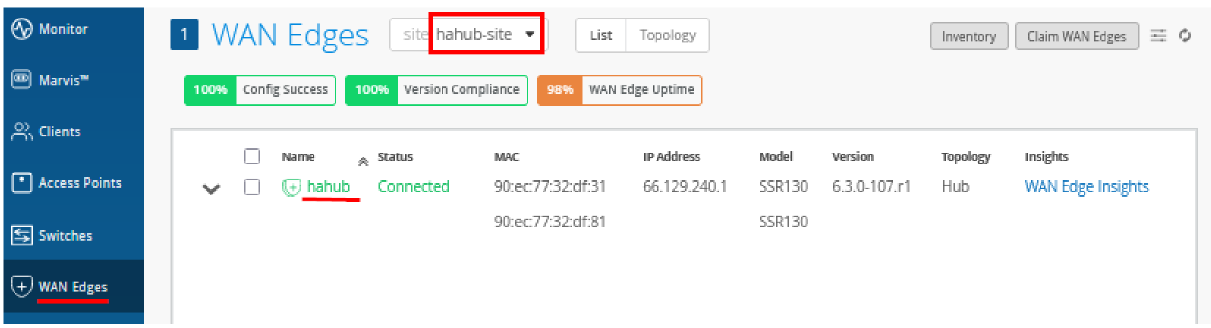
This will commit the needed cluster configuration for the HA spoke.

The hahub follows a similar process that we do not repeat here. You will have to claim two other devices and then use the new site "hahub-site".

After all clusters have been brought up (and you gave them a name) the inventory should look similar to the figure below:

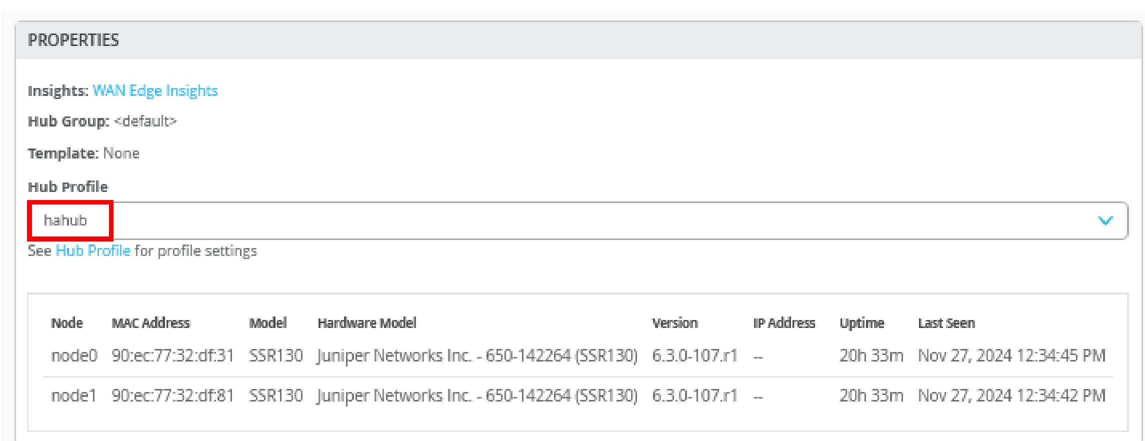


Do not forget to assign the hub profile as the last step. Go to **WAN Edges** and select site "hahub-site" and select the cluster device.



Then, under properties, configure the following:

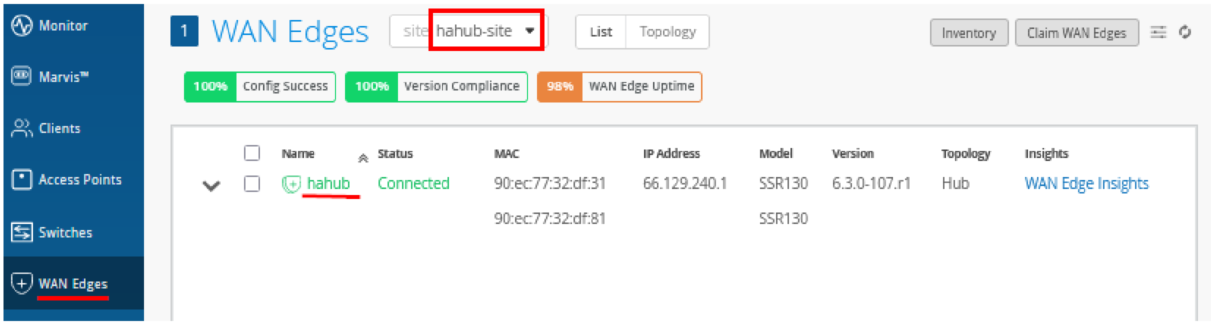
- Hub Profile=hahub



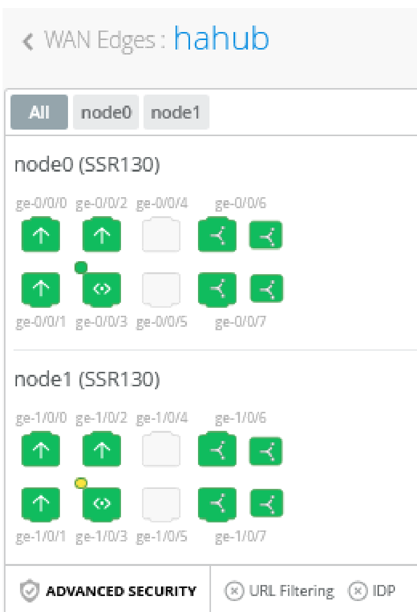
Test Your Network Configuration

We are now ready to test our configuration.

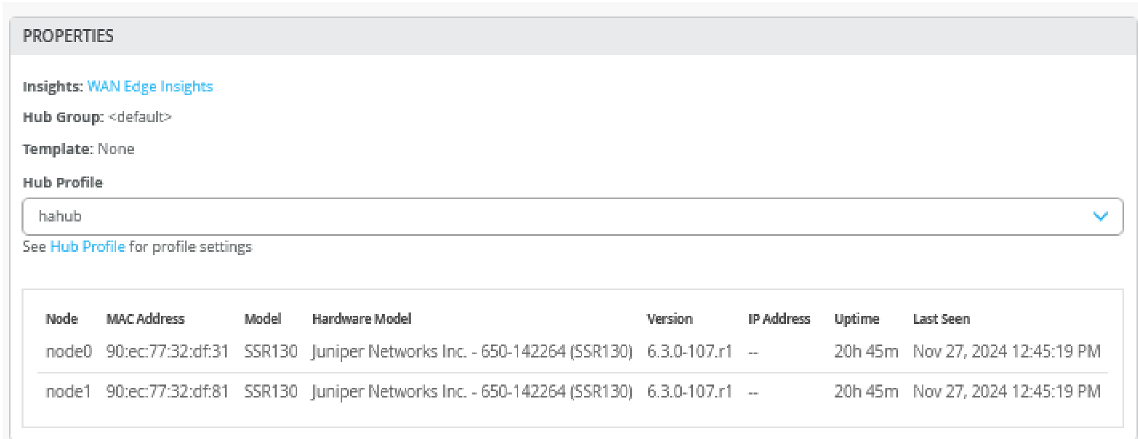
Go to **WAN Edges -> site=hahub-site** and click “hahub”.



Review the device information.

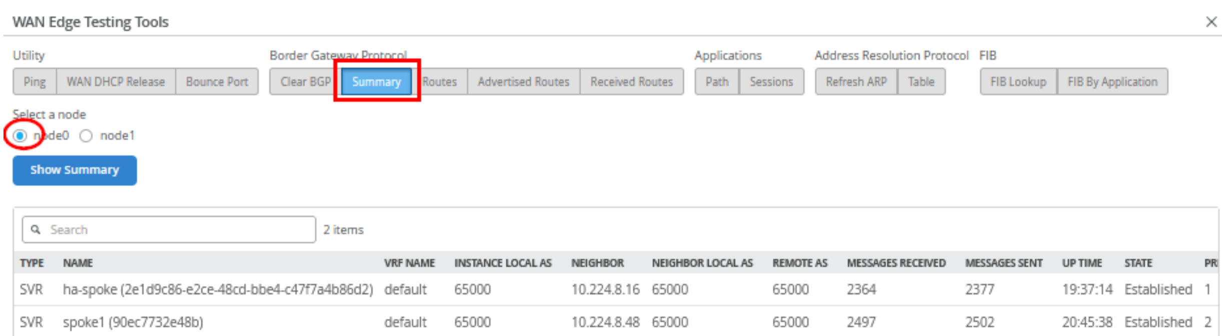


Review the properties information to determine which devices are Node0 and Node1.



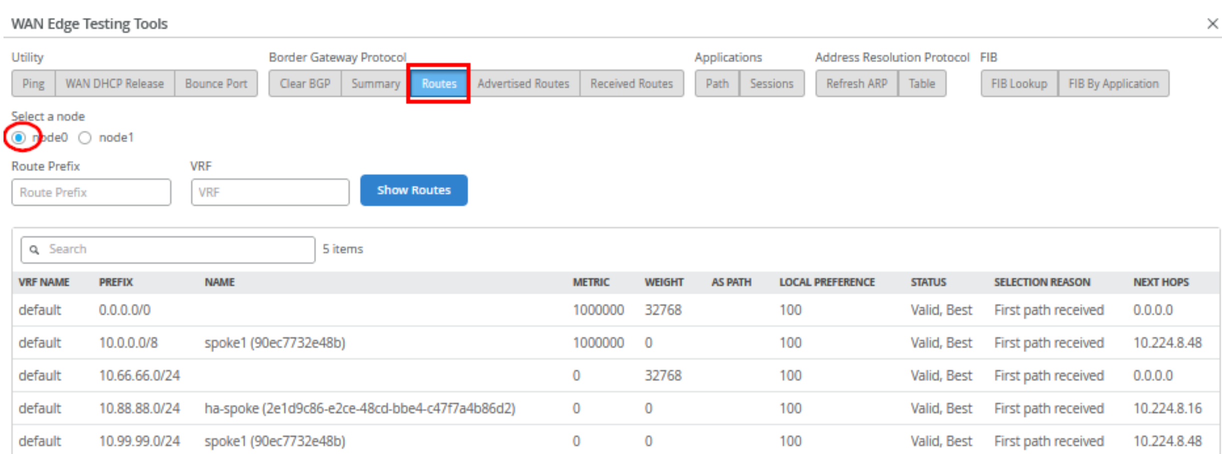
Node	MAC Address	Model	Hardware Model	Version	IP Address	Uptime	Last Seen
node0	90:ec:77:32:df:31	SSR130	Juniper Networks Inc. - 650-142264 (SSR130)	6.3.0-107.r1	--	20h 45m	Nov 27, 2024 12:45:19 PM
node1	90:ec:77:32:df:81	SSR130	Juniper Networks Inc. - 650-142264 (SSR130)	6.3.0-107.r1	--	20h 45m	Nov 27, 2024 12:45:19 PM

When you use **Utilities -> Testing Tools** and review the BGP neighbor summary, you will see two spokes (redundant and non-redundant) connected and exchanging routes.



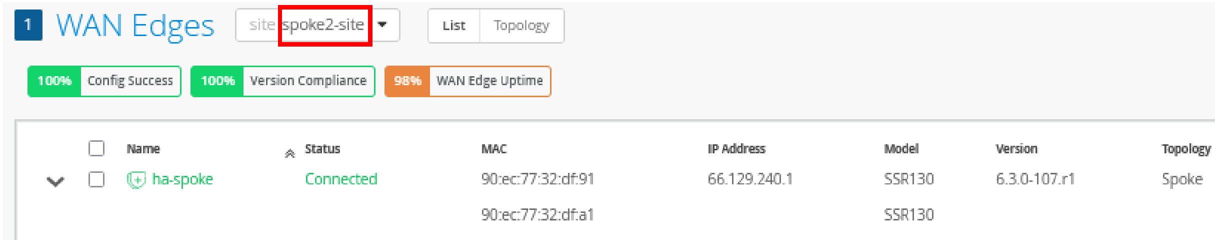
TYPE	NAME	VRF NAME	INSTANCE LOCAL AS	NEIGHBOR	NEIGHBOR LOCAL AS	REMOTE AS	MESSAGES RECEIVED	MESSAGES SENT	UP TIME	STATE	PR
SVR	ha-spoke (2e1d9c86-e2ce-48cd-bbe4-c47f7a4b86d2)	default	65000	10.224.8.16	65000	65000	2364	2377	19:37:14	Established	1
SVR	spoke1 (90ec7732e48b)	default	65000	10.224.8.48	65000	65000	2497	2502	20:45:38	Established	2

Also, review the routes distributed in the VPN.

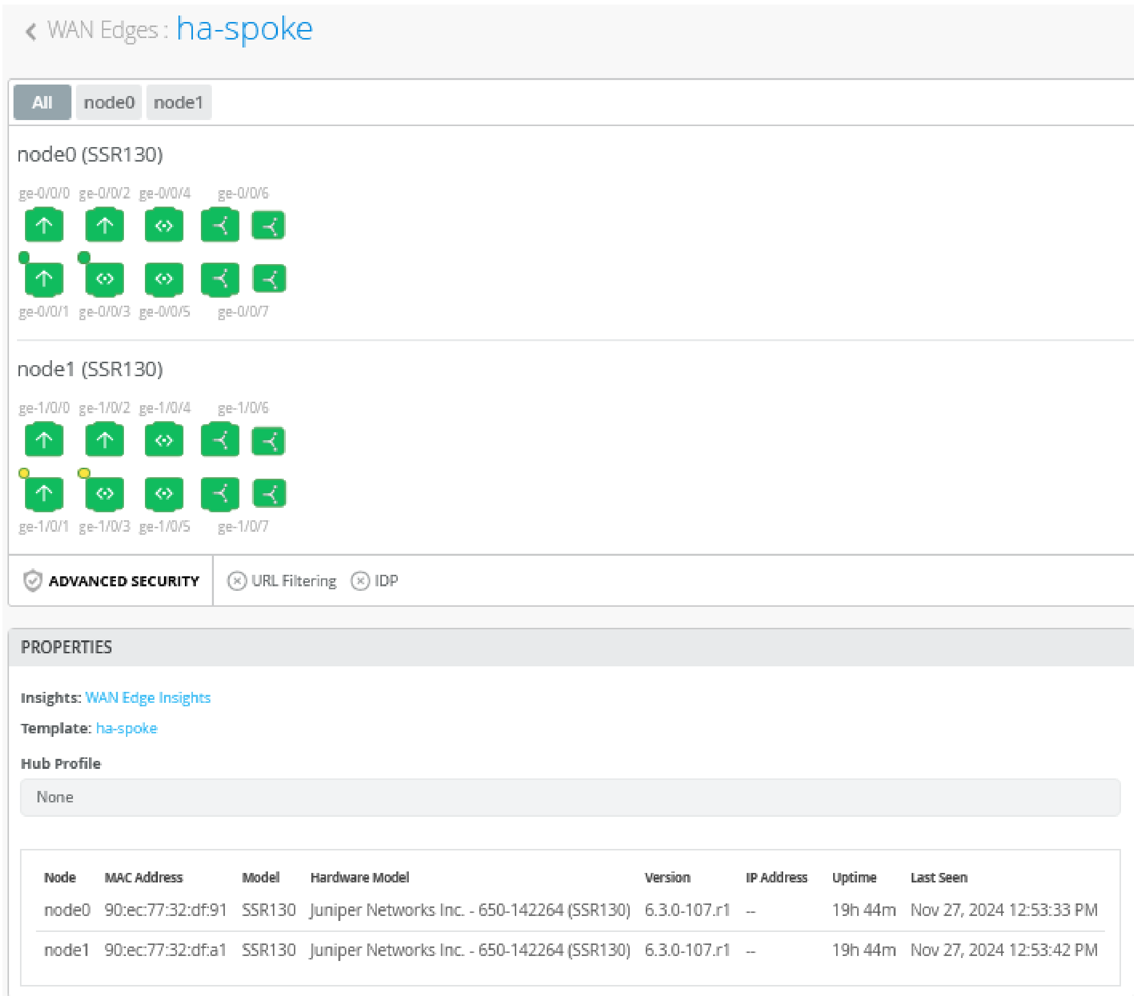


VRF NAME	PREFIX	NAME	METRIC	WEIGHT	AS PATH	LOCAL PREFERENCE	STATUS	SELECTION REASON	NEXT HOPS
default	0.0.0.0/0		1000000	32768		100	Valid, Best	First path received	0.0.0.0
default	10.0.0.0/8	spoke1 (90ec7732e48b)	1000000	0		100	Valid, Best	First path received	10.224.8.48
default	10.66.66.0/24		0	32768		100	Valid, Best	First path received	0.0.0.0
default	10.88.88.0/24	ha-spoke (2e1d9c86-e2ce-48cd-bbe4-c47f7a4b86d2)	0	0		100	Valid, Best	First path received	10.224.8.16
default	10.99.99.0/24	spoke1 (90ec7732e48b)	0	0		100	Valid, Best	First path received	10.224.8.48

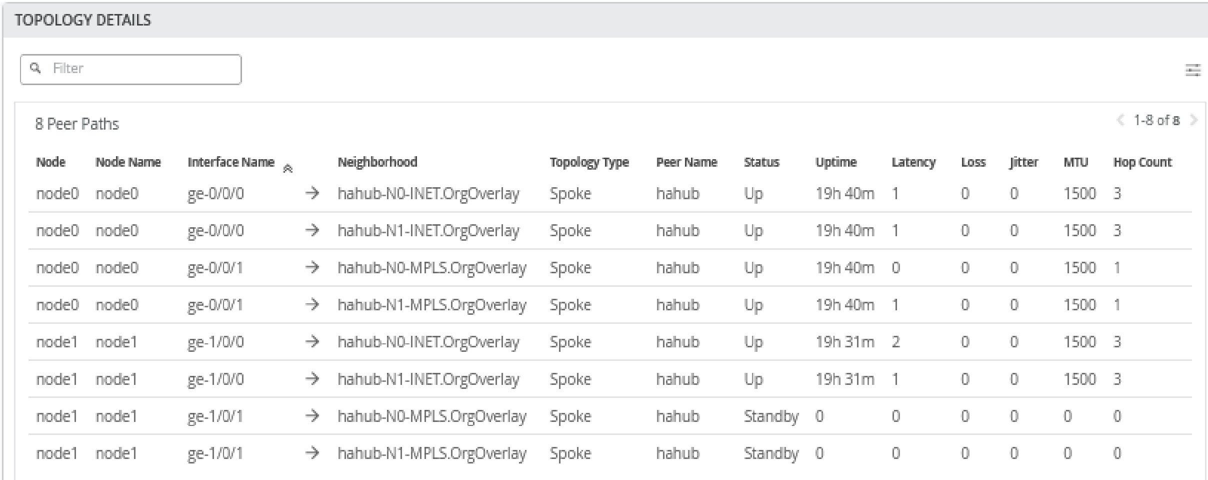
Go to **WAN Edges -> site=spoke2-site** and click on “ha-spoke”.



Review the device and properties information.

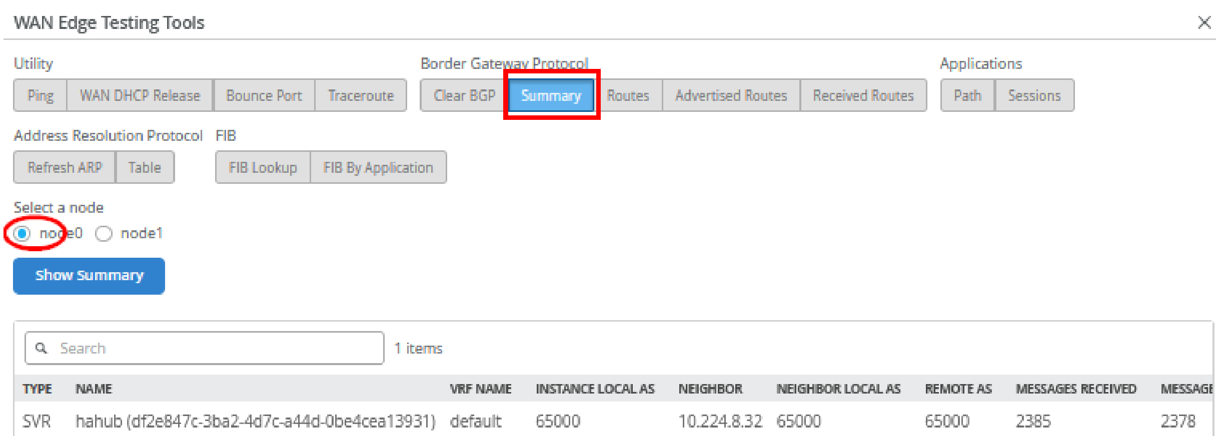


Review the topology details with the eight tunnels connecting this spoke to the HAHub. Two are down as the MPLS interface is configured for active-passive VRRP redundancy.



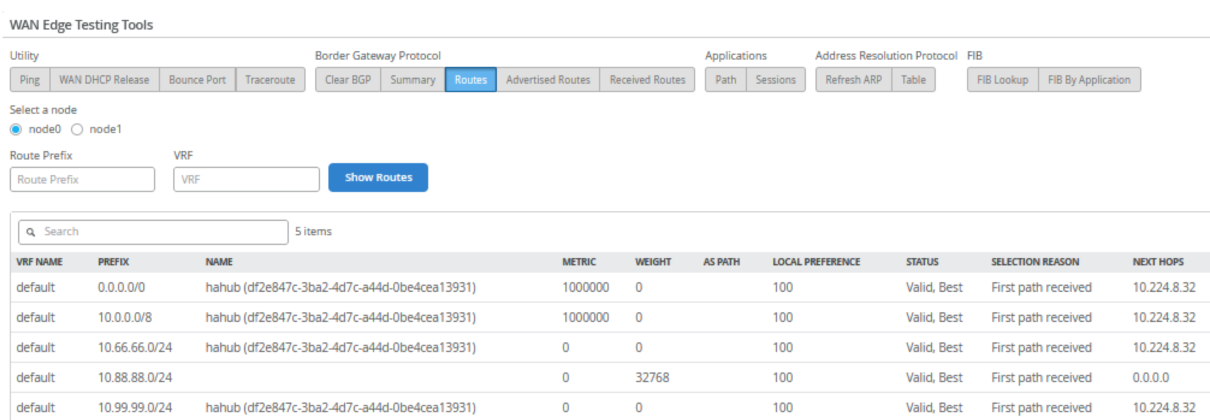
Node	Node Name	Interface Name		Neighborhood	Topology Type	Peer Name	Status	Uptime	Latency	Loss	Jitter	MTU	Hop Count
node0	node0	ge-0/0/0	→	hahub-N0-INET.OrgOverlay	Spoke	hahub	Up	19h 40m	1	0	0	1500	3
node0	node0	ge-0/0/0	→	hahub-N1-INET.OrgOverlay	Spoke	hahub	Up	19h 40m	1	0	0	1500	3
node0	node0	ge-0/0/1	→	hahub-N0-MPLS.OrgOverlay	Spoke	hahub	Up	19h 40m	0	0	0	1500	1
node0	node0	ge-0/0/1	→	hahub-N1-MPLS.OrgOverlay	Spoke	hahub	Up	19h 40m	1	0	0	1500	1
node1	node1	ge-1/0/0	→	hahub-N0-INET.OrgOverlay	Spoke	hahub	Up	19h 31m	2	0	0	1500	3
node1	node1	ge-1/0/0	→	hahub-N1-INET.OrgOverlay	Spoke	hahub	Up	19h 31m	1	0	0	1500	3
node1	node1	ge-1/0/1	→	hahub-N0-MPLS.OrgOverlay	Spoke	hahub	Standby	0	0	0	0	0	0
node1	node1	ge-1/0/1	→	hahub-N1-MPLS.OrgOverlay	Spoke	hahub	Standby	0	0	0	0	0	0

When you use **Utilities -> Testing Tools** and review the BGP neighbor summary, you will only see hahub connected and exchanging routes.



TYPE	NAME	VRF NAME	INSTANCE LOCAL AS	NEIGHBOR	NEIGHBOR LOCAL AS	REMOTE AS	MESSAGES RECEIVED	MESSAGE
SVR	hahub (df2e847c-3ba2-4d7c-a44d-0be4cea13931)	default	65000	10.224.8.32	65000	65000	2385	2378

Also review the routes distributed in the VPN.

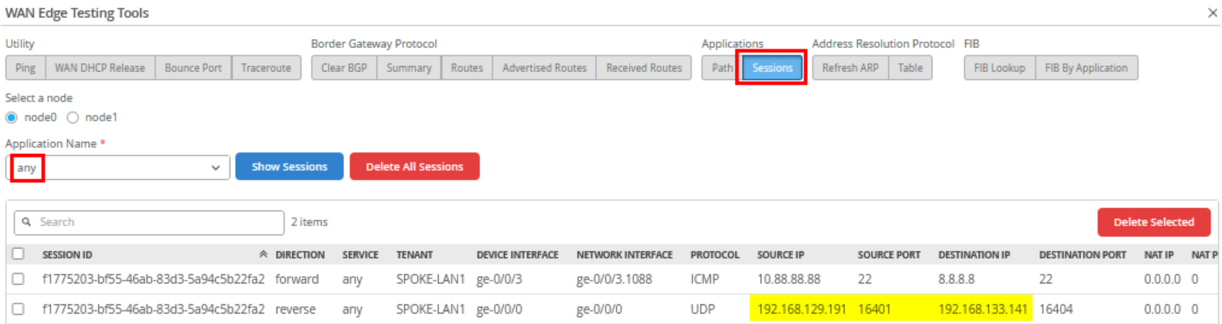


VRF NAME	PREFIX	NAME	METRIC	WEIGHT	AS PATH	LOCAL PREFERENCE	STATUS	SELECTION REASON	NEXT HOPS
default	0.0.0.0/0	hahub (df2e847c-3ba2-4d7c-a44d-0be4cea13931)	1000000	0		100	Valid, Best	First path received	10.224.8.32
default	10.0.0.0/8	hahub (df2e847c-3ba2-4d7c-a44d-0be4cea13931)	1000000	0		100	Valid, Best	First path received	10.224.8.32
default	10.66.66.0/24	hahub (df2e847c-3ba2-4d7c-a44d-0be4cea13931)	0	0		100	Valid, Best	First path received	10.224.8.32
default	10.88.88.0/24		0	32768		100	Valid, Best	First path received	0.0.0.0
default	10.99.99.0/24	hahub (df2e847c-3ba2-4d7c-a44d-0be4cea13931)	0	0		100	Valid, Best	First path received	10.224.8.32

We shall now continue our testing on the clients attached to the spokes. We connect to the desktop2 VM with IP address `10.88.88.88` attached to the redundant spoke2.

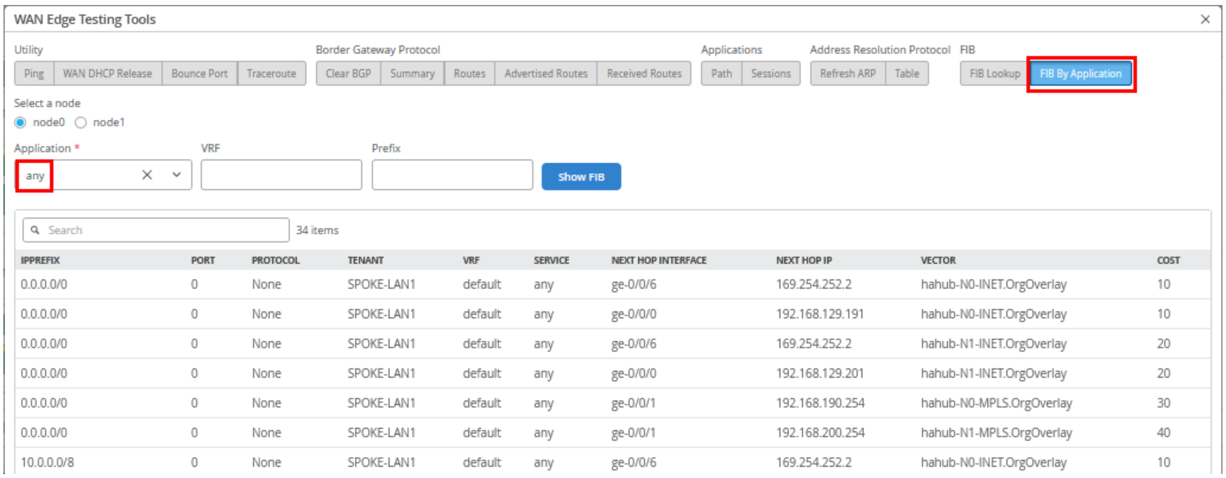
```
# try to reach the local WAN-Router interface desktop2 VM is attached to
root@desktop2:~# ping -c3 10.88.88.1
PING 10.88.88.1 (10.88.88.1) 56(84) bytes of data.
64 bytes from 10.88.88.1: icmp_seq=1 ttl=128 time=0.320 ms
64 bytes from 10.88.88.1: icmp_seq=2 ttl=128 time=0.287 ms
64 bytes from 10.88.88.1: icmp_seq=3 ttl=128 time=0.241 ms
#
# try to reach the client desktop1 VM attached to spoke1
# this causes relay on the hahub for this traffic
root@desktop2:~# ping -c3 10.99.99.99
PING 10.99.99.99 (10.99.99.99) 56(84) bytes of data.
64 bytes from 10.99.99.99: icmp_seq=1 ttl=56 time=7.30 ms
64 bytes from 10.99.99.99: icmp_seq=2 ttl=56 time=1.75 ms
64 bytes from 10.99.99.99: icmp_seq=3 ttl=56 time=1.66 ms
#
# try to reach the client desktop4 VM attached to hahub
root@desktop2:~# ping -c3 10.66.66.66
PING 10.66.66.66 (10.66.66.66) 56(84) bytes of data.
64 bytes from 10.66.66.66: icmp_seq=1 ttl=59 time=3.39 ms
64 bytes from 10.66.66.66: icmp_seq=2 ttl=59 time=1.19 ms
64 bytes from 10.66.66.66: icmp_seq=3 ttl=59 time=0.952 ms
#
# let a continued ping to the internet run
# in our case all traffic is sent to hub for central breakout
root@desktop2:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=46 time=7.93 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=46 time=3.90 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=46 time=3.98 ms
.
.
```

Use **Utilities -> Testing Tools** to review the application sessions with Application Name=`any` . Due to the reverse flow, we see that the traffic is received from Hub1's Internet public IP address `192.168.129.191`.



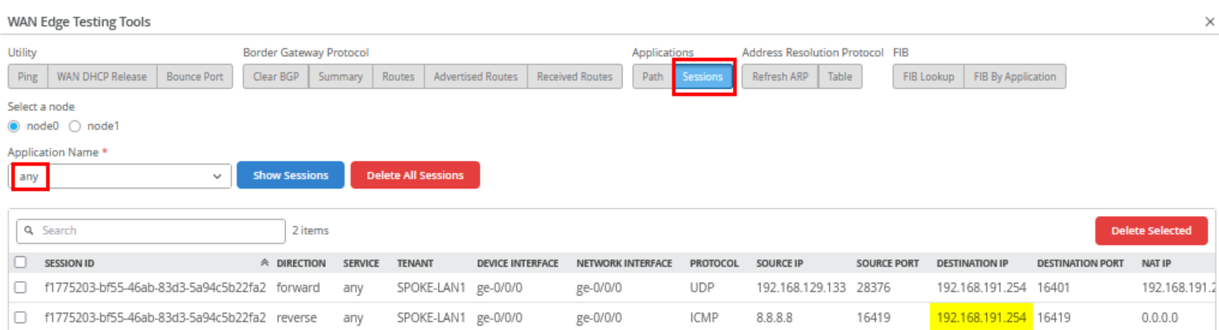
SESSION ID	DIRECTION	SERVICE	TENANT	DEVICE INTERFACE	NETWORK INTERFACE	PROTOCOL	SOURCE IP	SOURCE PORT	DESTINATION IP	DESTINATION PORT	NAT IP	NAT P
f1775203-bf55-46ab-83d3-5a94c5b22fa2	forward	any	SPOKE-LAN1	ge-0/0/3	ge-0/0/3.1088	ICMP	10.88.88.88	22	8.8.8.8	22	0.0.0.0	0
f1775203-bf55-46ab-83d3-5a94c5b22fa2	reverse	any	SPOKE-LAN1	ge-0/0/0	ge-0/0/0	UDP	192.168.129.191	16401	192.168.133.141	16404	0.0.0.0	0

Also review the FIB on spoke2.



IPPREFIX	PORT	PROTOCOL	TENANT	VRF	SERVICE	NEXT HOP INTERFACE	NEXT HOP IP	VECTOR	COST
0.0.0.0/0	0	None	SPOKE-LAN1	default	any	ge-0/0/6	169.254.252.2	hahub-N0-INET.OrgOverlay	10
0.0.0.0/0	0	None	SPOKE-LAN1	default	any	ge-0/0/0	192.168.129.191	hahub-N0-INET.OrgOverlay	10
0.0.0.0/0	0	None	SPOKE-LAN1	default	any	ge-0/0/6	169.254.252.2	hahub-N1-INET.OrgOverlay	20
0.0.0.0/0	0	None	SPOKE-LAN1	default	any	ge-0/0/0	192.168.129.201	hahub-N1-INET.OrgOverlay	20
0.0.0.0/0	0	None	SPOKE-LAN1	default	any	ge-0/0/1	192.168.190.254	hahub-N0-MPLS.OrgOverlay	30
0.0.0.0/0	0	None	SPOKE-LAN1	default	any	ge-0/0/1	192.168.200.254	hahub-N1-MPLS.OrgOverlay	40
10.0.0.0/8	0	None	SPOKE-LAN1	default	any	ge-0/0/6	169.254.252.2	hahub-N0-INET.OrgOverlay	10

Go to **WAN Edges -> site=hahub-site** and select “hahub”. Then, use **Utilities -> Testing Tools** and review the application sessions with Application Name=`any` again. Here, you can see the reverse flow ICMP responses to the source NATed interface `ge-0/0/0` where we forwarded our traffic to.



SESSION ID	DIRECTION	SERVICE	TENANT	DEVICE INTERFACE	NETWORK INTERFACE	PROTOCOL	SOURCE IP	SOURCE PORT	DESTINATION IP	DESTINATION PORT	NAT IP
f1775203-bf55-46ab-83d3-5a94c5b22fa2	forward	any	SPOKE-LAN1	ge-0/0/0	ge-0/0/0	UDP	192.168.129.133	28376	192.168.191.254	16401	192.168.191.2
f1775203-bf55-46ab-83d3-5a94c5b22fa2	reverse	any	SPOKE-LAN1	ge-0/0/0	ge-0/0/0	ICMP	8.8.8.8	16419	192.168.191.254	16419	0.0.0.0

The remaining testing is done with the clients attached to the redundant hub. We connect to the desktop4 VM with IP address `10.66.66.66` attached to hahub.

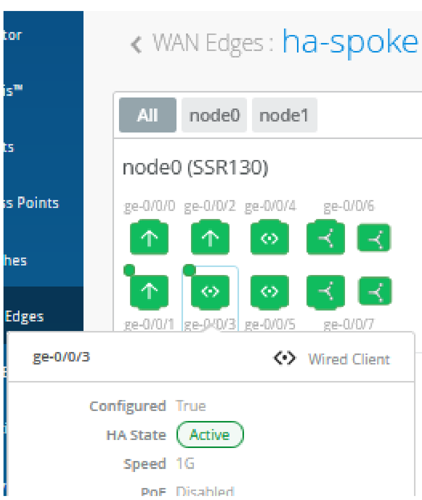
```
# try to reach the client desktop1 VM attached to singe spoke1
root@desktop4:~# ping -c3 10.99.99.99
PING 10.99.99.99 (10.99.99.99) 56(84) bytes of data.
```

```
64 bytes from 10.99.99.99: icmp_seq=1 ttl=59 time=4.98 ms
64 bytes from 10.99.99.99: icmp_seq=2 ttl=59 time=1.07 ms
64 bytes from 10.99.99.99: icmp_seq=3 ttl=59 time=1.03 ms
#
# try to reach the client desktop2 VM attached to redundant spoke2
root@desktop4:~# ping -c3 10.88.88.88
PING 10.88.88.88 (10.88.88.88) 56(84) bytes of data.
64 bytes from 10.88.88.88: icmp_seq=1 ttl=59 time=5.49 ms
64 bytes from 10.88.88.88: icmp_seq=2 ttl=59 time=1.15 ms
64 bytes from 10.88.88.88: icmp_seq=3 ttl=59 time=1.06 ms
#
# try services on the internet using the local breakout on the hub
root@desktop4:~# ping -c3 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=109 time=3.10 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=109 time=2.67 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=109 time=2.68 ms
```

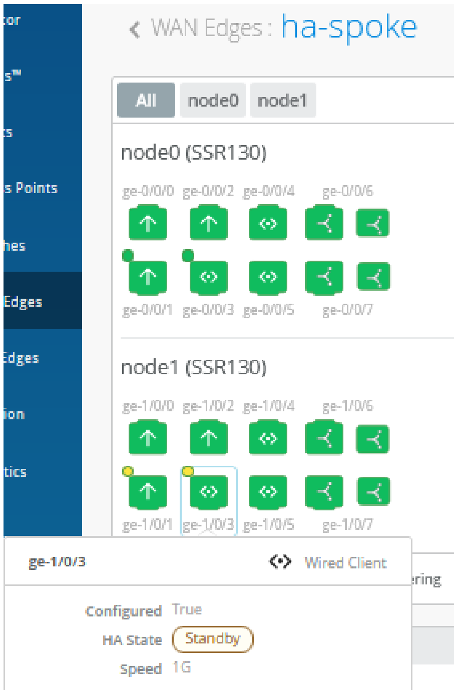
Your lab topology should be up and running now and you can explore failover testing as you like.

The best way to figure out which node is currently active is to look at the LAN interfaces that we've configured for redundancy on our high-availability spokes and hubs. These tasks for JVD testing were performed but not shared here in this document.

In the example below, the interface `ge-0/0/3` on Node0 is in an active HA state.



While interface `ge-1/0/3` on Node1 is in a standby HA state.



Appendix: Building a Full Stack Topology with Juniper EX Switch and Juniper AP

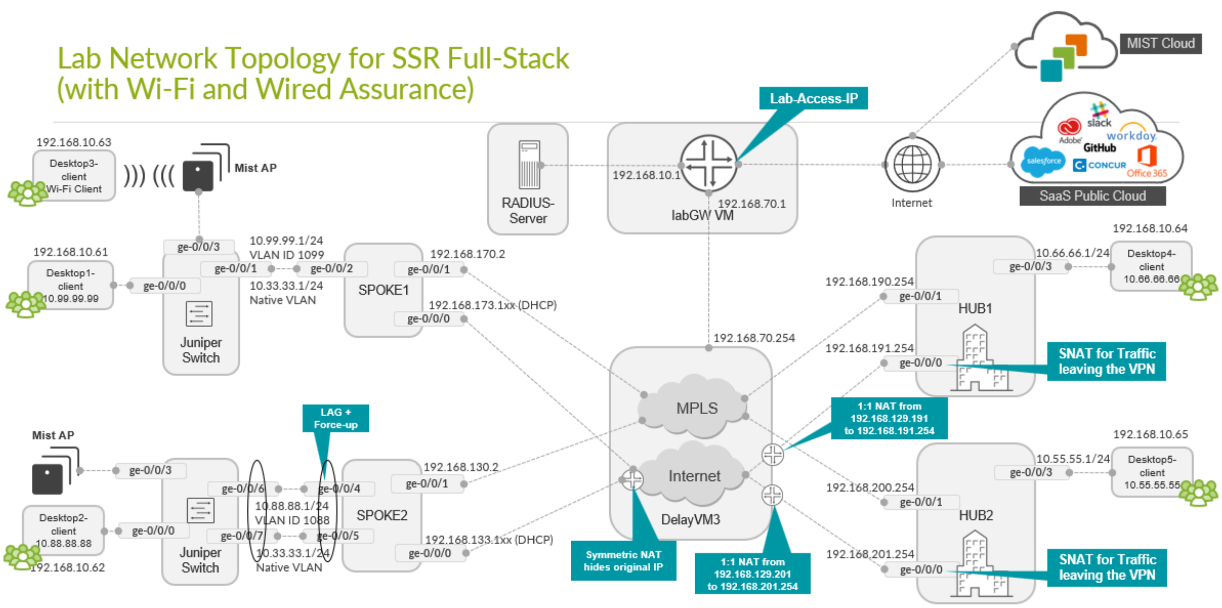
IN THIS SECTION

This lab is an extension of the previous lab "Appendix: Building a base SD-WAN Topology with Three Spokes and Two Hubs" on page 23. There are no hub configuration changes to be made as we do not touch the VPN configuration. We add the following changes to this lab:

- We define a new network intended to manage switches and APs attached to the WAN router.
  - This network will have the same IP address range `10.33.33.0/24` on all sites.

    - We do not propagate this IP address range to the VPN overlay.
    - This traffic will use local breakout on the WAN router to reach the Juniper Mist cloud managing it.
    - The WAN router will have a local DHCP server to hand out leases to the attached devices.
    - The network needs to be native at the LAN interface as the switch ports are initially in access mode.

- On Spoke1, we use the interface `ge-0/0/2` as the downlink to the switch. Hence, we assume this branch has no link redundancy requirement to the switch and attached APs.
- On Spoke2, we use the interfaces `ge-0/0/4` and `ge-0/0/5` as downlinks to the switch. Hence, we can build a LAG with LACP toward the switch to achieve redundancy and load-balancing for more throughput. We also utilize a feature called force-up to the attached switch to be able to reach the Juniper Mist cloud without an initial LAG configuration. This is further documented in the JVD for Distributed Branch EX Series . Please review for more details on switch management towards Juniper Mist cloud and on the advantages of force-up when using a LAG.



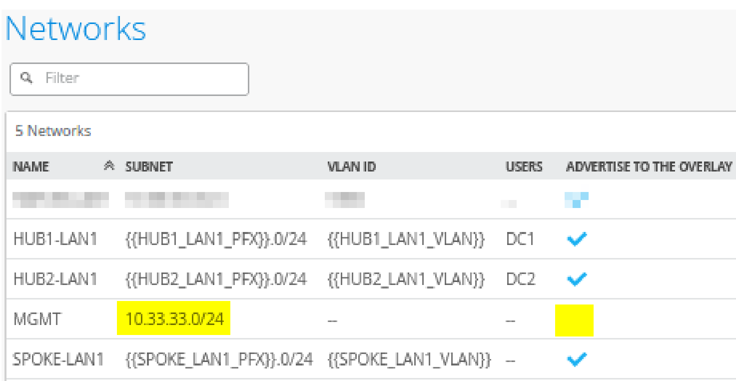
NOTE: When using force-up with a LAG, you must use firmware 6.3.0 or higher for Session Smart Routers.

Create a Management Network

Go to **Organization -> Networks**. Configure the first network in the following way:

- Name=`MGMT`
- Subnet IP Address=`10.33.33.0` (this will be the same on all sites)
- Prefix Length=`24`
- VLAN ID=`<default>/none` This ensures that it will be native on the trunk interface downlink to the switch.
- Access to Mist Cloud=`Checked/Enabled`. This is mandatory to be able to manage the attached devices.
- Advertised via Overlay=`Unchecked/Disabled`. This is mandatory as we can't have the same IP address range announced from multiple sites.

The result should look like the figure below:



Extend the WAN Edge Template for Spoke with One Downlink

Go to **Organization -> WAN Edge Templates.**

Should you choose to use the import option, click on **Import Profile** and import the below JSON as a file.

```
{
  "dhcpd_config": {
    "enabled": true,
    "SPOKE-LAN1": {
      "type": "local",
```

```
      "ip_start": "{{SPOKE_LAN1_PFX}}.10",
      "ip_end": "{{SPOKE_LAN1_PFX}}.250",
      "gateway": "{{SPOKE_LAN1_PFX}}.1",
      "dns_servers": [
        "8.8.8.8",
        "9.9.9.9"
      ],
      "options": {},
      "lease_time": 86400,
      "fixed_bindings": {}
    },
    "MGMT": {
      "type": "local",
      "ip_start": "10.33.33.10",
      "ip_end": "10.33.33.250",
      "gateway": "10.33.33.1",
      "dns_servers": [
        "8.8.8.8",
        "9.9.9.9"
      ],
      "options": {},
      "lease_time": 86400,
      "fixed_bindings": {}
    }
  },
  "ntpOverride": true,
  "dnsOverride": true,
  "service_policies": [
    {
      "name": "spoke-to-hub-dmz",
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "HUB1-LAN1",
        "HUB2-LAN1"
      ],
      "action": "allow",
      "idp": {
        "enabled": false
      },
      "path_preference": "VPN"
    },
```

```
    {
      "name": "hub-dmz-to-spoke",
      "tenants": [
        "HUB1-LAN1",
        "HUB2-LAN1"
      ],
      "services": [
        "SPOKE-LAN1"
      ],
      "action": "allow",
      "path_preference": "LAN",
      "idp": {
        "enabled": false
      }
    },
    {
      "name": "spoke-to-spoke-via-hub",
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "SPOKE-LAN1"
      ],
      "action": "allow",
      "idp": {
        "enabled": false
      },
      "local_routing": true
    },
    {
      "name": "mgmt-to-mist-cloud",
      "tenants": [
        "MGMT"
      ],
      "services": [
        "any"
      ],
      "action": "allow",
      "path_preference": "LBO",
      "idp": {
        "enabled": false
      }
    },
```

```
    {
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "any"
      ],
      "action": "allow",
      "name": "internet-via-hub-cbo",
      "idp": {
        "enabled": false
      },
      "path_preference": "VPN"
    }
  ],
  "ip_configs": {
    "SPOKE-LAN1": {
      "type": "static",
      "ip": "{{SPOKE_LAN1_PFX}}.1",
      "netmask": "/24"
    },
    "MGMT": {
      "type": "static",
      "ip": "10.33.33.1"
    }
  },
  "dns_servers": [
    "8.8.8.8",
    "9.9.9.9"
  ],
  "port_config": {
    "ge-0/0/0": {
      "name": "INET",
      "usage": "wan",
      "wan_type": "broadband",
      "aggregated": false,
      "redundant": false,
      "traffic_shaping": {
        "enabled": false
      },
      "ip_config": {
        "type": "dhcp"
      },
```

```
      "vpn_paths": {
        "hub1-INET.OrgOverlay": {
          "bfd_profile": "broadband",
          "role": "spoke"
        },
        "hub2-INET.OrgOverlay": {
          "bfd_profile": "broadband",
          "role": "spoke"
        }
      }
    },
    "ge-0/0/1": {
      "name": "MPLS",
      "usage": "wan",
      "wan_type": "broadband",
      "aggregated": false,
      "redundant": false,
      "traffic_shaping": {
        "enabled": false
      },
      "ip_config": {
        "type": "static",
        "ip": "{{WAN1_PFX}}.2",
        "netmask": "/24",
        "gateway": "{{WAN1_PFX}}.1"
      },
      "vpn_paths": {
        "hub1-MPLS.OrgOverlay": {
          "bfd_profile": "broadband",
          "role": "spoke"
        },
        "hub2-MPLS.OrgOverlay": {
          "bfd_profile": "broadband",
          "role": "spoke"
        }
      }
    },
    "ge-0/0/2": {
      "networks": [
        "SPOKE-LAN1",
        "MGMT"
      ],
      "usage": "lan",
```

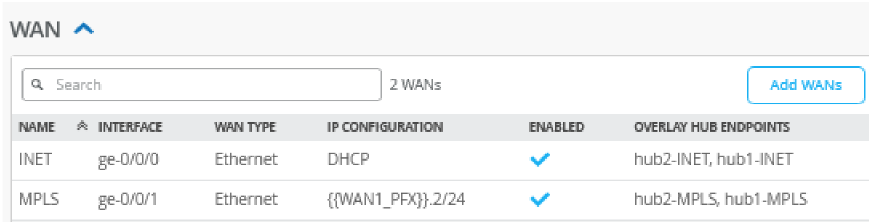
```
      "aggregated": false,
      "redundant": false,
      "critical": false,
      "disabled": false
    }
  },
  "bgp_config": {},
  "routing_policies": {},
  "extra_routes": {},
  "path_preferences": {
    "VPN": {
      "strategy": "weighted",
      "paths": [
        {
          "name": "hub1-INET.OrgOverlay",
          "cost": 10,
          "type": "vpn"
        },
        {
          "name": "hub2-INET.OrgOverlay",
          "cost": 20,
          "type": "vpn"
        },
        {
          "name": "hub1-MPLS.OrgOverlay",
          "cost": 30,
          "type": "vpn"
        },
        {
          "name": "hub2-MPLS.OrgOverlay",
          "cost": 40,
          "type": "vpn"
        }
      ]
    },
    "LAN": {
      "strategy": "ordered",
      "paths": [
        {
          "type": "local",
          "networks": [
            "SPOKE-LAN1"
          ]
```

```
        }
      ]
    },
    "LBO": {
      "strategy": "ordered",
      "paths": [
        {
          "name": "INET",
          "type": "wan"
        }
      ]
    }
  },
  "ospf_areas": {},
  "vrf_instances": {},
  "tunnel_configs": {},
  "oob_ip_config": {
    "type": "dhcp",
    "node1": {
      "type": "dhcp"
    }
  },
  "tunnel_provider_options": {
    "jse": {},
    "zscaler": {}
  },
  "ospf_config": {
    "enabled": false,
    "areas": {}
  },
  "type": "spoke",
  "name": "Spokes"
}
```


Should you decide to configure everything manually in the Juniper Mist portal, then use the following steps.

We modify the existing “Spokes” template in the following way to get the additional management network configured.

The WAN interface configuration does not need to be changed hence it should still look like the figure below:



WAN

Search | 2 WANs | Add WANs

NAME	INTERFACE	WAN TYPE	IP CONFIGURATION	ENABLED	OVERLAY HUB ENDPOINTS
INET	ge-0/0/0	Ethernet	DHCP	✓	hub2-INET, hub1-INET
MPLS	ge-0/0/1	Ethernet	{{WAN1_PFX}}.2/24	✓	hub2-MPLS, hub1-MPLS

In the LAN section, we need to add the following IP Configuration:

- Network=`MGMT`
- IP Address=`10.33.33.1`
- Prefix Length=24

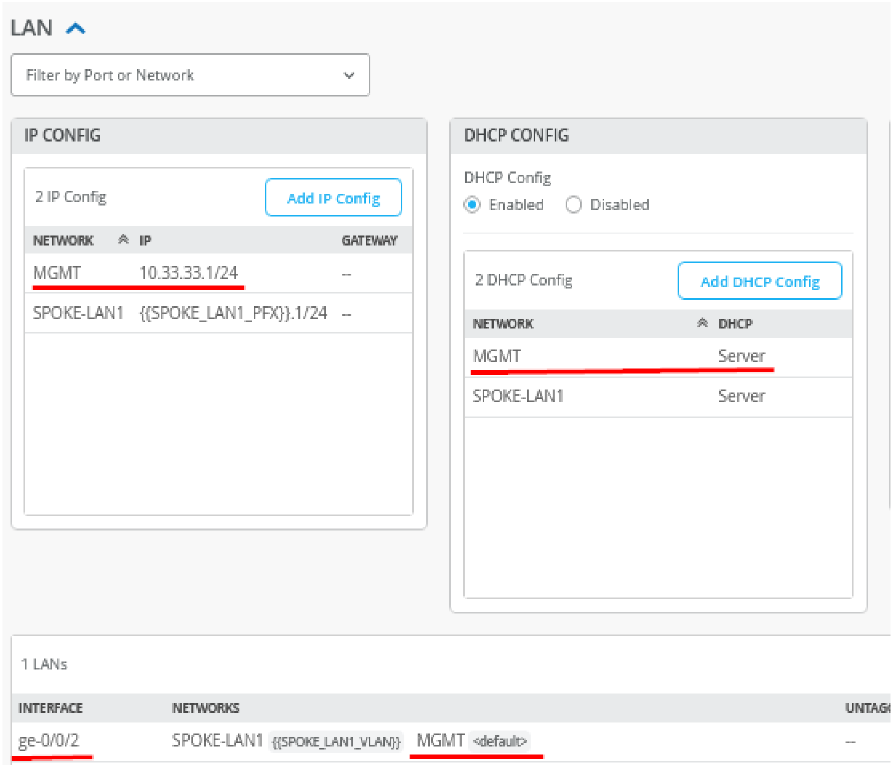
Then, we create an additional DHCP server for this network with the following configuration:

- Network=`MGMT`
- DHCP=`Server`
- IP Start=`10.33.33.10`
- IP End=`10.33.33.250`
- Gateway=`10.33.33.1`
- Maximum Lease Time=`86400`
- DNS Servers=`8.8.8.8, 9.9.9.9`

The LAN interface configuration is then changed to the following configuration:

- Interface=`ge-0/0/2`
- Networks=`SPOKE1-LAN1 + MGMT`
- Untagged VLAN=`None`

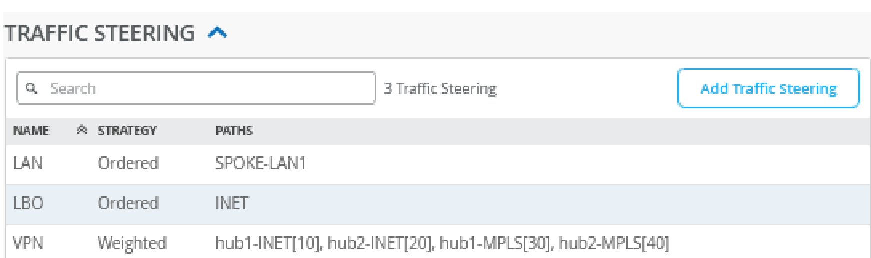
The result should look like the figure below:



We now need to configure an additional traffic steering profile for local breakout of the management network as we do not need this to be part of the overlay VPN. Add an additional traffic steering rule with the following configuration:

- Name=`LBO`
- Strategy=`Ordered`
- Paths
  - Path1 Type=`WAN: INET`

The result should look like the figure below:

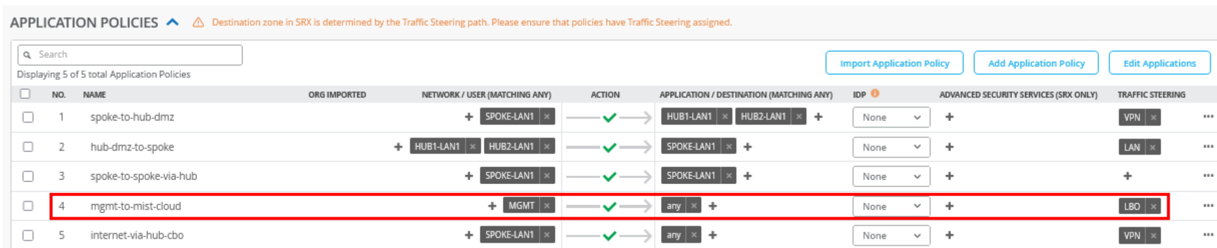


Insert the following application policy:

- Number=`4`

- Name=`mgmt-to-mist-cloud`
- Network=`MGMT`
- Action=`Pass`
- Application=`any`
- Traffic Steering=`LBO`

The result should look like the figure below:



Save your changes.

Create the WAN Edge Template for the Spoke with a LAG Towards the Switch

Go to **Organization -> WAN Edge Templates.**

Should you choose to use the import option, click on **Import Profile** and import the below JSON as a file.

```
{
  "type": "spoke",
  "dhcpd_config": {
    "enabled": true,
    "SPOKE-LAN1": {
      "type": "local",
      "ip_start": "{{SPOKE_LAN1_PFX}}.10",
      "ip_end": "{{SPOKE_LAN1_PFX}}.250",
      "gateway": "{{SPOKE_LAN1_PFX}}.1",
      "dns_servers": [
        "8.8.8.8",
        "9.9.9.9"
      ],
      "options": {},
```

```
      "lease_time": 86400,
      "fixed_bindings": {}
    },
    "MGMT": {
      "type": "local",
      "ip_start": "10.33.33.10",
      "ip_end": "10.33.33.250",
      "gateway": "10.33.33.1",
      "dns_servers": [
        "8.8.8.8",
        "9.9.9.9"
      ],
      "options": {},
      "lease_time": 86400,
      "fixed_bindings": {}
    }
  },
  "ntpOverride": true,
  "dnsOverride": true,
  "service_policies": [
    {
      "name": "spoke-to-hub-dmz",
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "HUB1-LAN1",
        "HUB2-LAN1"
      ],
      "action": "allow",
      "idp": {
        "enabled": false
      },
      "path_preference": "VPN"
    },
    {
      "name": "hub-dmz-to-spoke",
      "tenants": [
        "HUB1-LAN1",
        "HUB2-LAN1"
      ],
      "services": [
        "SPOKE-LAN1"
```

```
    ],
    "action": "allow",
    "path_preference": "LAN",
    "idp": {
      "enabled": false
    }
  },
  {
    "name": "spoke-to-spoke-via-hub",
    "tenants": [
      "SPOKE-LAN1"
    ],
    "services": [
      "SPOKE-LAN1"
    ],
    "action": "allow",
    "idp": {
      "enabled": false
    },
    "local_routing": true
  },
  {
    "name": "mgmt-to-mist-cloud",
    "tenants": [
      "MGMT"
    ],
    "services": [
      "any"
    ],
    "action": "allow",
    "path_preference": "LBO",
    "idp": {
      "enabled": false
    }
  },
  {
    "tenants": [
      "SPOKE-LAN1"
    ],
    "services": [
      "any"
    ],
    "action": "allow",
```

```
      "name": "internet-via-hub-cbo",
      "idp": {
        "enabled": false
      },
      "path_preference": "VPN"
    }
  ],
  "ip_configs": {
    "SPOKE-LAN1": {
      "type": "static",
      "ip": "{{SPOKE_LAN1_PFX}}.1",
      "netmask": "/24"
    },
    "MGMT": {
      "type": "static",
      "ip": "10.33.33.1"
    }
  },
  "dns_servers": [
    "8.8.8.8",
    "9.9.9.9"
  ],
  "port_config": {
    "ge-0/0/0": {
      "name": "INET",
      "usage": "wan",
      "wan_type": "broadband",
      "aggregated": false,
      "redundant": false,
      "traffic_shaping": {
        "enabled": false
      },
      "ip_config": {
        "type": "dhcp"
      },
      "vpn_paths": {
        "hub1-INET.OrgOverlay": {
          "bfd_profile": "broadband",
          "role": "spoke"
        },
        "hub2-INET.OrgOverlay": {
          "bfd_profile": "broadband",
          "role": "spoke"
```

```
      }
    }
  },
  "ge-0/0/1": {
    "name": "MPLS",
    "usage": "wan",
    "wan_type": "broadband",
    "aggregated": false,
    "redundant": false,
    "traffic_shaping": {
      "enabled": false
    },
    "ip_config": {
      "type": "static",
      "ip": "{{WAN1_PFX}}.2",
      "netmask": "/24",
      "gateway": "{{WAN1_PFX}}.1"
    },
    "vpn_paths": {
      "hub1-MPLS.OrgOverlay": {
        "bfd_profile": "broadband",
        "role": "spoke"
      },
      "hub2-MPLS.OrgOverlay": {
        "bfd_profile": "broadband",
        "role": "spoke"
      }
    }
  },
  "ge-0/0/4,ge-0/0/5": {
    "networks": [
      "SPOKE-LAN1",
      "MGMT"
    ],
    "usage": "lan",
    "aggregated": true,
    "ae_disable_lacp": false,
    "ae_lacp_force_up": true,
    "ae_idx": "0",
    "redundant": false,
    "critical": false,
    "disabled": false
  }
```

```
  },
  "bgp_config": {},
  "routing_policies": {},
  "extra_routes": {},
  "path_preferences": {
    "VPN": {
      "strategy": "weighted",
      "paths": [
        {
          "name": "hub1-INET.OrgOverlay",
          "cost": 10,
          "type": "vpn"
        },
        {
          "name": "hub2-INET.OrgOverlay",
          "cost": 20,
          "type": "vpn"
        },
        {
          "name": "hub1-MPLS.OrgOverlay",
          "cost": 30,
          "type": "vpn"
        },
        {
          "name": "hub2-MPLS.OrgOverlay",
          "cost": 40,
          "type": "vpn"
        }
      ]
    },
    "LAN": {
      "strategy": "ordered",
      "paths": [
        {
          "type": "local",
          "networks": [
            "SPOKE-LAN1"
          ]
        }
      ]
    },
    "LBO": {
      "strategy": "ordered",
```

```
      "paths": [
        {
          "name": "INET",
          "type": "wan"
        }
      ]
    }
  },
  "ospf_areas": {},
  "vrf_instances": {},
  "tunnel_configs": {},
  "oob_ip_config": {
    "type": "dhcp",
    "node1": {
      "type": "dhcp"
    }
  },
  "tunnel_provider_options": {
    "jse": {},
    "zscaler": {}
  },
  "ospf_config": {
    "enabled": false,
    "areas": {}
  },
  "name": "Spokes-with-LAN-LAG"
}
```

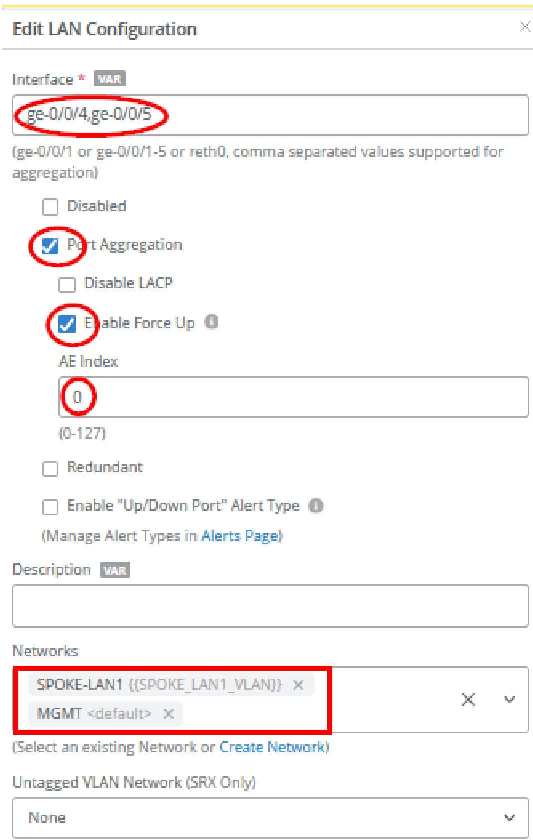

Should you decide to configure everything manually in the Juniper Mist portal, then use the following steps.

We recommend you clone the existing “Spokes” template that we modified already in the last section for this lab and name the new template “Spokes-with-LAN-LAG”.

Then, we only need to change the LAN interface configuration with the following configuration:

- Interface=`ge-0/0/4,ge-0/0/5`
- Port Aggregation=`Checked/Enabled`
  - Disable LACP=`Unchecked/Disabled`
  - Enable Force Up=`Checked/Enabled`
  - AE Index=`0`

- Networks=`SPOKE1-LAN1 + MGMT`
- Untagged VLAN=`None`



Save your changes and then apply this template to spoke2-site.

Test Your Network Configuration

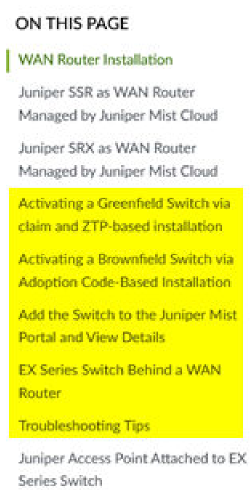
We are now ready to test our configuration. With the single downlink spoke configuration on Spoke1 in place and a console cable to the switch, you can evaluate the following:

```
# ensure you ask for a new DHCP-Lease
root@switch1> restart dhcp-service
Junos Dynamic Host Configuration Protocol process started, pid 55092
#
# wait a few seconds
#
# review your routing table
```

```
root@switch1> show route
.
inet.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
Limit/Threshold: 32768/32768 destinations
+ = Active Route, - = Last Active, * = Both
.
0.0.0.0/0          *[Access-internal/12] 00:00:02, metric 0
                    >  to 10.33.33.1 via irb.0
10.33.33.0/24      *[Direct/0] 00:00:02
                    >  via irb.0
10.33.33.10/32     *[Local/0] 00:00:02
                       Local via irb.0
.
#
# review MAC-Table
root@switch1> show ethernet-switching table
.
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static, C -
Control MAC
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC
           GBP - group based policy)
.
Ethernet switching table : 2 entries, 2 learned
Routing instance : default-switch
    Vlan                MAC                 MAC         Age   GBP     Logical
NH        RTR
    name                address             flags             Tag     interface
Index     ID
    default             90:ec:77:32:e4:8d   D             -            ge-0/0/1.0
0         0
    default             d4:20:b0:01:46:81   D             -            ge-0/0/3.0
0         0
#
# review IP address via ARP from WAN-Router received from an interface
root@switch1> show arp no-resolve
MAC Address       Address         Interface                Flags
90:ec:77:32:e4:8d 10.33.33.1      irb.0 [ge-0/0/1.0]       none
#
# confirm DNS and internet access
root@switch1> ping www.google.com inet
PING www.google.com (172.217.12.100): 56 data bytes
64 bytes from 172.217.12.100: icmp_seq=0 ttl=110 time=13.557 ms
64 bytes from 172.217.12.100: icmp_seq=1 ttl=110 time=15.349 ms
```

```
64 bytes from 172.217.12.100: icmp_seq=2 ttl=110 time=15.361 ms
^C
#
# review LLDP Neighbors
root@switch1> show lldp neighbors
Local Interface    Parent Interface    Chassis Id          Port info          System Name
ge-0/0/1           -                   90:ec:77:32:e4:8d   ge-0-2             spoke1
ge-0/0/3           -                   d4:20:b0:01:46:81   ETH0               d420b0014681
```

The test above shows that the switch obtained a DHCP lease and should be able to initiate traffic with the Juniper Mist cloud to be managed. The remaining steps to onboard an EX Series Switch are explained in the JVD Distributed Branch EX Series. In the Day 1 section, review the sections shown in the figure below:



With the two downlinks configured on Spoke2 and a console cable attached to the switch, you can evaluate the following:

```
# ensure you ask for a new DHCP-Lease
root@switch1> restart dhcp-service
Junos Dynamic Host Configuration Protocol process started, pid 59162
#
# wait a few seconds
#
# review your routing table
root@switch1> show route
.
inet.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
Limit/Threshold: 32768/32768 destinations
+ = Active Route, - = Last Active, * = Both
```

```
.
0.0.0.0/0          *[Access-internal/12] 00:00:03, metric 0
                    >  to 10.33.33.1 via irb.0
10.33.33.0/24      *[Direct/0] 00:00:03
                    >  via irb.0
10.33.33.12/32     *[Local/0] 00:00:03
                       Local via irb.0
.
#
# review MAC-Table
root@switch1> show ethernet-switching table
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static, C -
Control MAC
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC
           GBP - group based policy)
.
Ethernet switching table : 3 entries, 3 learned
Routing instance : default-switch
    Vlan                MAC                 MAC         Age   GBP     Logical
NH        RTR
    name                address             flags             Tag     interface
Index     ID
    default             90:ec:77:32:df:a5   D             -            ge-0/0/7.0
0         0
    default             d4:20:b0:01:46:bb   D             -            ge-0/0/3.0
0         0
#
# review  via ARP from WAN-Router received from an interface
MAC Address       Address         Interface                Flags
90:ec:77:32:df:a5 10.33.33.1      irb.0 [ge-0/0/7.0]       none
#
# confirm DNS and internet access
root@switch1> ping www.google.com inet
PING www.google.com (172.217.12.100): 56 data bytes
64 bytes from 172.217.12.100: icmp_seq=0 ttl=110 time=10.321 ms
64 bytes from 172.217.12.100: icmp_seq=1 ttl=110 time=16.570 ms
64 bytes from 172.217.12.100: icmp_seq=2 ttl=110 time=94.168 ms
^C
#
# review LLDP Neighbors
root@switch1> show lldp neighbors
Local Interface    Parent Interface    Chassis Id          Port info          System Name
ge-0/0/6           -                   90:ec:77:32:df:a5   ge-0-4             spoke2
```

```
ge-0/0/7           -                 90:ec:77:32:df:a6   ge-0-5           spoke2
ge-0/0/3           -                 d4:20:b0:01:46:bb   ETH0             d420b00146bb
#
# in this factory state there should not be yet any LACP configuration
root@switch1> show lacp interfaces
warning: lacp subsystem not running - not needed by configuration.
```

This section does not repeat the traffic topology tests, as the changes introduced are minimal. For detailed testing procedures, please refer to the "Test Your Network Configuration" on page 71 section in the first topology.

NOTE: Should you have this implemented then consider changing the spoke and hub LAN network configuration to no longer allow “Access to Mist Cloud” as we’ve done previously by default.

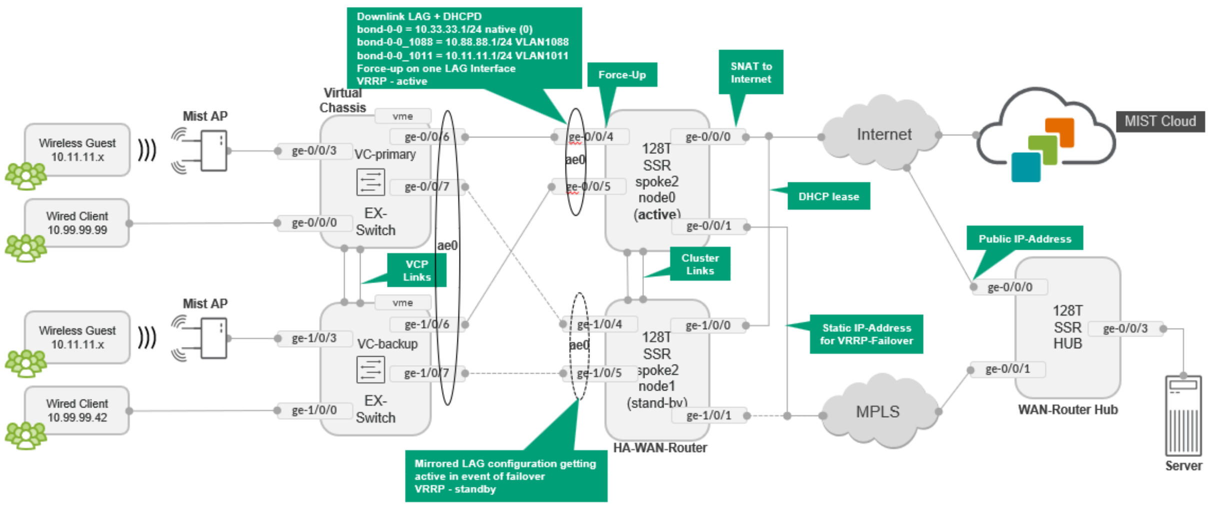
Appendix: Building an Extended Full Stack Topology with Juniper EX Switch Virtual Chassis and SSR HA Cluster

IN THIS SECTION

The last topology tested as part of this JVD is an extension of the lab from "Appendix: Building a Full Stack Topology with Juniper EX Switch and Juniper AP" on page 155. The configuration for a Virtual Chassis connected to a single spoke is omitted, as the only additional step for improved resiliency is connecting one of the downlinks to the backup member of the Virtual Chassis. In this lab we configure:

- A redundant high availability cluster spoke while you still have two separate hubs.

- A Virtual Chassis is built using Juniper EX Series Switches, with two uplinks connected to the WAN router nodes as shown in the topology diagram below, providing resiliency.
- A new network with the range 10.11.11.0/24 is added to all spokes to simulate a guest Wi-Fi network. This network is restricted from sending traffic into the VPN. In this lab setup, guest traffic is allowed to use local Internet breakout. Later, the goal is to route all guest traffic to a cloud service for inspection and compliance filtering before it reaches the Internet.

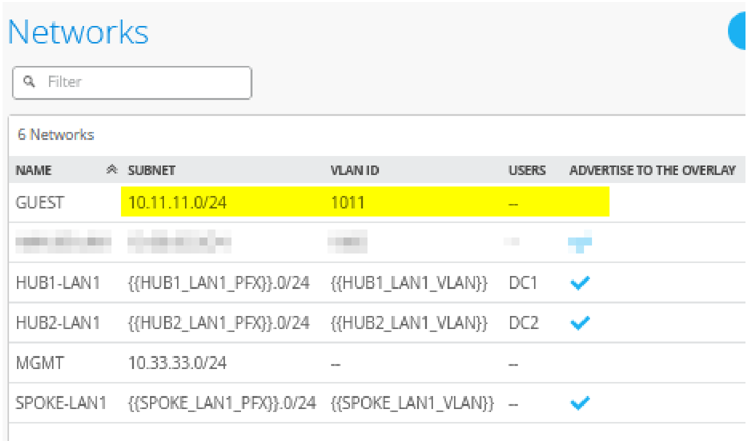


Create a Wi-Fi Guest Network

Go to **Organization -> Networks**. Configure the first network in the following way:

- Name=`GUEST`
- Subnet IP Address=`10.11.11.0`. This will be the same on all sites.
- Prefix Length=`24`
- VLAN ID=`1011`
- Access to Mist Cloud=`Unchecked/Disabled`. This should not be needed for guests.
- Advertised via Overlay=`Unchecked/Disabled`. This is mandatory here as we can't have the same IP address range announced from multiple sites.

The result should look like the figure below:



NAME	SUBNET	VLAN ID	USERS	ADVERTISE TO THE OVERLAY
GUEST	10.11.11.0/24	1011	–	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
HUB1-LAN1	{{HUB1_LAN1_PFX}}.0/24	{{HUB1_LAN1_VLAN}}	DC1	✓
HUB2-LAN1	{{HUB2_LAN1_PFX}}.0/24	{{HUB2_LAN1_VLAN}}	DC2	✓
MGMT	10.33.33.0/24	–	–	
SPOKE-LAN1	{{SPOKE_LAN1_PFX}}.0/24	{{SPOKE_LAN1_VLAN}}	–	✓

Create a WAN Edge Template for HA Spoke with LAG

Go to **Organization -> WAN Edge Templates.**

Should you choose to use the import option, click on **Import Profile** and import the below JSON as a file.

```
{
  "type": "spoke",
  "dhcpd_config": {
    "enabled": true,
    "SPOKE-LAN1": {
      "type": "local",
      "ip_start": "{{SPOKE_LAN1_PFX}}.10",
      "ip_end": "{{SPOKE_LAN1_PFX}}.250",
      "gateway": "{{SPOKE_LAN1_PFX}}.1",
      "dns_servers": [
        "8.8.8.8",
        "9.9.9.9"
      ],
      "options": {},
      "lease_time": 86400,
      "fixed_bindings": {}
    },
    "MGMT": {
      "type": "local",
      "ip_start": "10.33.33.10",
      "ip_end": "10.33.33.250",
      "gateway": "10.33.33.1",
```

```
      "dns_servers": [
        "8.8.8.8",
        "9.9.9.9"
      ],
      "options": {},
      "lease_time": 86400,
      "fixed_bindings": {}
    },
    "GUEST": {
      "type": "local",
      "ip_start": "10.11.11.10",
      "ip_end": "10.11.11.250",
      "gateway": "10.11.11.1",
      "dns_servers": [
        "8.8.8.8",
        "9.9.9.9"
      ],
      "options": {},
      "lease_time": 86400,
      "fixed_bindings": {}
    }
  },
  "ntpOverride": true,
  "dnsOverride": true,
  "service_policies": [
    {
      "name": "spoke-to-hub-dmz",
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "HUB1-LAN1",
        "HUB2-LAN1"
      ],
      "action": "allow",
      "idp": {
        "enabled": false
      },
      "path_preference": "VPN"
    },
    {
      "name": "hub-dmz-to-spoke",
      "tenants": [
```

```
        "HUB1-LAN1",
        "HUB2-LAN1"
      ],
      "services": [
        "SPOKE-LAN1"
      ],
      "action": "allow",
      "path_preference": "LAN",
      "idp": {
        "enabled": false
      }
    },
    {
      "name": "spoke-to-spoke-via-hub",
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "SPOKE-LAN1"
      ],
      "action": "allow",
      "idp": {
        "enabled": false
      },
      "local_routing": true
    },
    {
      "name": "mgmt-to-mist-cloud",
      "tenants": [
        "MGMT"
      ],
      "services": [
        "any"
      ],
      "action": "allow",
      "path_preference": "LBO",
      "idp": {
        "enabled": false
      }
    },
    {
      "name": "guest-to-lbo",
      "tenants": [
```

```
        "GUEST"
      ],
      "services": [
        "any"
      ],
      "action": "allow",
      "path_preference": "LBO",
      "idp": {
        "enabled": false
      }
    },
    {
      "tenants": [
        "SPOKE-LAN1"
      ],
      "services": [
        "any"
      ],
      "action": "allow",
      "name": "internet-via-hub-cbo",
      "idp": {
        "enabled": false
      },
      "path_preference": "VPN"
    }
  ],
  "ip_configs": {
    "SPOKE-LAN1": {
      "type": "static",
      "ip": "{{SPOKE_LAN1_PFX}}.1",
      "netmask": "/24"
    },
    "MGMT": {
      "type": "static",
      "ip": "10.33.33.1"
    },
    "GUEST": {
      "type": "static",
      "ip": "10.11.11.1"
    }
  },
  "dns_servers": [
    "8.8.8.8",
```

```
    "9.9.9.9"
  ],
  "port_config": {
    "ge-0/0/0": {
      "name": "N0-INET",
      "usage": "wan",
      "aggregated": false,
      "redundant": false,
      "critical": false,
      "disabled": false,
      "wan_type": "broadband",
      "ip_config": {
        "type": "dhcp"
      },
      "disable_autoneg": false,
      "wan_source_nat": {
        "disabled": false
      },
      "vpn_paths": {
        "hub1-INET.OrgOverlay": {
          "bfd_profile": "broadband",
          "role": "spoke"
        },
        "hub2-INET.OrgOverlay": {
          "bfd_profile": "broadband",
          "role": "spoke"
        }
      }
    },
    "ge-1/0/0": {
      "name": "N1-INET",
      "usage": "wan",
      "aggregated": false,
      "redundant": false,
      "critical": false,
      "disabled": false,
      "wan_type": "broadband",
      "ip_config": {
        "type": "dhcp"
      },
      "disable_autoneg": false,
      "wan_source_nat": {
        "disabled": false
```

```
      },
      "vpn_paths": {
        "hub1-INET.OrgOverlay": {
          "role": "spoke",
          "bfd_profile": "broadband"
        },
        "hub2-INET.OrgOverlay": {
          "role": "spoke",
          "bfd_profile": "broadband"
        }
      }
    },
    "ge-0/0/1,ge-1/0/1": {
      "name": "HA-MPLS",
      "usage": "wan",
      "aggregated": false,
      "redundant": true,
      "reth_idx": 1,
      "reth_node": "node0",
      "critical": false,
      "disabled": false,
      "wan_type": "broadband",
      "ip_config": {
        "type": "static",
        "ip": "{{WAN1_PFX}}.2",
        "netmask": "/24",
        "gateway": "{{WAN1_PFX}}.1"
      },
      "disable_autoneg": false,
      "wan_source_nat": {
        "disabled": false
      },
      "vpn_paths": {
        "hub1-MPLS.OrgOverlay": {
          "role": "spoke",
          "bfd_profile": "broadband"
        },
        "hub2-MPLS.OrgOverlay": {
          "role": "spoke",
          "bfd_profile": "broadband"
        }
      }
    },
```

```
    "ge-0/0/4,ge-0/0/5,ge-1/0/4,ge-1/0/5": {
      "networks": [
        "SPOKE-LAN1",
        "MGMT",
        "GUEST"
      ],
      "usage": "lan",
      "aggregated": true,
      "ae_disable_lacp": false,
      "ae_lacp_force_up": true,
      "ae_idx": "0",
      "redundant": true,
      "reth_idx": 4,
      "reth_node": "node0",
      "critical": false,
      "disabled": false
    }
  },
  "bgp_config": {},
  "routing_policies": {},
  "extra_routes": {},
  "path_preferences": {
    "VPN": {
      "strategy": "weighted",
      "paths": [
        {
          "name": "hub1-INET.OrgOverlay",
          "cost": 10,
          "type": "vpn"
        },
        {
          "name": "hub2-INET.OrgOverlay",
          "cost": 20,
          "type": "vpn"
        },
        {
          "name": "hub1-MPLS.OrgOverlay",
          "cost": 30,
          "type": "vpn"
        },
        {
          "name": "hub2-MPLS.OrgOverlay",
          "cost": 40,
```

```
        "type": "vpn"
      }
    ]
  },
  "LAN": {
    "strategy": "ordered",
    "paths": [
      {
        "type": "local",
        "networks": [
          "SPOKE-LAN1"
        ]
      }
    ]
  },
  "LBO": {
    "strategy": "ordered",
    "paths": [
      {
        "name": "N0-INET",
        "type": "wan"
      },
      {
        "name": "N1-INET",
        "type": "wan"
      }
    ]
  }
},
"ospf_areas": {},
"vrf_instances": {},
"tunnel_configs": {},
"oob_ip_config": {
  "type": "dhcp",
  "node1": {
    "type": "dhcp"
  }
},
"tunnel_provider_options": {
  "jse": {},
  "zscaler": {}
},
"ospf_config": {
```

```
    "enabled": false,
    "areas": {}
  },
  "name": "haspoke-with-lag"
}
```


Should you decide to configure everything manually in the Juniper Mist portal, then use the following steps.

We recommend you clone the existing “Spokes-with-LAN-LAG” template and name the new template “haspoke-with-lag”. Then make the following changes as the WAN and LAN interfaces change with the high-availability configuration.

Delete all prior WAN interfaces and add the following three new WAN interfaces.

Configure a first WAN interface for Node0 as follows:

- Name=`N0-INET`. This indicates which topology and node it’s going to use.
- WAN Type=`Ethernet`
- Interface=`ge-0/0/0` as all interfaces starting with `ge-0` are on Node0.
- IP Configuration=`DHCP`
- Source NAT=`Interface`
- Overlay Hub Endpoints
  - Endpoint1=`hub1-INET`
  - BFD Profile1=`Broadband`
  - Endpoint2=`hub2-INET`
  - BFD Profile2=`Broadband`

Then configure the first WAN interface for Node1 as follows:

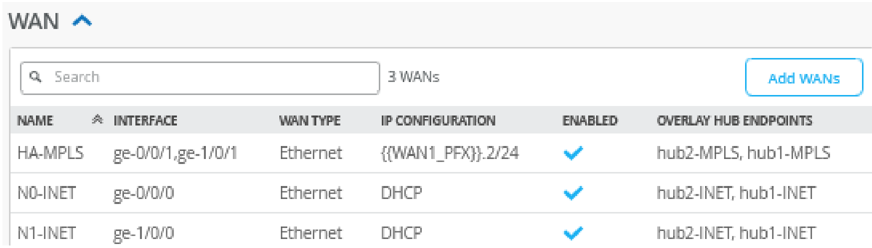
- Name=`N1-INET`. This indicates which topology and node it’s going to use.
- WAN Type=`Ethernet`
- Interface=`ge-1/0/0` as all interfaces starting with `ge-1` are on Node1.
- IP Configuration=`DHCP`
- Source NAT=`Interface`
- Overlay Hub Endpoints

  - Endpoint1=`hub1-INET`
  - BFD Profile1=`Broadband`
  - Endpoint2=`hub2-INET`
  - BFD Profile2=`Broadband`

Then, configure the third and redundant WAN interface for the MPLS path for Node0 and Node1 as follows:

- Name=`HA-MPLS`. This indicates which topology and node it's going to use.
- WAN Type=`Ethernet`
- Interface=`ge-0/0/1` and `ge-1/0/1`
- Redundant=Checked/Enabled
- Redundant Index=1. This is not required for an SSR, but we add for compatibility.
- Primary Node=node0
- IP Configuration=`Static`
- `IP Address={{WAN1_PFX}}.2`
- `Prefix Length=24`
- `Gateway={{WAN1_PFX}}.1`
- Source NAT=`Interface`
- Overlay Hub Endpoints
  - Endpoint1=`hub1-MPLS`
  - BFD Profile1=`Broadband`
  - Endpoint2=`hub2-MPLS`
  - BFD Profile2=`Broadband`

The result should look like the figure below:



WAN

Search | 3 WANs | Add WANs

NAME	INTERFACE	WAN TYPE	IP CONFIGURATION	ENABLED	OVERLAY HUB ENDPOINTS
HA-MPLS	ge-0/0/1,ge-1/0/1	Ethernet	{{WAN1_PFX}}.2/24	✓	hub2-MPLS, hub1-MPLS
N0-INET	ge-0/0/0	Ethernet	DHCP	✓	hub2-INET, hub1-INET
N1-INET	ge-1/0/0	Ethernet	DHCP	✓	hub2-INET, hub1-INET

In the LAN section, we need to add the following IP Configuration:

- Network=`GUEST`
- IP Address=`10.11.11.1`
- Prefix Length=`24`

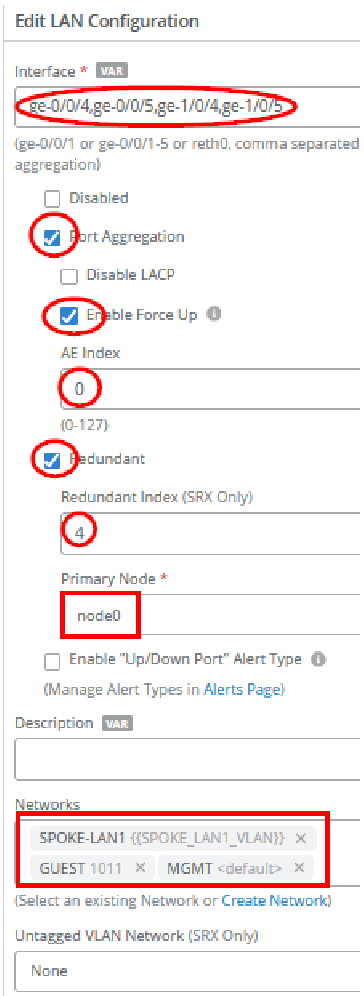
Then, we create an additional DHCP server for this network with the following configuration:

- Network=`GUEST`
- DHCP=`Server`
- IP Start=`10.11.11.10`
- IP End=`10.11.11.250`
- Gateway=`10.11.11.1`
- Maximum Lease Time=`86400`
- DNS Servers=`8.8.8.8, 9.9.9.9`

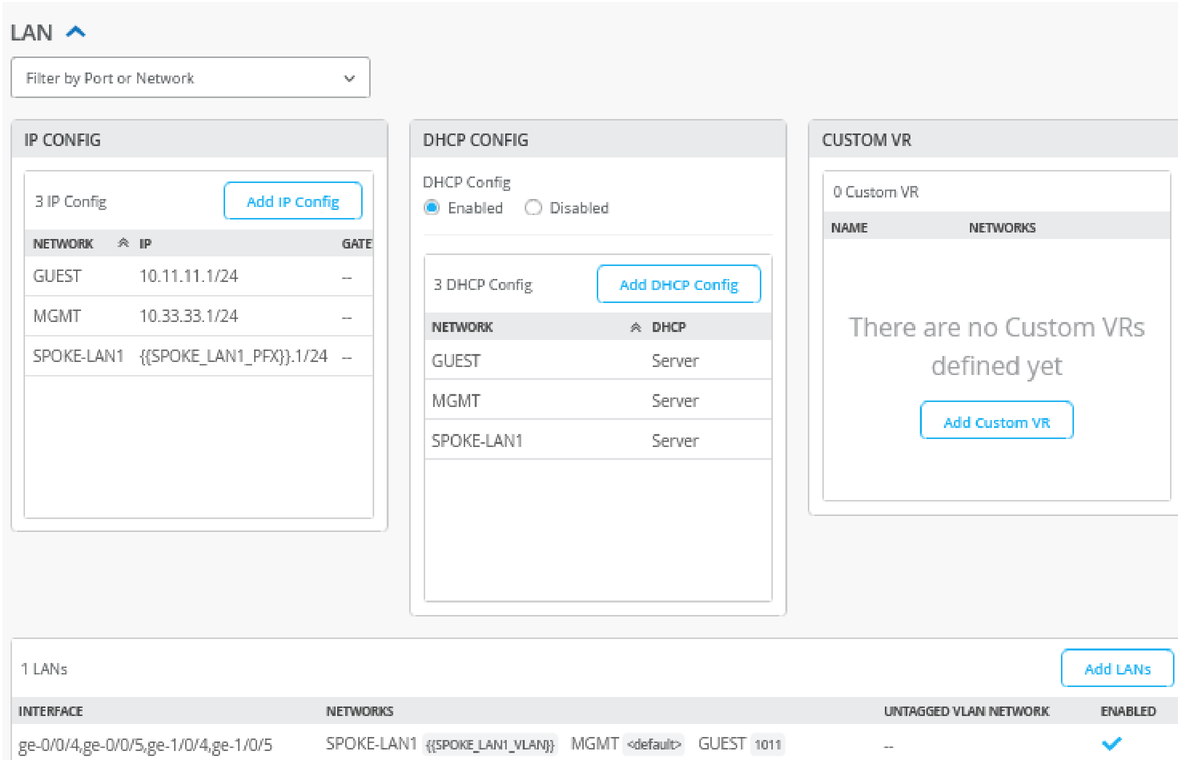
Delete the existing LAN interface configuration and create the following new LAG + redundant interface:

- Interface=`ge-0/0/4,ge-0/0/5,ge-1/0/4,ge-1/0/5`
- Port Aggregation=`Checked/Enabled`
  - Disable LACP=`Unchecked/Disabled`
  - Enable Force Up=`Checked/Enabled`
  - AE Index=0
- Redundant=`Checked/Enabled`
  - `Redundant Index=4`
  - `Primary Node=node0`

- `Networks=SPOKE-LAN1 + GUEST + MGMT`
- `Untagged VLAN Network=None`



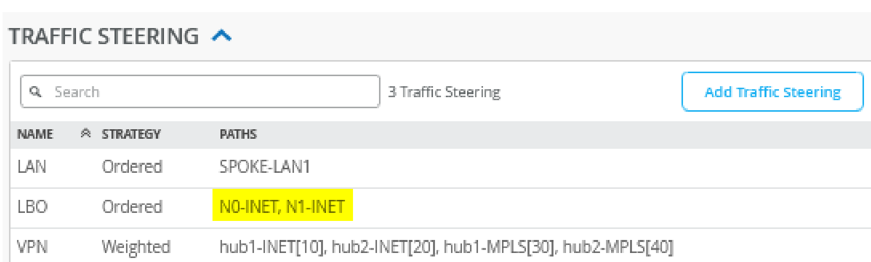
The result should look like the figure below:



You need to change the existing “LBO” traffic steering profile for the Node0 and Node1 redundant WAN interfaces in the following way:

- Name=`LBO`
- Strategy=Ordered
- Paths
  - Path1 Type=`WAN: N0-INET`
  - Path2 Type=`WAN: N1-INET`

The result should look like the figure below:

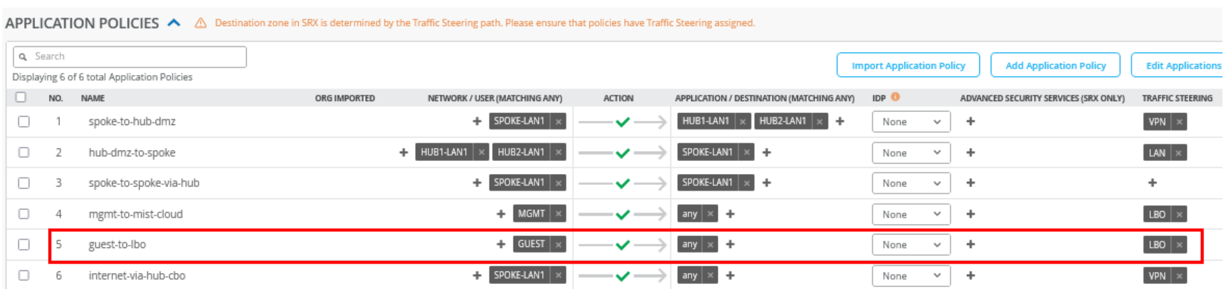


Insert the following application policy:

- Number=`5`

- Name=`guest-to-lbo`
- Network=`GUEST`
- Action=`Pass`
- Application=`any`
- Traffic Steering=`LBO`

The result should look like the figure below:



Remaining Tasks for This Lab

First you need to assign the new spoke templates to their sites.

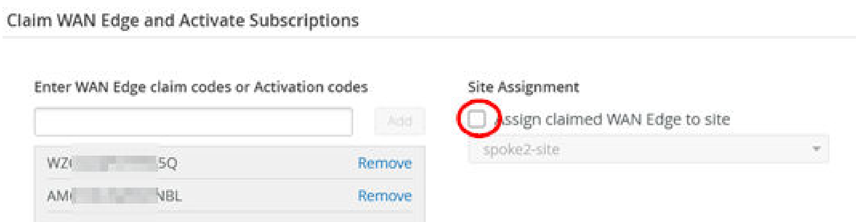
Go to **Organization -> WAN Edge Templates ->** “haspoke-with-lag” template and click **Assign to Sites**. Then, you assign spoke2-site to this template.

If you have the spoke devices in use from previous labs, then go to the inventory and release them. This will bring them back into a factory state.

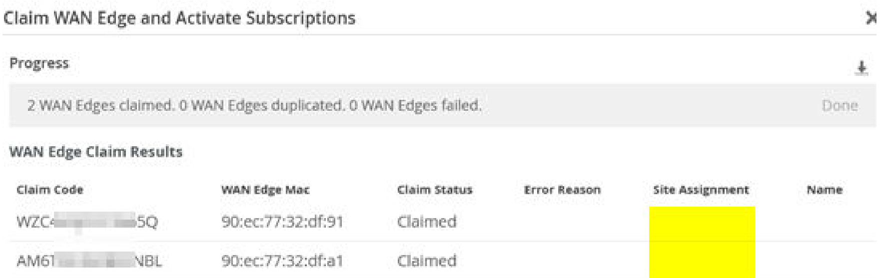
Here, we describe how to build a cluster during the onboarding assuming all devices are in a factory default state.

Go to **Organization -> Inventory** and select **WAN Edges** and click **Claim WAN Edges**. Then, do the following:

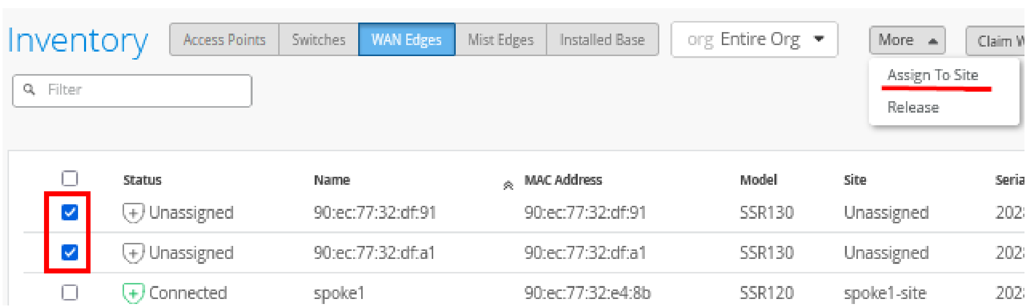
- Assign claimed WAN Edge to site=`Unchecked/Disabled`
- Enter the claim codes for the two devices.



This will claim the two devices without assigning them to a site.

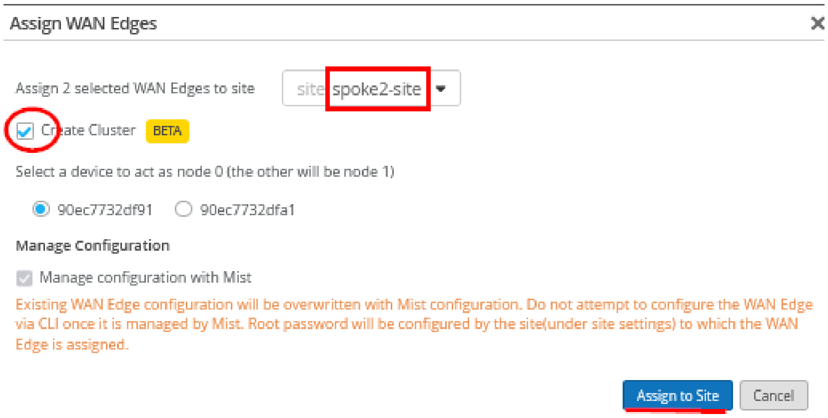


Select the two devices and click on **Assign To Site** from the **More** menu.



Now configure:

- Assign 2 selected WAN Edges to site=`spoke2-site`
- Create Cluster=`Checked/Enabled`
- Select a device to act as node0 = select that as required
- Manage Configuration with Mist=`Enabled` (automatically)



This will commit the needed cluster configuration for the HA Spoke.

Test Your Network Configuration

We are now ready to test our configuration. In our case, you'll notice the following differences compared to the lab setup described in the previous section:

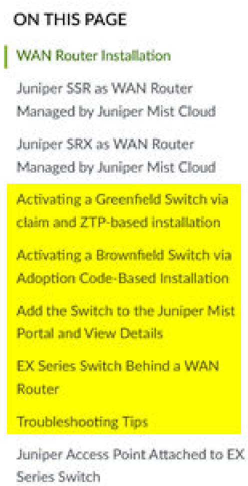
- The reported MAC address of the WAN router will now be `00:00:5e:00:01:01` as the LAG also has a redundant VRRP configuration.
- The Virtual Chassis was already built automatically before, depending on the EX Series Switch model, when following the appropriate power-up sequence. Please review the Day 0 section in the JVD for Distributed Branch EX Series for more information.

With the redundant spoke configuration on Spoke2 in place and a console cable attached to the switch, you can evaluate the following.

```
# ensure you ask for a new DHCP-Lease
root@switch1> restart dhcp-service
Junos Dynamic Host Configuration Protocol process started, pid 55092
#
# wait a few seconds
#
# review your routing table
root@switch1> show route
.
inet.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
Limit/Threshold: 32768/32768 destinations
+ = Active Route, - = Last Active, * = Both
```

```
.
0.0.0.0/0          *[Access-internal/12] 00:00:14, metric 0
                    >  to 10.33.33.1 via irb.0
10.33.33.0/24      *[Direct/0] 00:00:14
                    >  via irb.0
10.33.33.11/32     *[Local/0] 00:00:14
                       Local via irb.0
.
#
# review MAC-Table
root@switch1> show ethernet-switching table
.
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static, C -
Control MAC
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC
           GBP - group based policy)
.
Ethernet switching table : 3 entries, 3 learned
Routing instance : default-switch
    Vlan                MAC                 MAC         Age   GBP     Logical
NH        RTR
    name                address             flags             Tag     interface
Index     ID
    default             00:00:5e:00:01:01   D              -           ge-0/0/6.0
0         0
    default             d4:20:b0:01:46:81   D              -           ge-0/0/3.0
0         0
    default             d4:20:b0:01:46:bb   D              -           ge-1/0/3.0
0         0
#
# review IP address via ARP from WAN-Router received from an interface
root@switch1> show arp no-resolve
MAC Address       Address         Interface                Flags
00:00:5e:00:01:01 10.33.33.1      irb.0 [ge-0/0/6.0]       none
#
# confirm DNS and internet access
root@switch1> ping www.google.com inet
PING www.google.com (172.217.12.100): 56 data bytes
64 bytes from 172.217.12.100: icmp_seq=0 ttl=110 time=10.619 ms
64 bytes from 172.217.12.100: icmp_seq=1 ttl=110 time=11.276 ms
64 bytes from 172.217.12.100: icmp_seq=2 ttl=110 time=7.940 ms
^C
```

The test above shows that the switch obtained a DHCP lease and should be able to initiate traffic with the Juniper Mist cloud to be managed. The remaining steps to onboard an EX Series Switch are explained in the JVD Distributed Branch EX Series. In the Day 1 section, review the sections shown in the figure below:



This section does not repeat the traffic topology tests, as the changes introduced are minimal. For detailed testing procedures, please refer to the "Test Your Network Configuration" on page 71 section in the first topology.

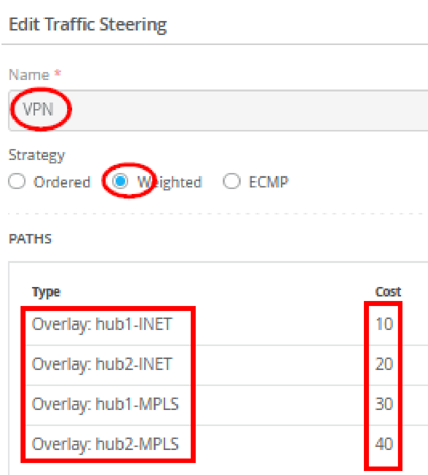
Appendix: Topology Optimizations, Enhancements, and Extensions Valid for All Topologies

IN THIS SECTION

The test cases for this JVD, described in this appendix, are optional extensions to the five validated topologies. While they are not required to establish a basic VPN between branch spokes and hubs, they serve as supplemental optimizations commonly implemented in practice. Review these cases and apply them as needed based on your specific use case.

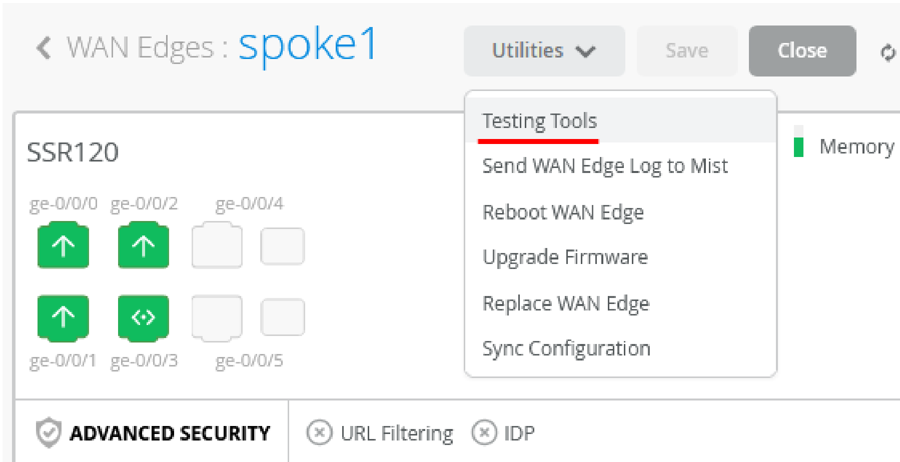
Changing the Hub Used for Central Breakout When Traffic Destination Is "Any"

When we created the WAN edge spoke template for the first lab above "Create the WAN Edge Template for Spokes" on page 53 we defined a weighted VPN traffic steering policy that looked like the figure below:

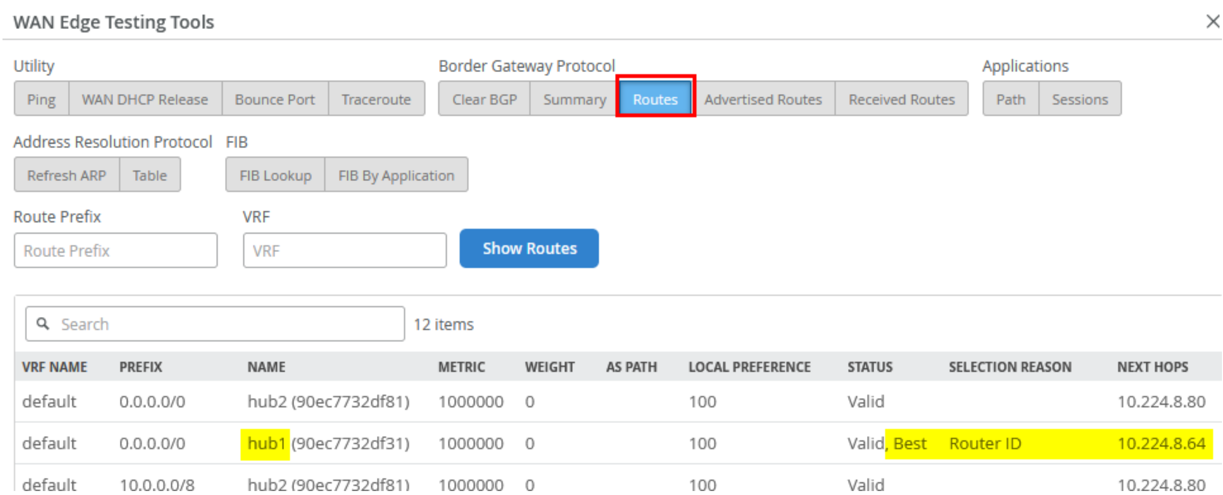


Next, we applied Rule 4 from the application policies to steer all traffic not destined for the VPN toward the two hubs for centralized Internet breakout, using "any" as the traffic destination:

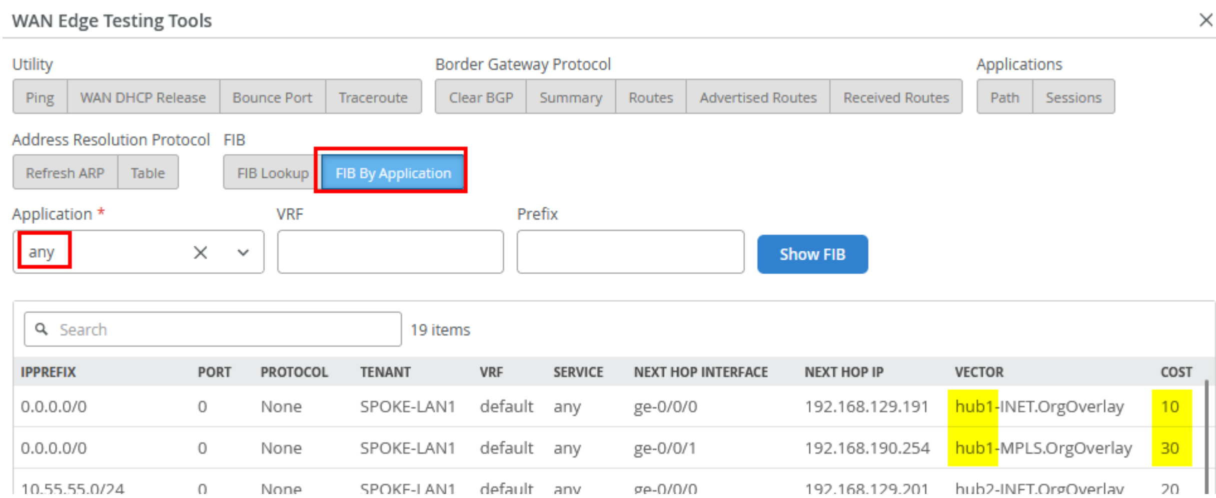
When you go to an individual spoke device under the **Testing Tools** dropdown:



Upon inspecting the BGP routes, you'll notice that two default routes are received—one from each hub—but currently, only the route from Hub1 is being used:



Why this route is selected is also explained because Hub1 in our case has the neighbor IP `10.224.8.64` which is lower than `10.224.8.80` from hub2. This is a normal process for BGP when there is a tie for the same IP prefix obtained. As a consequence, review the resulting forwarding table (FIB) via **FIB By Application** selecting “any” as Application.

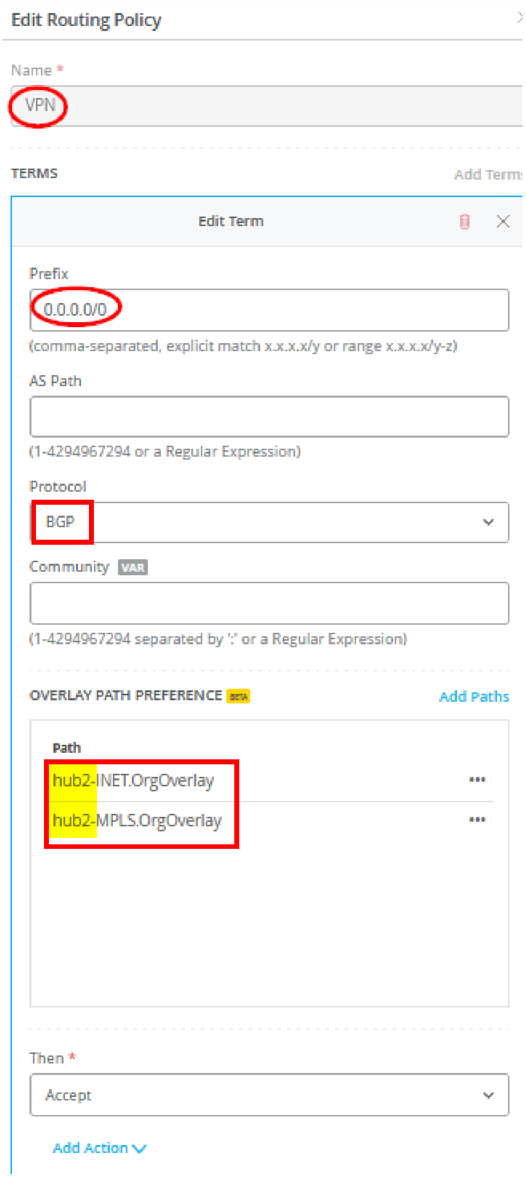


Currently, only two routes are installed—those received from the hub selected as the default based on the lowest router ID. The routes from Hub2 will only be used if the primary hub becomes unavailable. This behavior is intentional, as we want to avoid frequent switching between hubs for this `0.0.0.0/0` traffic. Keep in mind that each hub performs source NAT for Internet-bound traffic during central breakout. If traffic were to alternate between hubs, applications on the Internet could see the same VPN client appearing to come from different public IP addresses, which can cause issues.

However, the router ID is typically not configurable and is often assigned automatically during device installation. In our case, Hub1 was installed before Hub2, but that doesn't guarantee Hub1 will have the lower router ID.

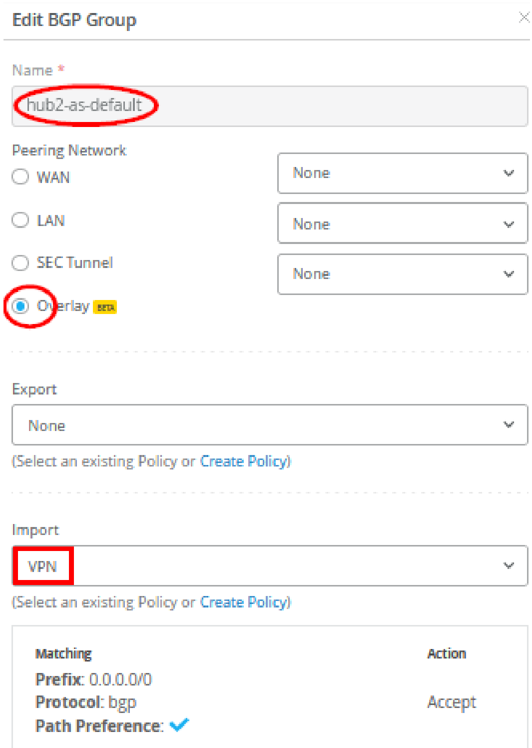
If, for any reason, you want Hub2 to be preferred over Hub1—even if Hub2 has a higher router ID—you can achieve this by modifying the spoke template. To do so, add a new routing policy with the following configuration:

- Name=`VPN`
- Prefix=`0.0.0.0/0`
- Protocol=`BGP`
  - Overlay Path Preference. Do not use any Hub1 path here due to router ID.
  - Path1=`hub2-INET.OrgOverlay`
  - Path2=`hub2-MPLS.OrgOverlay`
- Then=`Accept`



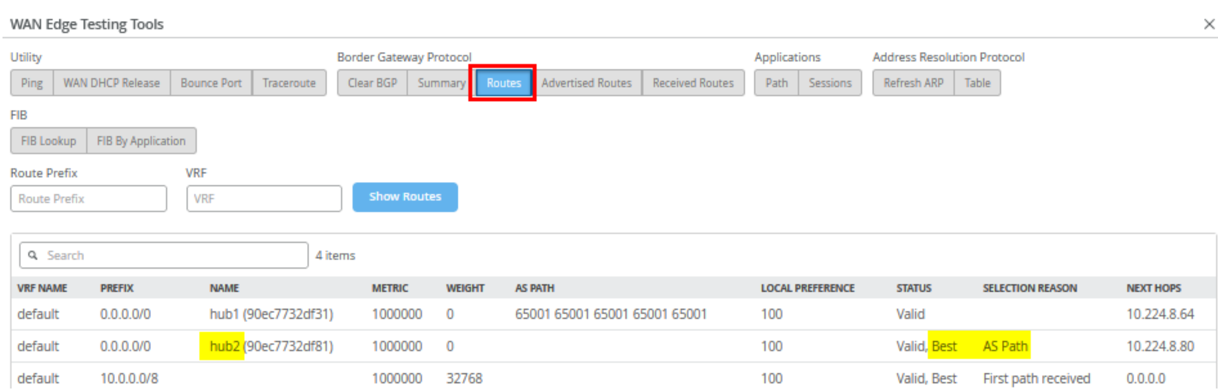
Then add a new BGP group as follows:

- Name=`hub2-as-default`
- Overlay=`Checked/Enabled`
- Export=`None`
- Import=`VPN`

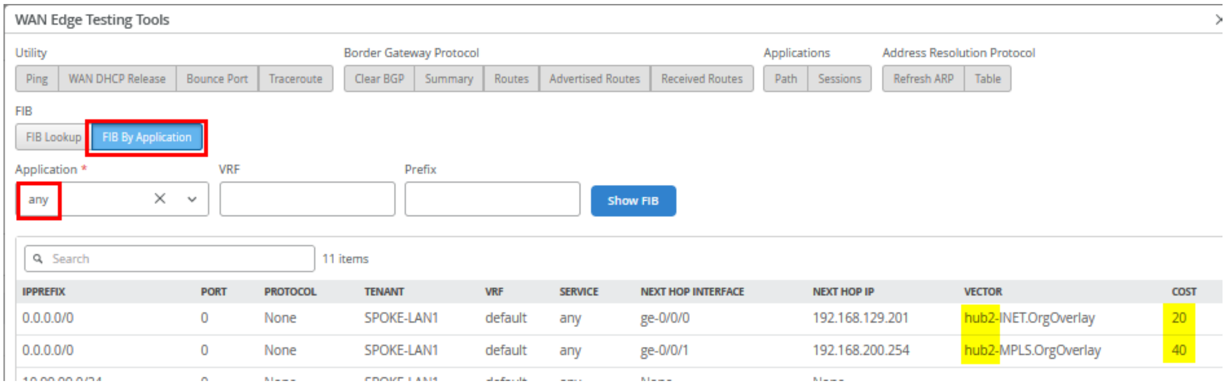


Save your template so that it gets applied to your spokes.

When you go back to the spoke and review the routes again, you will see that Hub2 now offers the best routes by BGP AS Path selection.

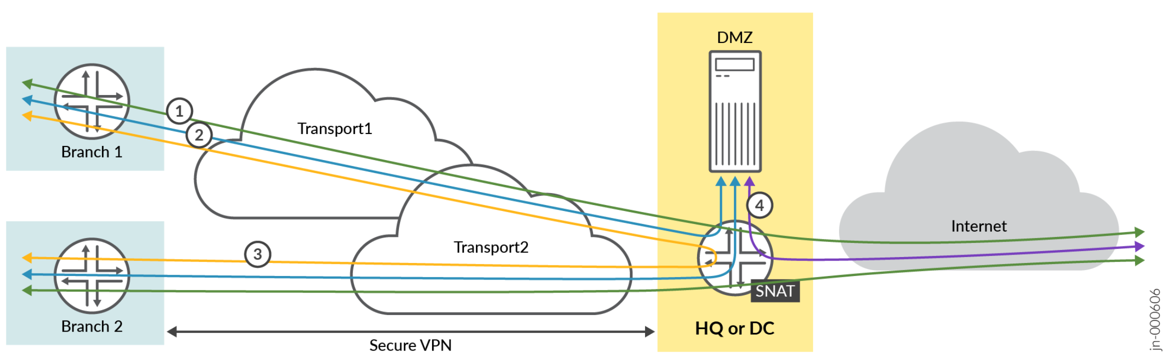


Upon rechecking the forwarding table (FIB), you'll see that the change we made now selects Hub2 as the destination for this traffic—exactly the outcome we intended through our local BGP route manipulation.



Local Traffic Breakout at the Spoke

In this section, we will modify the default forwarding behavior of the VPN that was configured in the first lab "Appendix: Building a base SD-WAN Topology with Three Spokes and Two Hubs" on page 23. Remember, the model used routes all Internet-bound traffic through the hub, which then performs central breakout using source NAT before forwarding it to the Internet.



However, this configuration may not align with customer-specific designs or application requirements for Internet access. In some cases, it may be preferable for the spoke to handle local internet breakout through its own WAN interfaces, performing source NAT for the following reasons:

- Reduced Latency: Local breakout at the spoke typically provides a shorter and more direct path to Internet applications compared to routing through the hub for central breakout.
- Optimized VPN Resources: By offloading Internet-bound traffic locally, more bandwidth and resources are available for the VPN overlay.
- Traffic Isolation: You may not want traffic from spoke-local networks—such as an isolated Wi-Fi guest network not advertised in the VPN overlay—to traverse the VPN. This is similar to the guest network

example described in the lab "Appendix: Building an Extended Full Stack Topology with Juniper EX Switch Virtual Chassis and SSR HA Cluster" on page 177 example above.

Using the lab scenario "Appendix: Building a base SD-WAN Topology with Three Spokes and Two Hubs" on page 23 as a foundation, this section demonstrates how to configure:

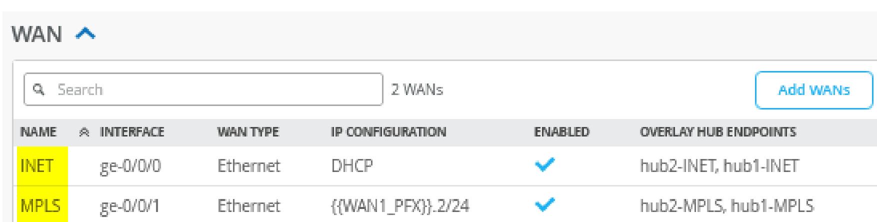
- Local breakout at the spoke via WAN interfaces with source NAT.
- Local breakout for all Internet traffic instead of routing it through the hub for central breakout.
- Local breakout for specific applications identified by DNS, while keeping central breakout as the default for all other traffic.

Local breakout for an isolated LAN network can be reviewed when following the instructions for lab "Appendix: Building a Full Stack Topology with Juniper EX Switch and Juniper AP" on page 155 and lab "Appendix: Building an Extended Full Stack Topology with Juniper EX Switch Virtual Chassis and SSR HA Cluster" on page 177 already.

> **NOTE**: The examples for local breakout can be extended by the information shared below in section "Advanced Application Steering" on page 229. This section is not intended to present a full list of possible configurations.

Checking the WAN Interfaces to Perform Source NAT

First, check your existing WAN interface configuration in the spoke WAN Edge template. In our case, we have two called `INET` and `MPLS` as shown in the figure below:



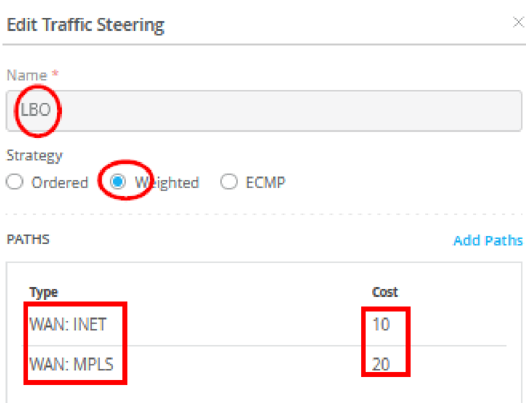
Every interface that we want to utilize for local breakout must have the following configured:

- Source NAT=`Interface`

Configure an LBO Traffic Steering Policy

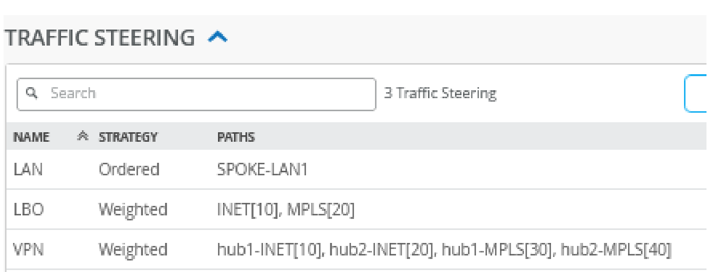
In our case, we want to use the Internet WAN interface as a primary interface for local breakout and the MPLS interface as a secondary interface. Hence, we configure an additional traffic steering policy in the spoke WAN Edge template:

- Name=`LBO`
- Strategy=`Weighted`
- Paths
  - Type1=`WAN: INET`
  - Cost1=`10`
  - Type2=`WAN: MPLS`
  - Cost2=`20`



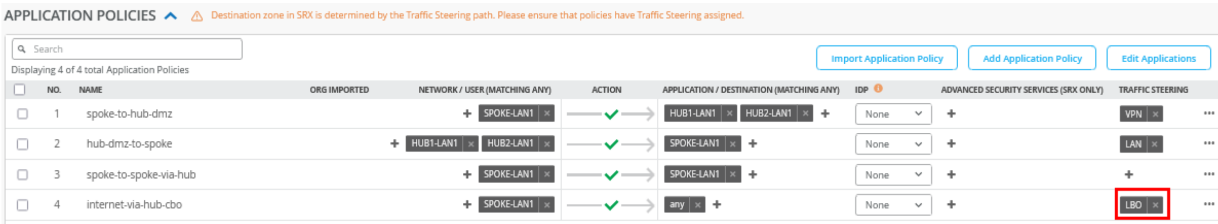
NOTE: When configuring the traffic steering policy for local breakout using WAN interfaces, avoid using the ECMP option. ECMP can cause traffic to be load-balanced across multiple WAN interfaces on a per-flow basis, potentially resulting in different public IP addresses due to varying Internet paths. This behavior can lead to issues with applications that expect a consistent source IP address..

The result should look like the figure below:

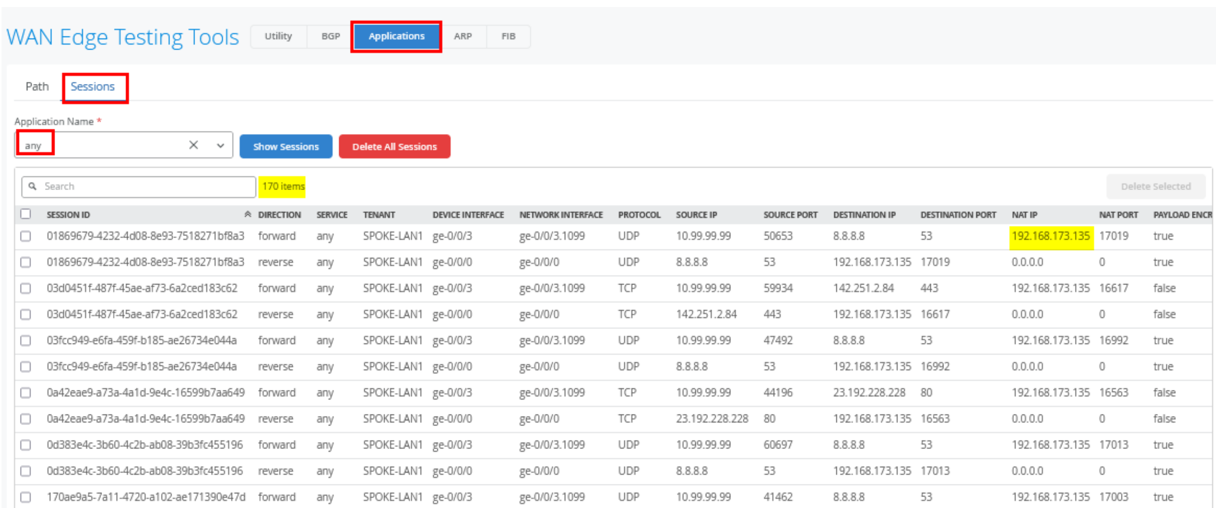


Local Breakout of all Internet Traffic Instead of Central Breakout

For this example, you just need to change Rule 4 from `VPN` to `LBO` as indicated in the figure below:



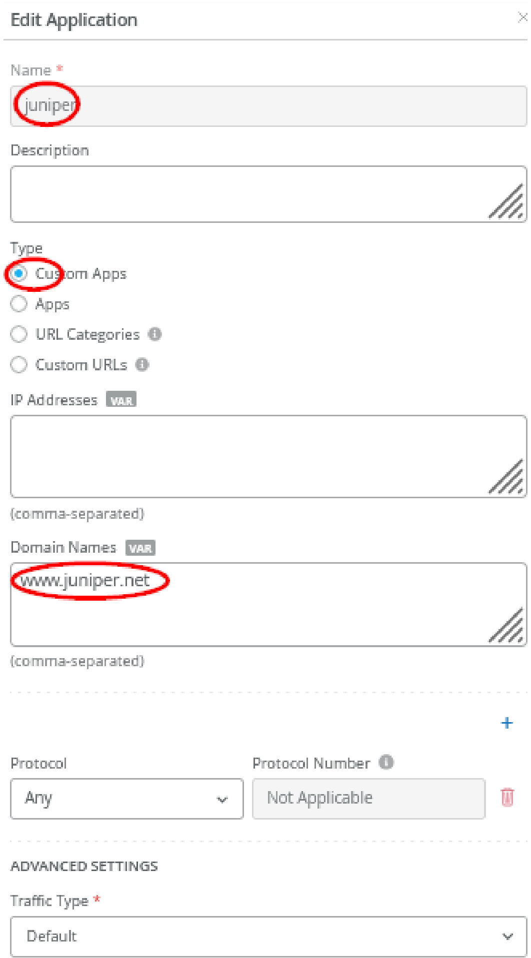
Once this configuration change is applied, generate internet-bound traffic from clients connected to your spokes—such as the desktop1 VM connected to Spoke1. After initiating traffic, use the spoke's **Testing Tools** to analyze sessions under **Applications -> Session** with Application Name=`any`, as shown in the figure below. For a busy website like reddit.com, you'll see a large number of captured sessions. The NAT IP allows you to infer which interface was used for source NAT.



Local Breakout for a Custom Application Identified by DNS

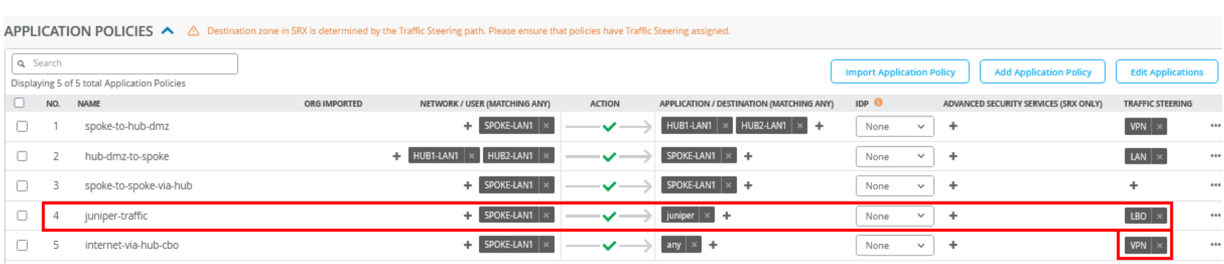
In this example, we create a new application which we will identify by DNS and then send its traffic towards the local breakout. Go to **Organization -> Applications** and configure the following:

- Name=`juniper`
- Type=`Custom Apps`
- Domain Names=`www.juniper.net`



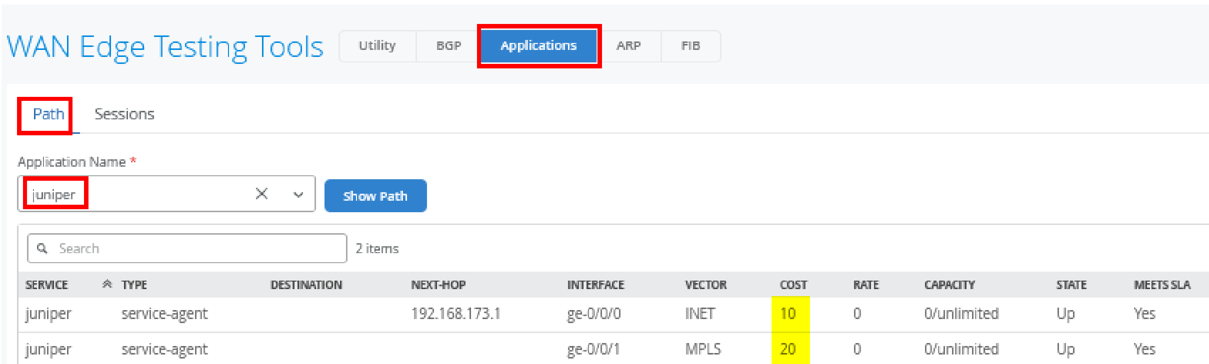
Then, go to your existing WAN Edge template and insert the following application policy:

- Number=`4`
  - Name=`juniper-traffic`
  - Network=`SPOKE-LAN1`
  - Action=`Pass`
  - Application=`juniper`
  - Traffic Steering=`LBO`

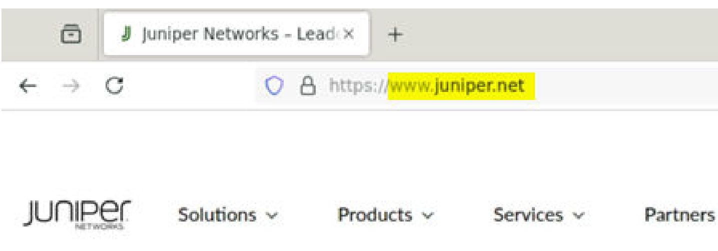


We also reverted the “any” rule to use the VPN for central breakout. Apply this change.

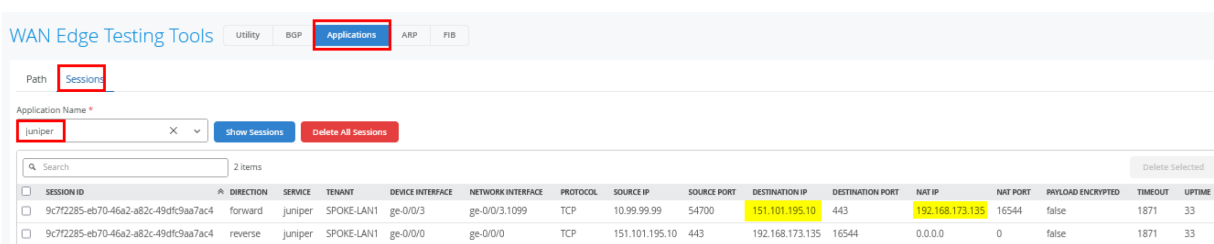
On your WAN Edge spoke, navigate to the **Testing Tools** section and select **Applications -> Path** with Application Name=`juniper`, as shown in the figure below. This view displays the configured WAN interfaces used for the traffic along with their associated path costs.



Now, we need to generate traffic using this custom application. In our example, we use the desktop1 VM attached to Spoke1 and pointing a browser towards https://www.juniper.net.

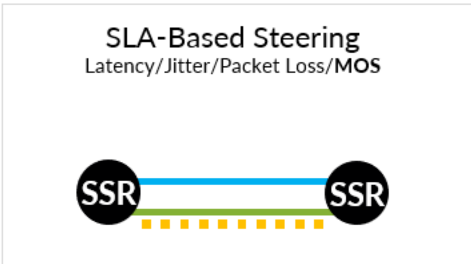


Using **Testing Tools** again, select **Applications -> Sessions** with Application Name=`juniper` as indicated in the figure below. You should see a few sessions using this path now (most of the content comes from CDN which we did not configure here).



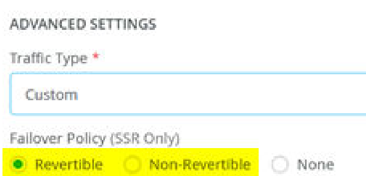
Traffic Path SLA-Based Failover

When deploying your SD-WAN VPN with multiple paths, you can leverage the SLA-based failover mechanism provided by the Session Smart Router. The router continuously monitors latency, jitter, and packet loss on each path. Administrators can define traffic SLAs using custom or pre-configured thresholds for these metrics. Based on real-time measurements, the system can detect when a path no longer meets the defined SLA. If an alternate path remains within acceptable thresholds, the traffic is automatically rerouted to maintain performance and avoid service degradation.



Except for the predefined “any” application, you can define such traffic SLAs under **Advanced Settings** by changing the traffic type from the default setting.

Changing the default traffic type value enables you to influence the traffic failover policy indicated in the figure below:



- **Revertible** means that the traffic after an SLA-based failover will switch back to the old path once the SLA for that path has recovered.
- **Non-Revertible** means that the traffic after an SLA-based failover will stay on the new path until it experiences an SLA violation, and a new failover decision needs to be made.
- **None** disables SLA-based failovers.

The table below outlines the predefined traffic types available for selection, or you can choose the custom option to define your own.

Traffic Type	Traffic Class	DSCP Class	Max. Latency	Max. Jitter	Max. Loss
Custom	Default=Best Effort	Default=8	Custom defined	Custom defined	Custom defined

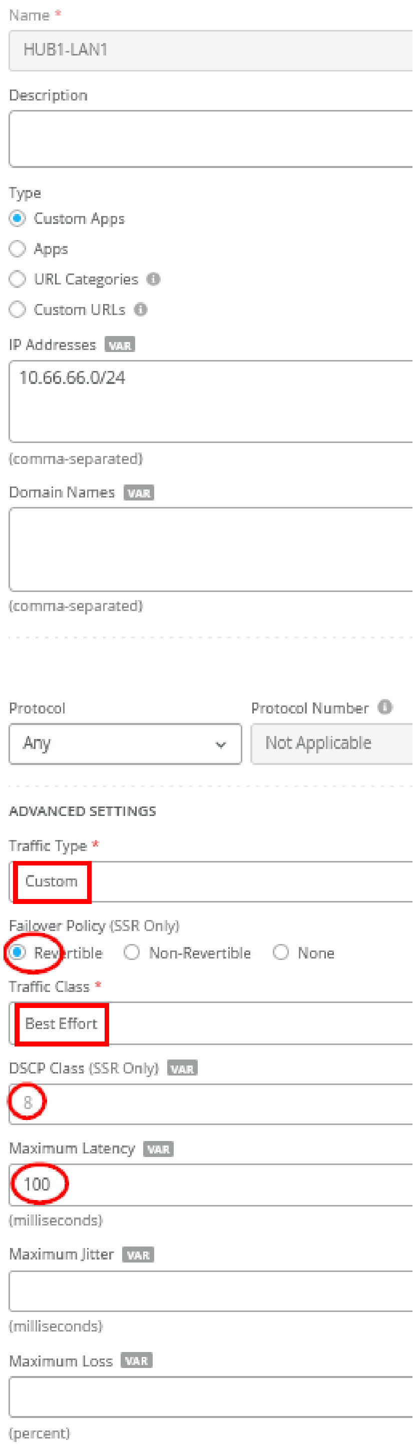
(Continued)

Traffic Type	Traffic Class	DSCP Class	Max. Latency	Max. Jitter	Max. Loss
Data Best Effort	Low	0	1625	N/A	30
Data Interactive	Medium	18	600	N/A	30
Data Mission Critical	Medium	26	750	N/A	25
Data Scavenger	Best Effort	8	1625	N/A	30
Gaming	Medium	18	500	200	25
Management Interactive	Medium	16	650	N/A	25
Management M2M	Medium	16	1000	N/A	N/A
Remote Desktop	Medium	32	1300	500	25
Video Streaming	Medium	26	3000	200	30
Video Streaming Scavenger	Best Effort	8	3000	250	35
VoIP Audio	High	46	150	30	35
VoIP Signaling	Medium	40	250	N/A	N/A
VoIP Video	Medium	32	1500	250	35

Here is an example of how you can test this: Go to **Organization -> Applications** and edit the existing Application `HUB1-LAN1` in the following way:

- Traffic Type=`Custom`
- Failover Policy=`Revertible`
- Traffic Class=`Best Effort`

- DSCP Class=`8`
- Maximum Latency=`100`



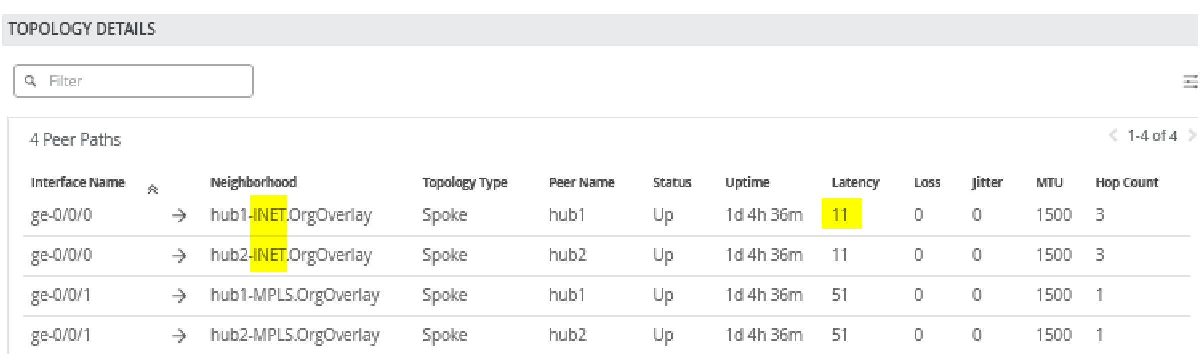
Save your configuration so that it gets applied.

We shall now continue our testing on the clients attached to the spokes. We connect to the desktop1 VM with IP address `10.99.99.99` attached to Spoke1 and set up a continuous ping to desktop4 VM with IP address 10.66.66.66 which is attached to Hub1. This generates traffic utilizing the SLA.

```
root@desktop1:~# ping 10.66.66.66
PING 10.66.66.66 (10.66.66.66) 56(84) bytes of data.
64 bytes from 10.66.66.66: icmp_seq=1 ttl=59 time=12.7 ms
64 bytes from 10.66.66.66: icmp_seq=2 ttl=59 time=11.4 ms
64 bytes from 10.66.66.66: icmp_seq=3 ttl=59 time=11.3 ms
64 bytes from 10.66.66.66: icmp_seq=4 ttl=59 time=11.3 ms
64 bytes from 10.66.66.66: icmp_seq=5 ttl=59 time=11.2 ms
64 bytes from 10.66.66.66: icmp_seq=6 ttl=59 time=11.2 ms
64 bytes from 10.66.66.66: icmp_seq=7 ttl=59 time=11.3 ms
64 bytes from 10.66.66.66: icmp_seq=8 ttl=59 time=11.1 ms
64 bytes from 10.66.66.66: icmp_seq=9 ttl=59 time=11.3 ms
64 bytes from 10.66.66.66: icmp_seq=10 ttl=59 time=11.2 ms
64 bytes from 10.66.66.66: icmp_seq=11 ttl=59 time=11.3 ms
64 bytes from 10.66.66.66: icmp_seq=12 ttl=59 time=11.2 ms
64 bytes from 10.66.66.66: icmp_seq=13 ttl=59 time=11.2 ms
64 bytes from 10.66.66.66: icmp_seq=14 ttl=59 time=11.4 ms
64 bytes from 10.66.66.66: icmp_seq=15 ttl=59 time=11.3 ms
64 bytes from 10.66.66.66: icmp_seq=16 ttl=59 time=11.2 ms
64 bytes from 10.66.66.66: icmp_seq=17 ttl=59 time=11.1 ms
```

When you review the current topology report you will see the following:

- Both paths toward the hubs connected through the simulated internet have about 11ms latency. That will change in the next step.
- Both paths toward the hubs connected through MPLS will have about 51ms latency. We won't change that throughout this exercise, hence all traffic with around 50ms will indicate that the MPLS path was used.



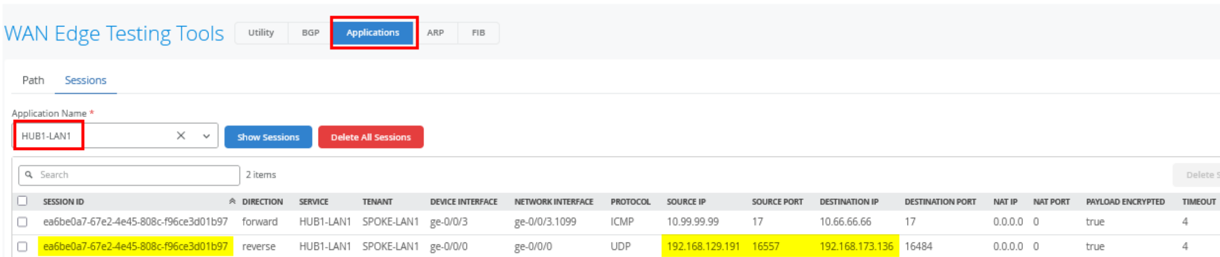
TOPOLOGY DETAILS

Filter

4 Peer Paths — 1-4 of 4

Interface Name		Neighborhood	Topology Type	Peer Name	Status	Uptime	Latency	Loss	Jitter	MTU	Hop Count
ge-0/0/0	→	hub1-INET.OrgOverlay	Spoke	hub1	Up	1d 4h 36m	11	0	0	1500	3
ge-0/0/0	→	hub2-INET.OrgOverlay	Spoke	hub2	Up	1d 4h 36m	11	0	0	1500	3
ge-0/0/1	→	hub1-MPLS.OrgOverlay	Spoke	hub1	Up	1d 4h 36m	51	0	0	1500	1
ge-0/0/1	→	hub2-MPLS.OrgOverlay	Spoke	hub2	Up	1d 4h 36m	51	0	0	1500	1

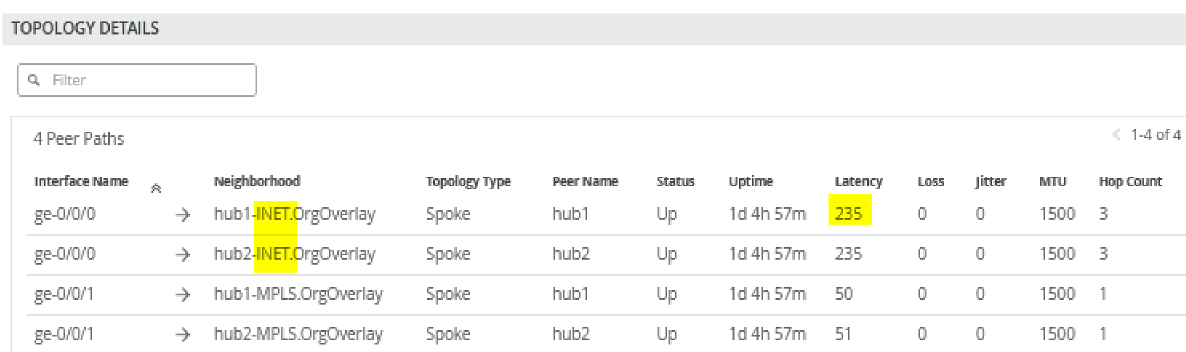
When you open **Testing Tools** and navigate to **Applications -> Sessions** with Application Name=`HUB1-LAN1`, you'll see in the return flow the source and destination IP addresses used between the hub and spoke over the underlay WAN for Internet traffic.



At this point, we changed the latency of the Internet path from 10 to 234ms, which you can see in the VPN overlay.

```
root@desktop1:~# ping 10.66.66.66
PING 10.66.66.66 (10.66.66.66) 56(84) bytes of data.
.
64 bytes from 10.66.66.66: icmp_seq=18 ttl=59 time=235 ms
64 bytes from 10.66.66.66: icmp_seq=19 ttl=59 time=235 ms
64 bytes from 10.66.66.66: icmp_seq=20 ttl=59 time=235 ms
64 bytes from 10.66.66.66: icmp_seq=21 ttl=59 time=236 ms
.
.
64 bytes from 10.66.66.66: icmp_seq=50 ttl=59 time=235 ms
64 bytes from 10.66.66.66: icmp_seq=51 ttl=59 time=235 ms
64 bytes from 10.66.66.66: icmp_seq=52 ttl=59 time=235 ms
64 bytes from 10.66.66.66: icmp_seq=53 ttl=59 time=235 ms
```

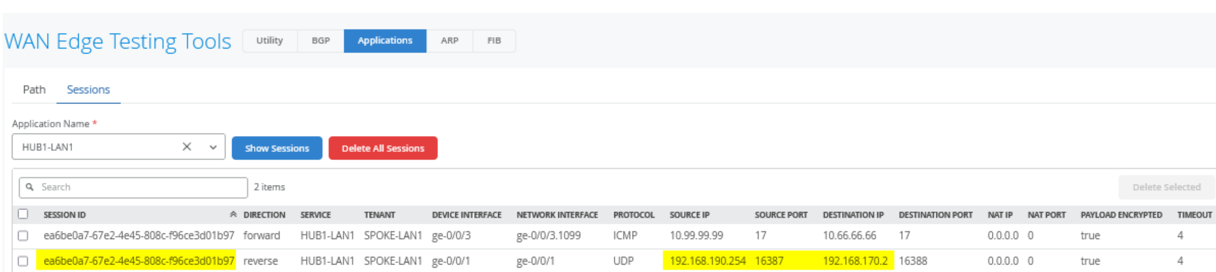
Again, the topology view indicates the higher latency for the Internet path due to our change.



After about 35 seconds, you see a change in latency to about 51ms which means the traffic has switched to the secondary MPLS path which has this latency in our lab.

```
root@desktop1:~# ping 10.66.66.66
PING 10.66.66.66 (10.66.66.66) 56(84) bytes of data.
.
64 bytes from 10.66.66.66: icmp_seq=54 ttl=61 time=52.9 ms
64 bytes from 10.66.66.66: icmp_seq=55 ttl=61 time=50.9 ms
64 bytes from 10.66.66.66: icmp_seq=56 ttl=61 time=51.1 ms
.
.
64 bytes from 10.66.66.66: icmp_seq=97 ttl=61 time=51.2 ms
64 bytes from 10.66.66.66: icmp_seq=98 ttl=61 time=51.3 ms
64 bytes from 10.66.66.66: icmp_seq=99 ttl=61 time=51.1 ms
# this is the point on time we did heal the original internet path back to 10ms
64 bytes from 10.66.66.66: icmp_seq=100 ttl=61 time=51.2 ms
64 bytes from 10.66.66.66: icmp_seq=101 ttl=61 time=51.3 ms
64 bytes from 10.66.66.66: icmp_seq=102 ttl=61 time=51.3 ms
64 bytes from 10.66.66.66: icmp_seq=103 ttl=61 time=51.3 ms
.
.
64 bytes from 10.66.66.66: icmp_seq=146 ttl=61 time=51.3 ms
64 bytes from 10.66.66.66: icmp_seq=147 ttl=61 time=51.2 ms
64 bytes from 10.66.66.66: icmp_seq=148 ttl=61 time=51.2 ms
64 bytes from 10.66.66.66: icmp_seq=149 ttl=61 time=51.2 ms
```

When you open **Testing Tools** and review **Applications -> Sessions** with Application Name=`HUB1-LAN1`, you'll see in the return flow the source and destination IP addresses used between the hub and spoke over the underlay WAN for MPLS traffic. The session ID is still the same as well.



We recover the Internet path around ping sequence 99, and after about 50 seconds, the traffic is switched back to the original path since our failover policy is set to revertible and the latency returns back to about 11ms.

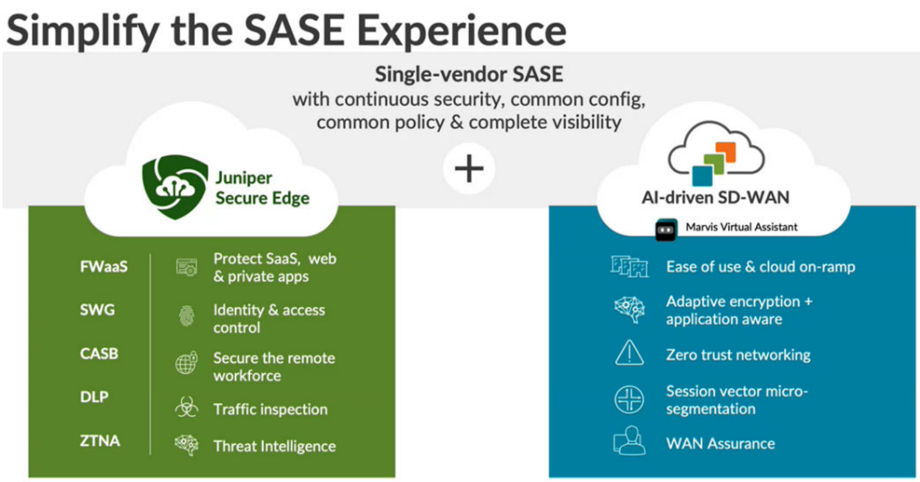
```
root@desktop1:~# ping 10.66.66.66
PING 10.66.66.66 (10.66.66.66) 56(84) bytes of data.
.
64 bytes from 10.66.66.66: icmp_seq=150 ttl=59 time=12.6 ms
64 bytes from 10.66.66.66: icmp_seq=151 ttl=59 time=11.2 ms
64 bytes from 10.66.66.66: icmp_seq=152 ttl=59 time=11.3 ms
64 bytes from 10.66.66.66: icmp_seq=153 ttl=59 time=11.2 ms
64 bytes from 10.66.66.66: icmp_seq=154 ttl=59 time=11.2 ms
64 bytes from 10.66.66.66: icmp_seq=155 ttl=59 time=11.2 ms
64 bytes from 10.66.66.66: icmp_seq=156 ttl=59 time=11.3 ms
```

NOTE: Failover and revert times depend on how much the measured performance deviates from the defined SLA. The greater the margin of violation, the more aggressively the system responds. However, the system avoids reacting too aggressively to normal path fluctuations.

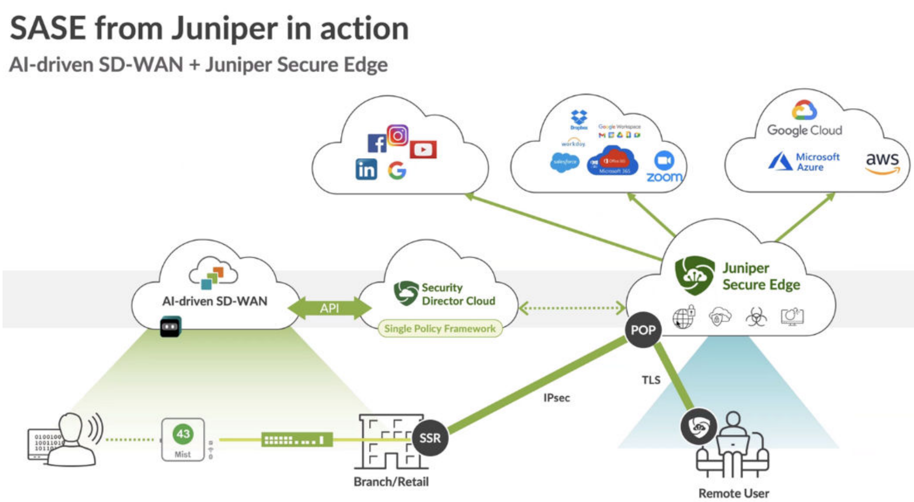
Secure Edge Connector

Juniper® Secure Edge provides full-stack security service edge (SSE) capabilities to protect access to web, SaaS, and on-premises applications. These capabilities also provide consistent threat protection, an optimized network experience, and security policies that follow users wherever they go. Secure Edge acts as an advanced cloud-based security scanner. It enables organizations to protect data and provide users with consistent, secure network access whether users are in the office, on campus, or on the move.

Juniper Mist works with Juniper Secure Edge by providing a Secure Edge Connector (SEC) that can establish a secure tunnel with the Juniper Secure Edge cloud service.



Secure Edge capabilities are all managed by Juniper Security Director Cloud, Juniper's simple and seamless management experience delivered in a single user interface.



For more information, see Juniper Secure Edge.

Secure Edge Connector Overview

The Juniper Mist cloud works with Secure Edge to perform traffic inspection from edge devices by using the Secure Edge connector feature. This feature allows the SRX Series Firewall, deployed as a WAN edge device, to send a portion of traffic to Secure Edge for an inspection.

With this solution, you send the Internet-bound traffic from the LAN side of a spoke or hub device to Secure Edge for inspection before the traffic reaches the Internet.

To perform traffic inspection by Secure Edge:

- In Security Director Cloud, create and configure the service locations, IPsec profiles, sites, and policies for Secure Edge. These are the cloud-based resources that provide security services and connectivity for the WAN edge devices.

- In Juniper Mist cloud, create and configure the WAN edge devices, such as Session Smart Router that connect to the LAN networks. These are the devices that provide routing, switching, and SD-WAN capabilities for the branches or campuses.

- On the Juniper Mist WAN edge, create and configure the Secure Edge tunnels that connect the WAN edge devices to the service locations. These are the IPsec tunnels that provide secure and reliable transport for the traffic that needs to be inspected by Secure Edge.

- In Juniper Mist cloud, assign the Secure Edge tunnels to the sites or device profiles that correspond to the WAN edge devices. This enables the traffic steering from the LAN networks to the Secure Edge cloud-based on the defined data policies and other match criteria.

Before You Begin

- Read about the Secure Edge subscription requirements. See Juniper Secure Edge Subscriptions Overview.

- Ensure that you have completed the prerequisites to access the Juniper Security Director Cloud portal. See Prerequisites.

- Create your Secure Edge tenant. See Create Your Secure Edge Tenant.

- Assume that, in the Juniper Mist cloud, you have adopted and configured the WAN edge devices, such as Session Smart Routers that connect to the LAN networks.

Access Juniper Security Director Cloud and Check Active Subscriptions

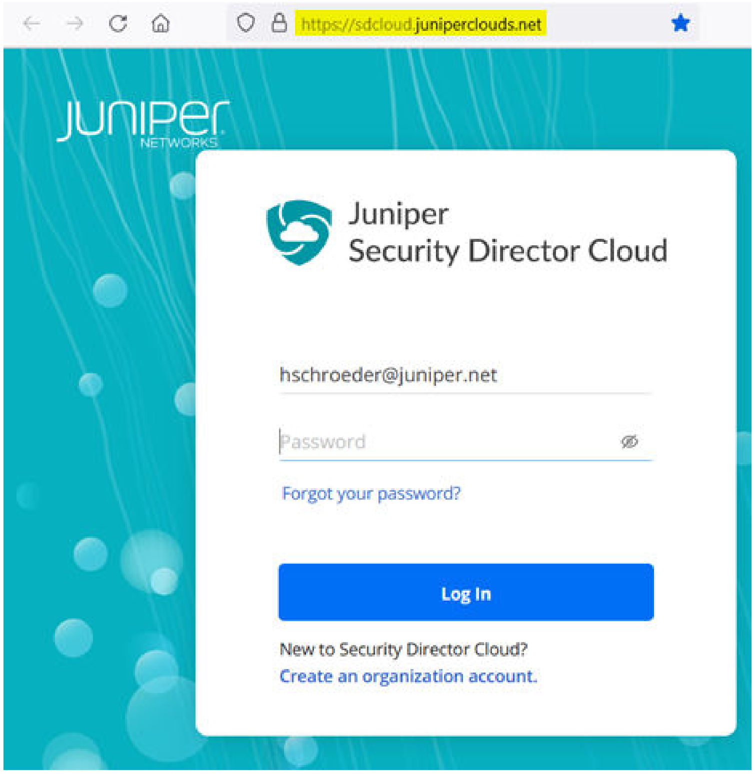
A tenant in Secure Edge is an organization account that you create to access the Juniper Security Director Cloud portal and manage your Secure Edge services. A tenant is associated with a unique e-mail address and a subscription plan. A tenant can have multiple service locations, which are vSRX based WAN edge devices hosted in a public cloud for your organization.

A tenant can have one or more service locations, which are the connection points for end users. To create a tenant, you need to have an account on the Juniper Security Director Cloud. See Create Your Secure Edge Tenant for details.

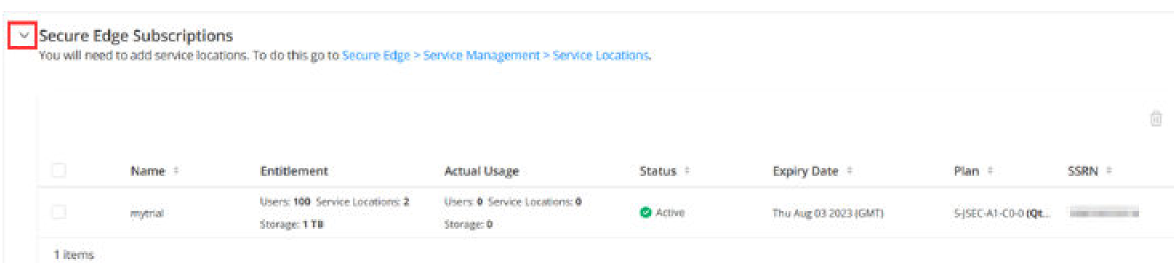
After you create your Secure Edge tenant in the Juniper Security Director Cloud portal, access the portal and check your subscriptions.

To access Juniper Security Director Cloud and check active subscriptions:

- Open the URL to the Juniper Security Director Cloud. Enter your e-mail address and password to log in and start using the Juniper Security Director Cloud portal.



- Select the required tenant in the upper-right corner of the portal to continue.
- Select **Administration -> Subscriptions** to access the Juniper Security Director Cloud subscriptions page.



- Scroll to the Secure Edge subscriptions section to check whether you have an active subscription. For details, see About the Subscriptions Page. If you have active subscriptions, continue with the next steps.

Generate Device Certificates in Juniper Security Director Cloud

Now that you have configured service locations in Juniper Security Director Cloud, generate device certificates to secure network traffic.

You use a Transport Layer Security/Secure Sockets Layer (TLS/SSL) certificate to establish secure communications between Secure Edge and WAN edge devices. All the client browsers on your network must trust the certificates signed by Juniper Networks and the SRX Series Firewalls to use an SSL proxy.

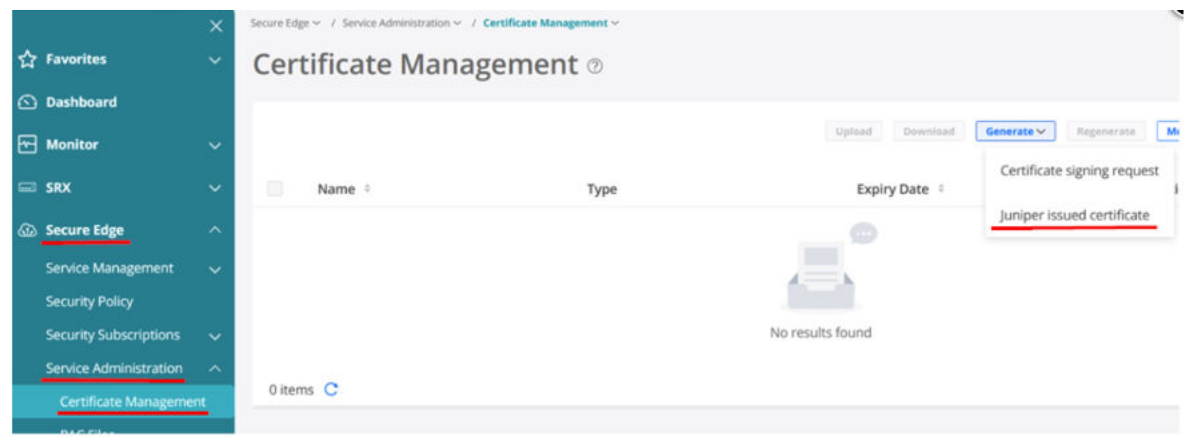
In Juniper Security Director Cloud, you have the following choices for generating certificates:

- Create a new certificate signing request (CSR), and your own certificate authority (CA) can use the CSR to generate a new certificate.
- Select the option to have Juniper Networks create a new certificate.

> **NOTE**: This topic describes how to generate a TLS/SSL certificate. How you import and use the certificate depends on your company's client-management requirements and is beyond the scope of this topic.

To generate device certificates in Juniper Security Director Cloud:

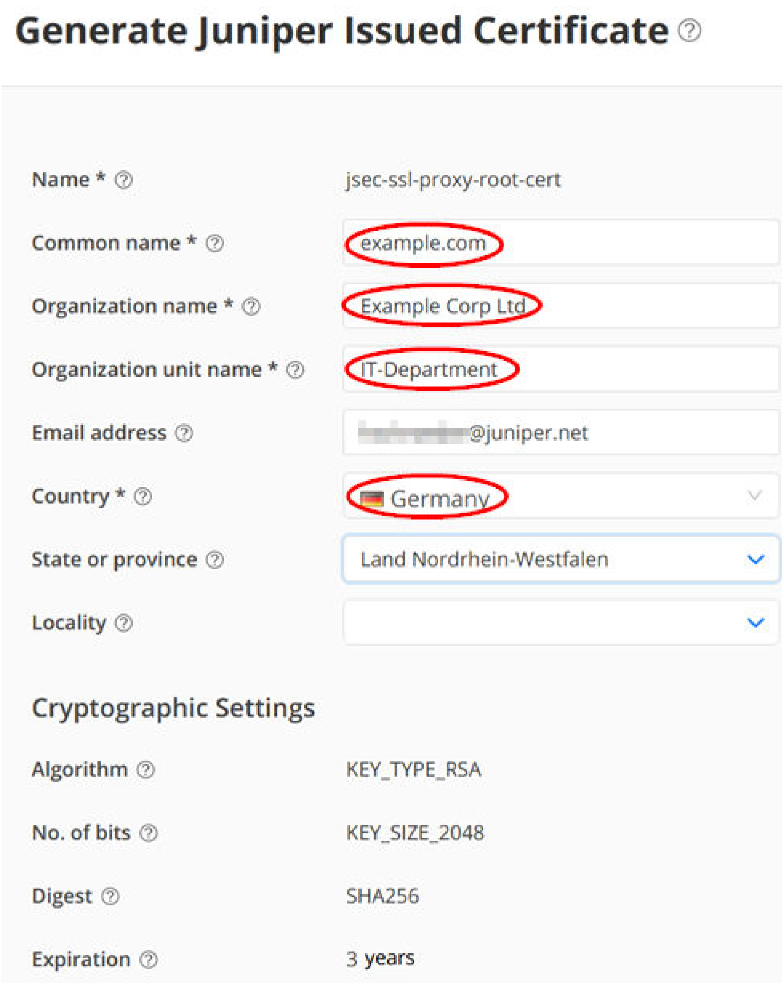
- Select **Secure Edge** > **Service Management** > **Service Administration** > **Certificate Management**.
- The Certificate Management page appears.
  - From the Generate list, you can generate either a new CSR, or a Juniper-issued certificate.



- Select the relevant option:
  - If your company has its own CA and you want to generate a CSR, click **Certificate signing request**.

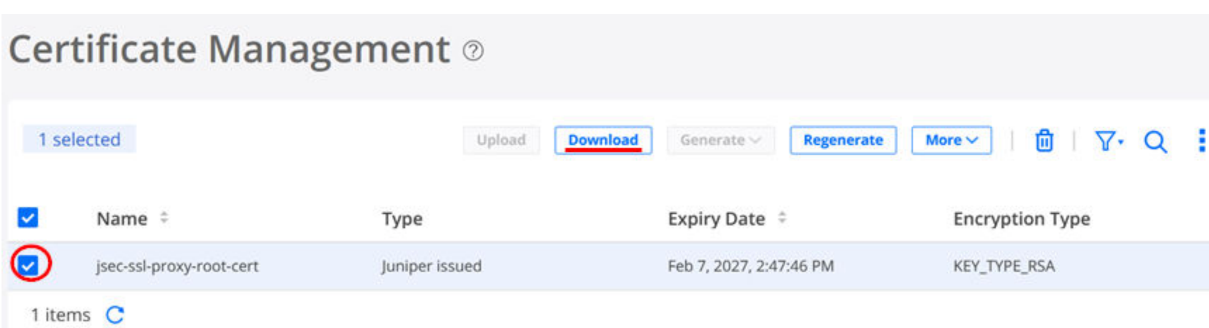
    After Secure Edge generates a CSR, download the CSR and submit it to your CA to generate a new certificate. Once generated, click **Upload** to upload the certificate on the Certificate Management page.

  - If your company does not have its own CA, click **Juniper issued certificate**, and then click **Generate** to generate the certificate. Juniper Networks will generate and keep the certificate on the system. In this task, select Juniper issued certificate and continue with next step.
- Enter the certificate details. In the **Common name** field, use the certificate's fully qualified domain name (FQDN).



The Certificate Management page opens with a message indicating that the certificate is created successfully.

- Download the generated certificate.



The following sample shows the downloaded certificate:

```
-----BEGIN CERTIFICATE-----
MIIG4jCCBMqgAwIBAgIIX3yPMZ7QT9MwDQYJKoZIhvcNAQEMBQAwgYgxCzAJBgNV
BAYTAlVTMQswCQYDVQQIEwJDQTESMBAGA1UEBxMJU3Vubnl2YWxlMR4wHAYDVQQK
.
.
JwePvBrmKGPph8k+8gL9Gqw+wnfaARP3fqp4TXUcp6twDMyP0OJR8tRm51keplVw
RAfTzy91Bhf261E62+MzKeh8J0Wi8q8Amaw6+aNVj8TcA9T/zotCI5JSkqV6+Wap
btLaf5DXSYliXWnDgt72sURF3bmUYjfDTmPgwzeMi/dal4IWUqk=
-----END CERTIFICATE-----
```

After you download the certificate to your system, add the certificate to client browsers.

Configure a Service Location in Juniper Security Director Cloud

After ensuring that you have an active license to Juniper Security Director Cloud, you configure a service location. This is your first main task in setting up a Secure Edge connector for Session Smart Router.

A service location in Juniper Security Director Cloud is also known as POP (point of presence) and represents a Secure Edge instance in a cloud location. The service location is the connection (access) point for both on-premises and roaming users.

Service locations are places where vSRX creates secure connections between different networks using a public cloud service. The public IP address (unique per tenant and service location) is used to:

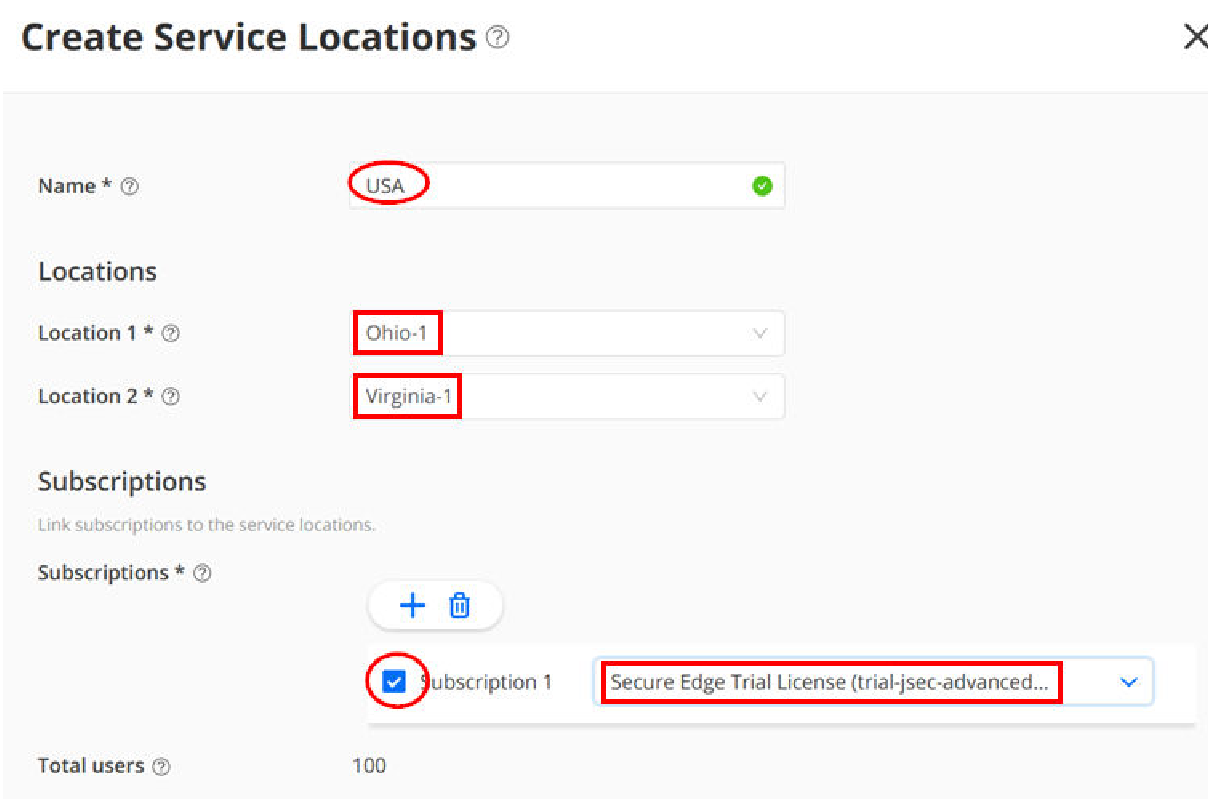
- Set up an IPsec tunnel between the branch device and the Juniper Security Director Cloud.
- Centrally distribute the traffic when the destination is on the Internet.

To configure a service location in Juniper Security Director Cloud:

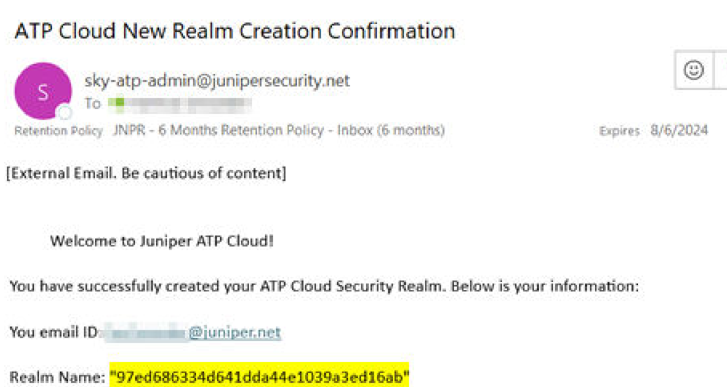
- In the Juniper Security Director Cloud menu, select **Secure Edge** > **Service Management** > **Service Locations**.
- The Service Locations page appears. Click the Add (+) icon to create a new service location. Enter the details for the following fields:
  - **Name** — Give a name like “USA” below.
  - **Location 1** — Select the location for the Secure Edge in the region.
  - **Location 2** — Select the location for the Secure Edge in the region. Ensure that it is not another instance in the same region as Location 1. You usually want a backup in case the entire region fails.

- **Subscriptions** — Select at least one subscription which has a minimum of 100 Users.

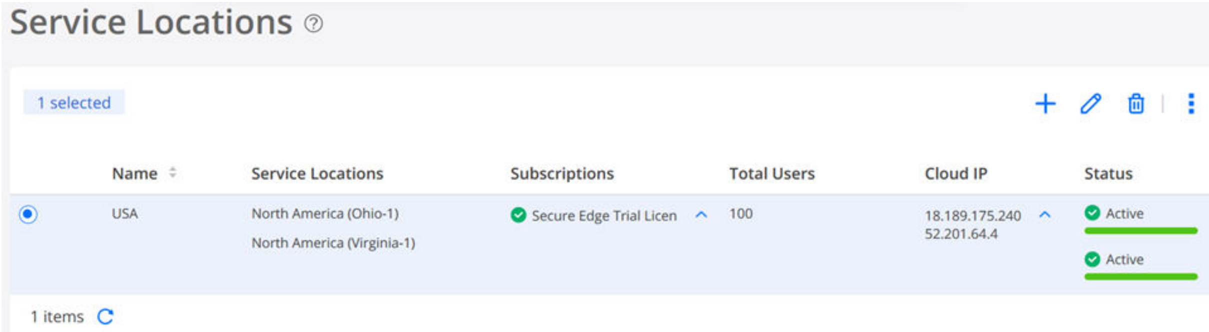
The figure below shows examples of service locations:



- Click OK. Security Director Cloud creates a new service location and lists it on the Service Locations page.
- You will receive an email confirming your action like as shown in the example below:



The status of the service location shows **In Progress** until the Secure Edge instance is fully deployed, as shown in the figure below:

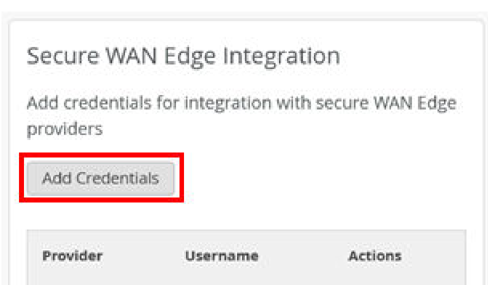


When you create a new service location, the system starts the deployment of two vSRX instances as WAN edges for your tenant system. In this deployment, vSRX instances are not shared with other tenants.

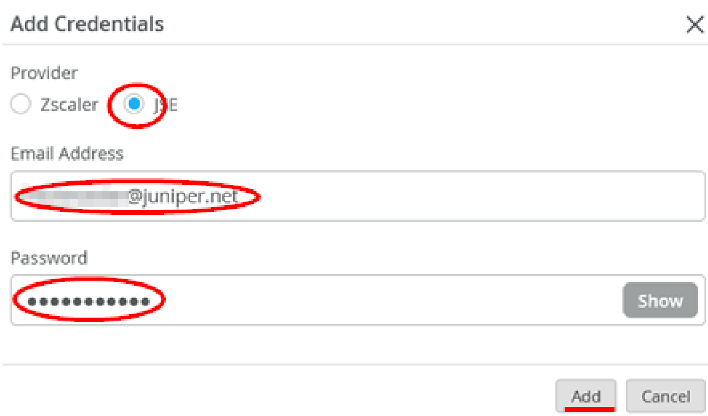
We suggest that you review the security policies https://sdcloud.juniperclouds.net/secure-edge/secure-edge-policy of your tenant. You may need to make changes to allow or deny certain Internet traffic. For basic troubleshooting, it's recommended to allow ICMP pings to the internet. This enables you to easily verify reachability and measure path latency from a branch-connected client to the intended internet destination, when traffic is redirected through Secure Edge to the Juniper Security Director Cloud environment. An example of such a configuration is given here.

Add Juniper Security Director Cloud Account Credentials to the Mist Cloud Organization

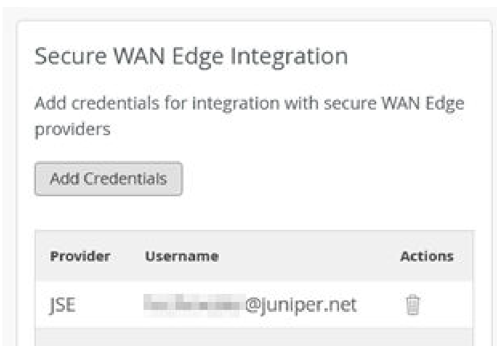
- Go to **Organization > Settings**
- Click on **Add Credentials**



- Fill in your Credentials
  - Set Provider: "JSE"
  - Add your own email address on the Juniper Security Director Cloud instance
  - Add your own password on the Juniper Security Director Cloud instance



The result of should look similar to the figure below:



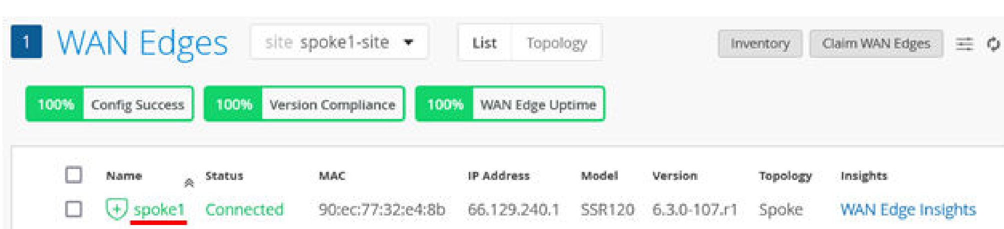
Create Secure Edge Connectors in the Juniper Mist Cloud Portal

You create Secure Edge connectors in the Juniper Mist cloud portal. This task completes the configuration on the Mist cloud side of the tunnels to establish an IPsec tunnel between WAN edge devices managed by Juniper Mist and Security Director Cloud. Before you create the connectors, ensure that your site has a deployed Session Smart Router.

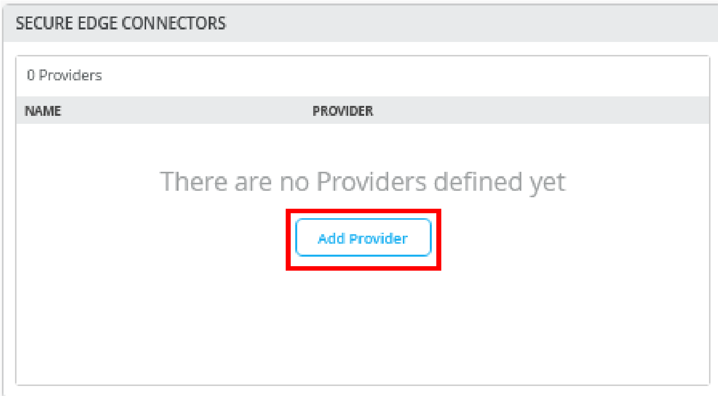
To create Secure Edge connectors:

- In the Juniper Mist cloud portal, click **WAN Edges**.

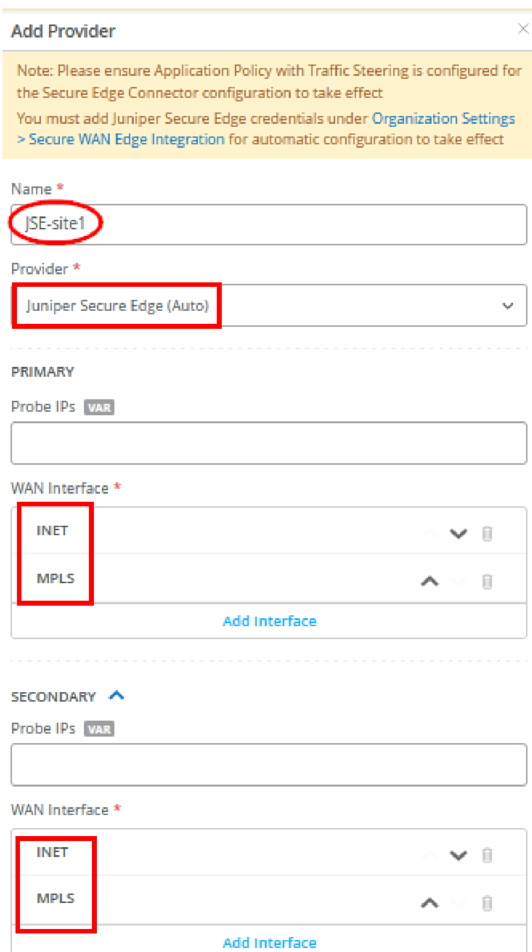
The WAN Edges page displays site details.



- Select a site with a deployed branch device.
- In the Secure Edge Connectors pane, click **Add Provider**.

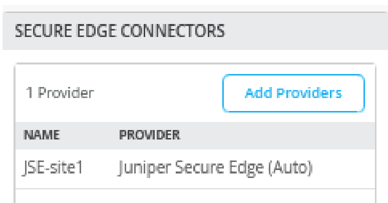


- Enter the Secure Edge connector details as shown in the figure below. Under **Provider**, select **Juniper Secure Edge (Auto)**, and specify the WAN interfaces to be used for connecting to the two service locations. This setup is very similar to how two hubs are defined for standard VPN connections.

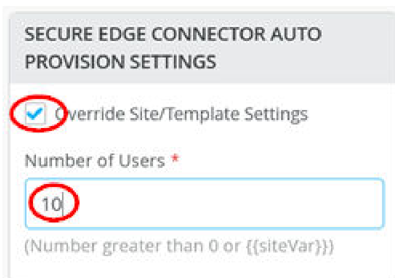


NOTE: You don't need to enter the probe IP values. IPsec tunnels do not need additional monitoring like GRE needs.

- Verify that the Juniper Mist cloud portal has added the Secure Edge connector you just configured.

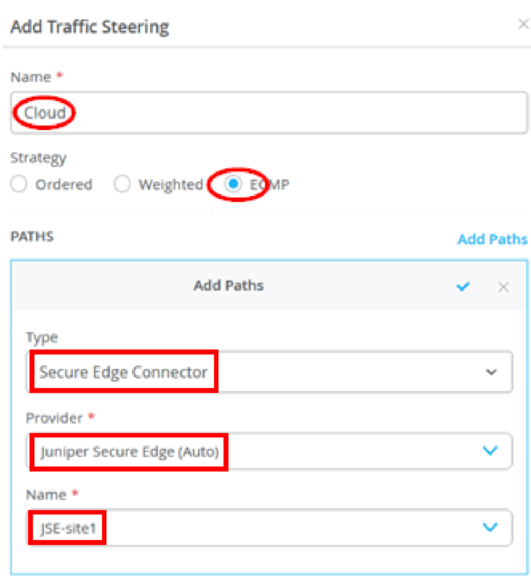


- Next, add a few user sessions to your Secure Edge Connector

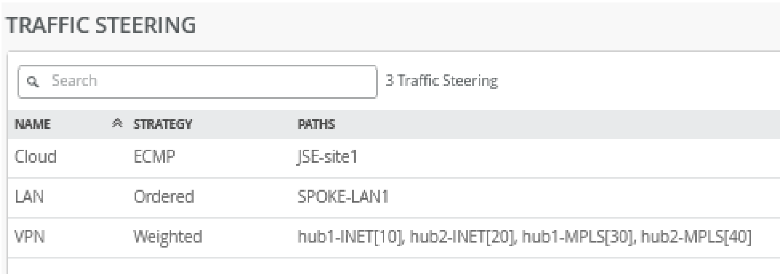


- Add the traffic steering paths.

Add a new traffic steering path on the WAN edge template or WAN edge device, according to the values provided in the figure below:



- The figure below displays the configured traffic steering paths:


TRAFFIC STEERING

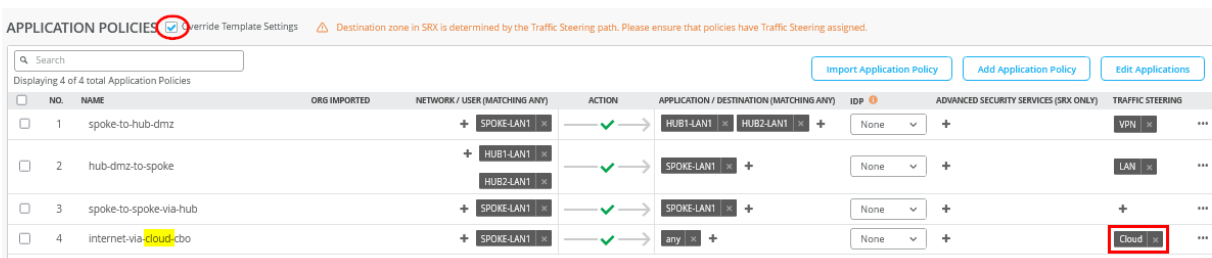
3 Traffic Steering

NAME	STRATEGY	PATHS
Cloud	ECMP	JSE-site1
LAN	Ordered	SPOKE-LAN1
VPN	Weighted	hub1-INET[10], hub2-INET[20], hub1-MPLS[30], hub2-MPLS[40]

Modifying the Application Policies

After you create Secure Edge connectors in the Juniper Mist portal, the next step is to modify application policies on the branch device. For example, you already allow traffic from a spoke device to a hub device and vice versa. You can also allow traffic from a spoke device to another spoke device in the VPN tunnel. After that, you can send traffic from spokes to the Internet through Juniper Security Director Cloud instead of sending traffic from spokes to a hub for central breakout.

In the example shown below, we modify, using **Override Template Settings**, the policy rule set so that instead of central breakout at the hub, all branch traffic towards the Internet gets shifted to the Secure Edge in the cloud.



- Select the policy that you want to modify and apply the following changes:
  - Check the **Override Template Settings** option.
  - Change the traffic steering to “Cloud” in the last rule `internet-via-cloud-cbo` .
- **Save** your changes.

Juniper Mist cloud builds new tunnels to Juniper Security Director Cloud.

Verify the Configuration

After you modify the application policy, you confirm that your configuration is working as expected.

With the desired configuration saved, you can verify if Juniper Mist cloud routes the Internet-bound traffic from spokes to Juniper Security Director Cloud instead of routing it to a hub for central breakout.

To verify the configuration:

- Verify the WAN Insights of the device's established tunnels in the Juniper Mist cloud portal.

SECURE EDGE CONNECTOR DETAILS

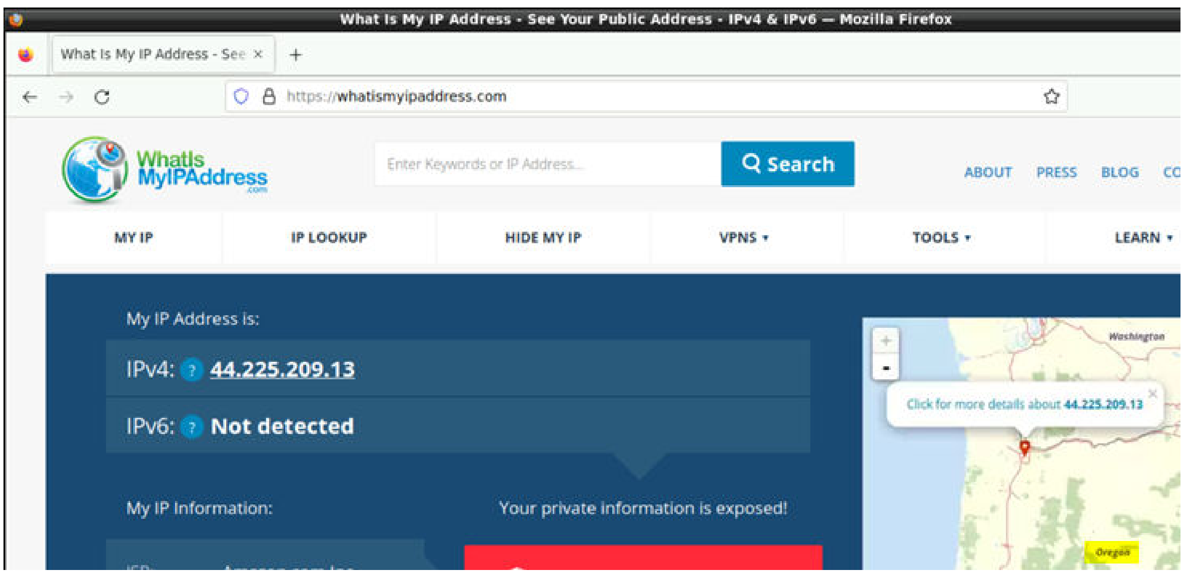
2 Tunnels

Tunnel Name	Peer IP	Status	Last Seen	Node	RX Bytes	TX Bytes	RX Packets	TX Packets	Last Event	Protocol	Uptime
site1-to-sdcloud	3.130.70.175	Connected	10:48 AM Mar 10	standalone	461.4 kB	663 kB	11.0 k	11.0 k	–	IPsec	15h 15m
site1-to-sdcloud	44.225.209.13	Connected	10:48 AM Mar 10	standalone	461.2 kB	662.9 kB	11.0 k	11.0 k	–	IPsec	15h 15m

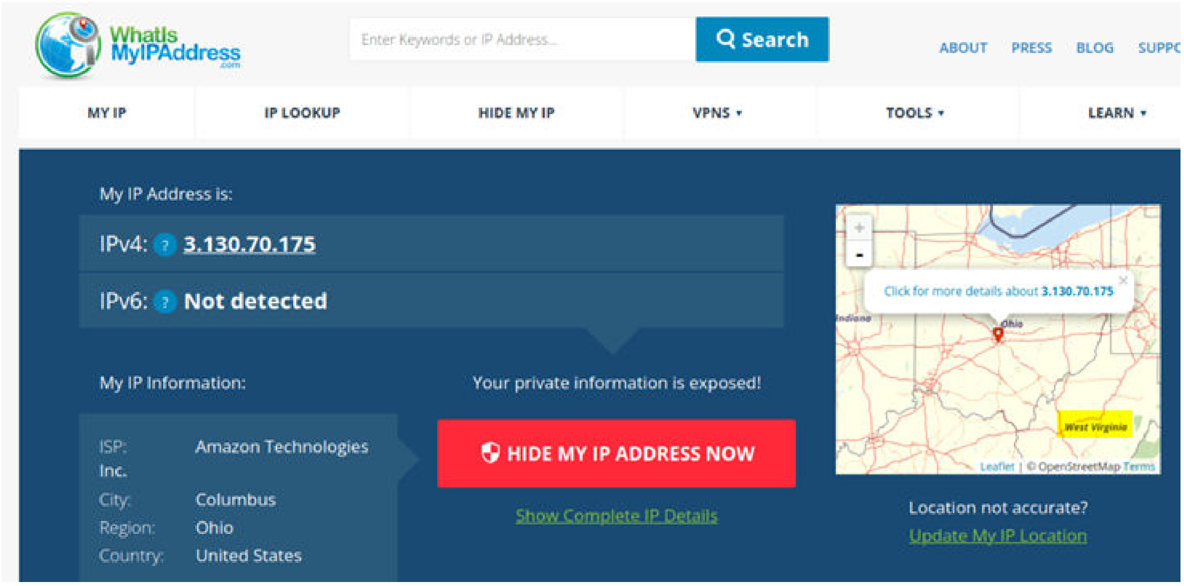
You can also check the established tunnels in the Juniper Security Director Cloud dashboard and in the service location.

- Verify the new traffic flow using a client connected to the LAN interface of the spoke. On the client, open a browser and navigate to https://whatismyipaddress.com/ to view the source IP address being used to route Juniper Mist network traffic from the service location to the internet.

The two figures below show traffic from the primary and secondary service locations:



and



One of the two IP addresses of the service location is a public IP address and serves two purposes:

- Terminates the IPsec tunnel, hence the spoke uses it to establish the tunnel with Juniper Security Director Cloud.
- Acts as a new source IP address for traffic leaving the VPN which we can detect with the above.

Remember that a service location in Juniper Security Director Cloud is also known as a POP and represents a Secure Edge instance in a cloud location. The service location is the connection (access) point for both on-premises and roaming users.

Appendix: Common Test Cases for All Topologies

IN THIS SECTION

The test cases described in this appendix are common among the five topologies found in this JVD.

Advanced Application Steering

Applications represent traffic destinations. In Juniper® Session Smart® Networking, applications determine the traffic destination used in an application policy.

In a Juniper Mist WAN Assurance design, **applications** refer to the services or programs that network users access. These can be manually defined within the Juniper Mist portal by selecting a category (for example, Social Media) or choosing specific applications (for example, Microsoft Teams) from a predefined list. Alternatively, you can use a built-in list of common traffic types or create custom application definitions as needed.

> **NOTE**: When configuring your internal VPN, always start by using IP prefixes in a custom rule to establish the fundamental traffic forwarding policies. Once these basic rules have been tested and verified, you can refine the configuration by incorporating additional criteria to detect and steer specific applications.

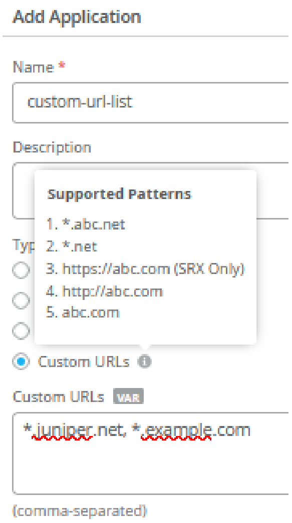
To enable users to access applications, you must first define the applications and then use application policies to control access—either allowing or denying it. This involves associating applications with specific users or networks and assigning both a traffic steering policy and an access rule.

When defining applications, you have the following options defined by their types:

- Using **Custom Apps** as application identifier one can configure:
  - IP address or IP prefix. Enter one or more IP addresses or subnets, separated by commas.
  - Domain Names. Use FQDN-DNS names; multiple entries can be separated by commas.
  - Protocol. TCP, UDP, GRE, or a custom value are allowed.
  - Destination ports. Specify start and end ports, applicable to protocols that support port numbers.
- Using **Apps** as application identifier:
  - You can select each pre-configured and known application individually from the drop-down menu or search for them.
- Using **URL Categories** as application identifier
  - Define applications based on categorized websites using the built-in URL categorization database. You can apply rules using **URL Category Groups**, **URL Categories**, or **URL Subcategories**, depending on the level of granularity required.
  - URL Category Groups are: All, Standard, Strict

- URL Categories are: Adult, Advertisement, Arts and Entertainment, Business, Career and Education, Collaboration, Conferencing, Device IOT, File Sharing, Financial, Games, Government, Images, Infrastructure, Malware, Networking, News and Reference, Recreation, Religion, Remote Desktop, Search Engines, Security, Shopping, Social Media, Software Updates, Sports, Streaming Media, Technology, Violence
- URL Subcategories are: Abortion, Adult Content, Adult Material, Advanced Malware Command and Control, Advanced Malware Payloads, Advertisements, Alcohol and Tobacco, Alternative Journals, Application and Software Download, Bandwidth, Blog Commenting, Blog Posting, Blogs and Personal Sites, Bot Networks, Business and Economy, Classifieds Posting, Collaboration Office, Compromised Websites Computer Security, Content Delivery Networks, Cultural Institutions, Education, Educational Institutions, Educational Materials, Educational Video, Emerging Exploits, Entertainment, File Download Servers, Files Containing Passwords, Financial Data and Services, Freeware and Software Download, Games, Gay or Lesbian or Bisexual Interest, Government, Hacking, Hobbies, Hosted Business Applications, Image Servers, Images Media, Information Technology, Internet Auctions, Internet Radio and TV, Internet Telephony, Intolerance, Job Search, Keyloggers, Lingerie and Swimsuit, Malicious Embedded iFrame, Malicious Embedded Link, Malicious Web Sites, Media File Download, Militancy and Extremist, Military, Mobile Malware, Network Errors, News and Media, Non Traditional Religions, Non Traditional Religions and Occult and Folklore, Nudity, Office Apps, Office Documents, Office Drive, Office Mail, Online Brokerage and Trading, Parked Domain, Peer to Peer File Sharing, Personal Network Storage and Backup, Personals and Dating, Phishing and Other Frauds, Political Organizations, Potentially Exploited Documents, Potentially Unwanted Software, Private IP Addresses, Pro Choice, Pro Life, Professional and Worker Organizations, Proxy Avoidance, Real Estate, Reference Materials, Religion, Restaurants and Dining, Search Engines and Portals, Security, Service and Philanthropic Organizations, Sex, Sex Education, Shopping, Social and Affiliation Organizations, Social Organizations, Social Web Facebook Social Web Linkedin, Social Web Twitter, Social Web Youtube, Society and Lifestyles, Special Events, Sport Hunting and Gun Clubs, Sports, Spyware, Streaming Media, Surveillance, Suspicious Content, Suspicious Embedded Link, Tasteless, Traditional Religions, Unauthorized Mobile Marketplaces, Violence, Viral Video, Web Analytics, Web and Email Marketing, Web and Email Spam, Web Hosting, Web Images, Web Infrastructure, Website Translation

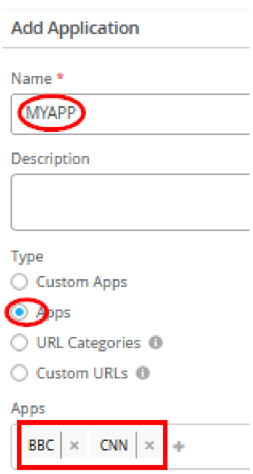
- Using **Custom URLs** as application identifier
  - With custom URLs, you can identify services and applications that are not pre-defined apps or URL categories in some way. Please see the figure below for an example:



Example Usage of a Predefined Application

Go to **Organization -> Applications** and create a new application with the following configuration:

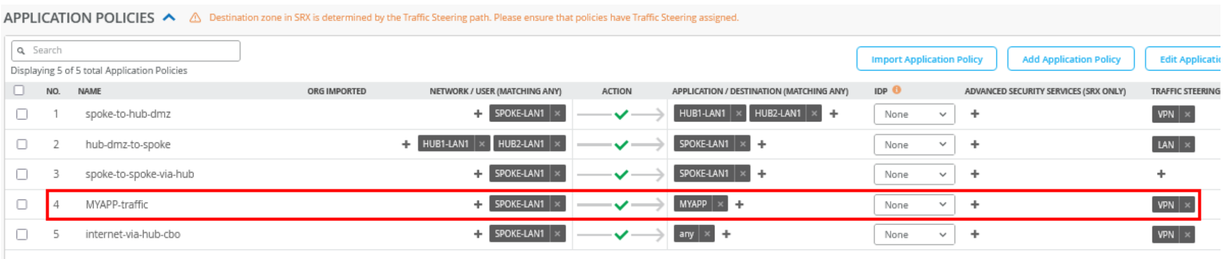
- Name=`MYAPP`
- Type=`Apps`
- Apps=`BBC + CNN`



Then, go to your existing WAN Edge template and insert the following application policy:

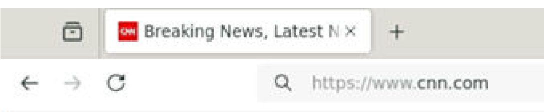
- Number=`4`
  - Name=`MYAPP-traffic`
  - Network=`SPOKE-LAN1`
  - Action=`Pass`

- Application=MYAPP
- Traffic Steering=VPN. If you have already configured LBO from the previous lab "Local Traffic Breakout at the Spoke" on page 202 you can use that as well.

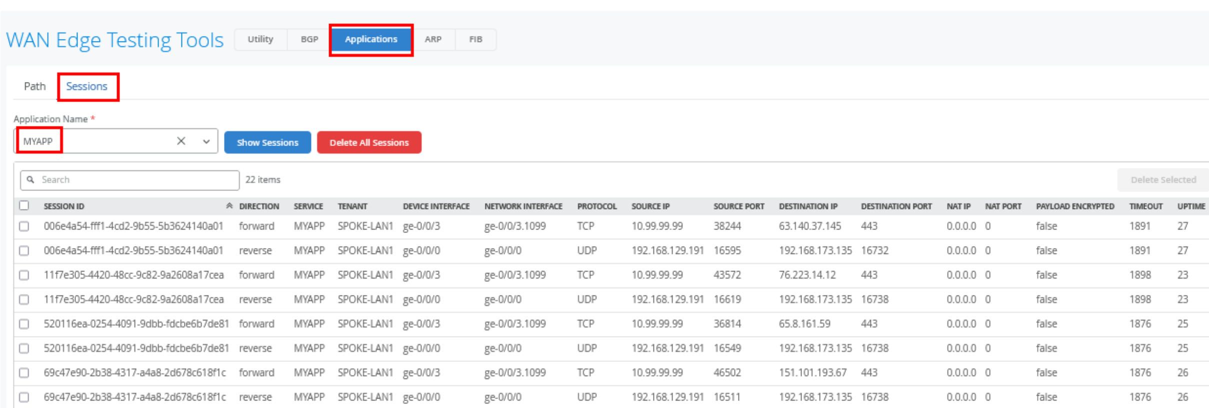


The changes above do not alter the actual traffic flow—it will still use central breakout at the hub due to the broader rule defined below. This step is simply to verify that our rule correctly identifies the traffic. Ensure your changes are saved and successfully applied to the spoke.

Now we need to generate traffic for this custom application. In our example, we use desktop1 VM attached to spoke1 by pointing a browser towards https://www.cnn.com.



After you generate the traffic, use the **Testing Tools** on your spoke by going to **Applications -> Session** with Application Name=MYAPP as shown in the figure below. You see that this traffic was identified and handled correctly.



IDP-Based Threat Detection

An Intrusion Detection and Prevention (IDP) policy lets you selectively enforce various attack detection and prevention techniques on network traffic. You can enable IDP on the Session Smart Router operating as a spoke device in your Juniper Mist network by activating it in an application policy.

Intrusion detection is the process of monitoring the events occurring on your network and analyzing them for signs of incidents, violations, or imminent threats to your security policies. Intrusion prevention is the process of performing intrusion detection and then stopping the detected incidents. For details, see Intrusion Detection and Prevention Overview.

NOTE:

1. You can configure IDP on Session Smart Routers only when the devices are operating as spoke devices.

2. Consider a maintenance window when activating IDP for the first time. The start of the IDP engine and inclusion into the path from LAN to WAN (that is, service-chaining) might take a few minutes and might also interrupt ongoing communications.

3. When using traffic steering for local breakout, a matching rule on the hub is not required. However, with Session Smart Routers, if IDP is to inspect traffic crossing the VPN overlay to a hub, you must also configure a matching rule with IDP enabled on the hub. This is necessary because the internal service name changes when IDP is applied, and a corresponding service name must exist on the remote hub—even if the hub itself is not performing IDP inspection.

Juniper Mist cloud supports the following IDP profiles:

- Standard—The standard profile is the default profile and represents the set of IDP signatures and rules that we recommend. Each attack type and severity have a Juniper-defined, non-configurable action that the IDP engine enforces when it detects an attack. The possible actions are as follows:
  - Close the client and server TCP connection.
  - Drop the current packet and all subsequent packets
  - Send an alert only (no additional action).
- Alert—Alert profiles are suitable only for low-severity attacks. When the IDP engine detects malicious traffic on the network, the system generates an alert, but it does not take additional measures to prevent the attack. The IDP signature and rules are the same as in the standard profile.
- Strict—The strict profile contains a similar set of IDP signatures and rules as the standard profile. However, when the system detects an attack, this profile actively blocks any malicious traffic or other attacks detected on the network.

You can apply an IDP profile to an application policy. Each profile has an associated traffic action, and these actions define how to apply a rule set to a service or an application policy. Actions in the IDP profile are preconfigured and are not available for users to configure.

Example IDP Test Case

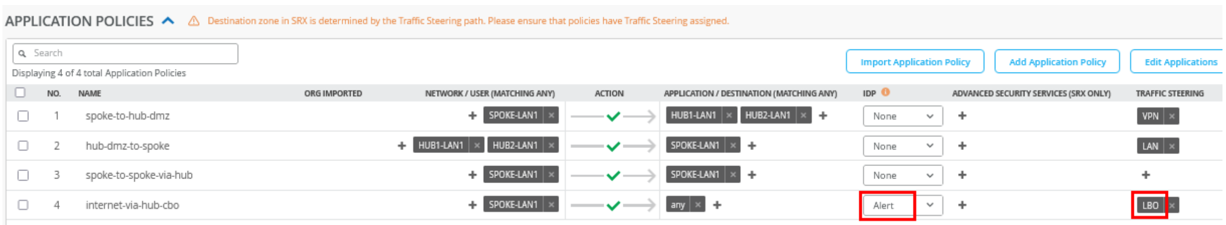
In this test case, we modify the first lab "Appendix: Building a base SD-WAN Topology with Three Spokes and Two Hubs" on page 23 to do the following:

- Local breakout for all non-VPN traffic instead of central breakout.
- Activate IDP alerting for this traffic.
- Install a security scanner on a client attached to a spoke.
- Run the security scanner to inspect a webserver in our lab (not part of the VPN).
- Review the captured IDP events in the Juniper Mist portal.

To configure our example with IDP-based threat detection:

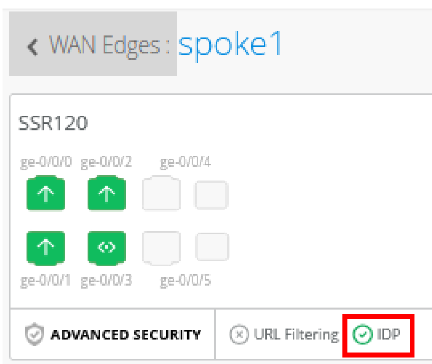
- In the Juniper Mist cloud portal, click **Organization > WAN Edge** Templates and select a template for your spoke device.
- Then, configure an additional traffic steering policy in the spoke WAN Edge template:
  - Name=`LBO`
  - Strategy=`Weighted`
  - Paths
    - Type1=`WAN: INET`
    - Cost1=`10`
    - Type2=`WAN: MPLS`
    - Cost2=`20`
- Modify the existing **Applications Policies**
  - Number=`4`
    - Name= `internet-via-hub-cbo`
    - Network=`SPOKE-LAN1`
    - Action=`Pass`
    - Application=`any`

- IDP=`Alert`
- Traffic Steering=`LBO`



Save your template to get this IDP configuration committed on the spokes.

Inspect the WAN Edge spoke after a few minutes. Under **Advanced Security**, the IDP service should now be activated.

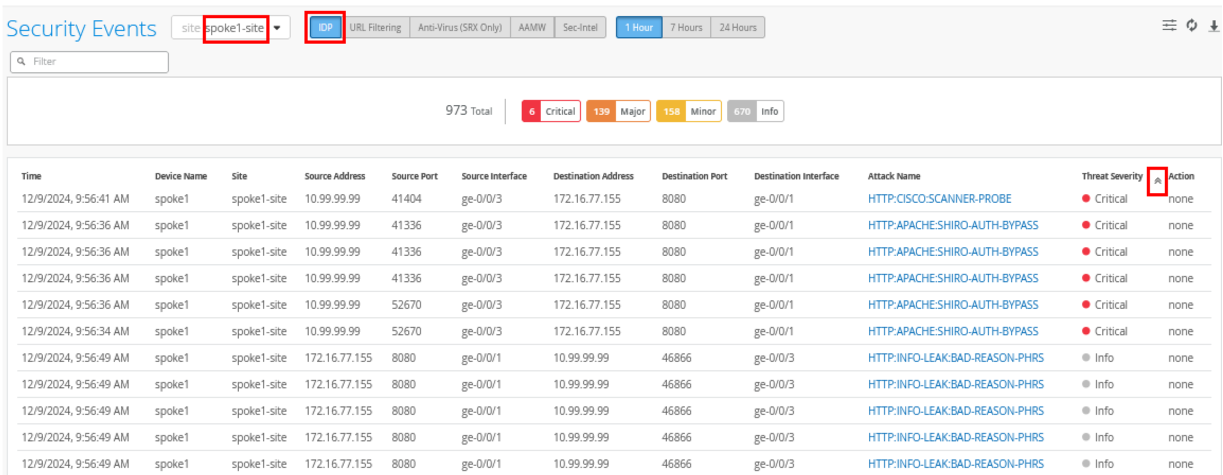


We shall now continue our testing on the clients attached to the spokes. We connect to desktop1 VM with IP address `10.99.99.99` attached to Spoke1. There, we install a security scanner service called nikto and let it inspect a local webserver that our lab happens to have.

```
root@desktop1:~# apt-get install -y nikto
.
root@desktop1:~# nikto -h http://172.16.77.155:8080
- Nikto v2.1.5
---------------------------------------------------------------------------
+ Target IP:          172.16.77.155
+ Target Hostname:    172.16.77.155
+ Target Port:        8080
+ Start Time:         2024-12-09 08:51:20 (GMT0)
---------------------------------------------------------------------------
+ Server: No banner retrieved
+ Server leaks inodes via ETags, header found with file /, fields: 0xW/1895 0x1733226983740
+ The anti-clickjacking X-Frame-Options header is not present.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
```

```
+ /: Appears to be a default Apache Tomcat install.
+ 6544 items checked: 0 error(s) and 3 item(s) reported on remote host
+ End Time:           2024-12-09 08:51:37 (GMT0) (17 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Then, we go to **Site -> Security Events** and check **IDP** to review the captured IDP events as seen in the figure below.



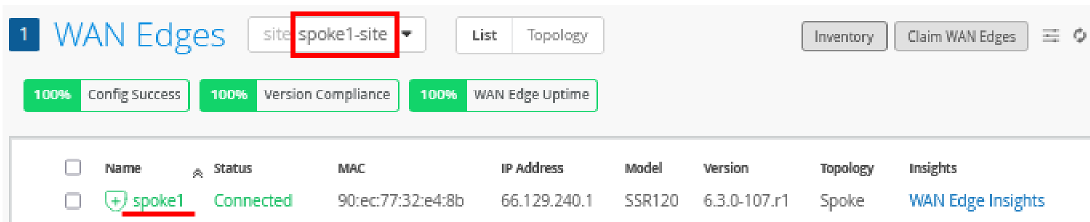
Appendix: Device, Application and WAN Monitoring

IN THIS SECTION

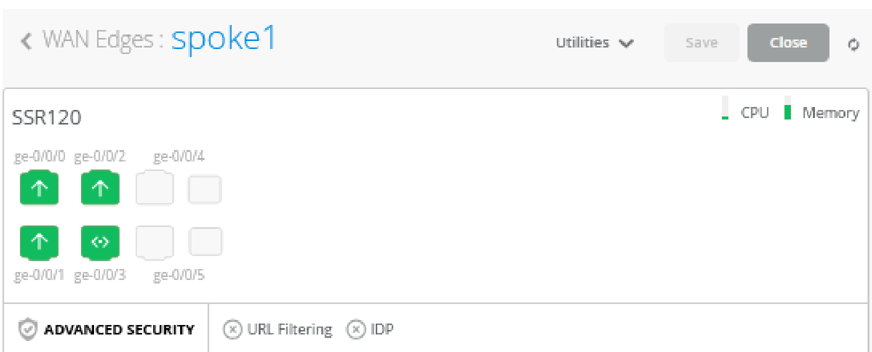
This section covers most of the Day 2 aspects of your SD-WAN installation.

Device Information Page

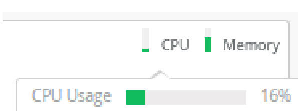
To get to the basic device monitoring page, click **WAN Edges**, select a site, and then click on the device as shown below:



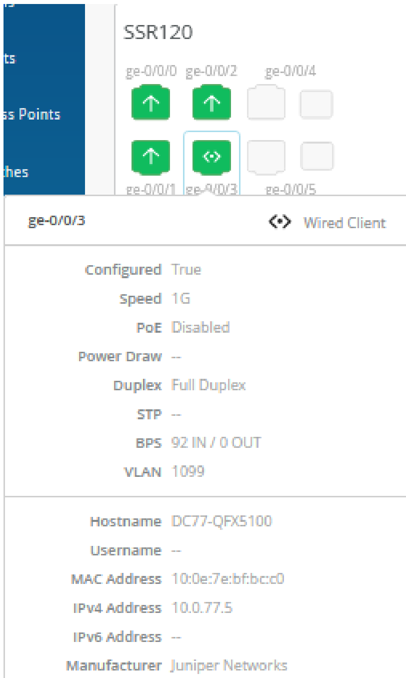
At the top of the device information page, you see a graphical front view of the device, its ports, and some baseline status information.



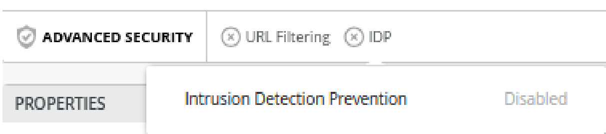
Hover the mouse over each status icon for CPU and memory to see how the device is behaving.



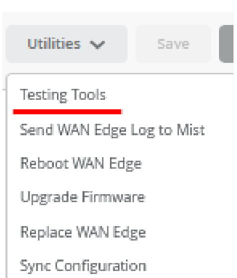
Next, hover the mouse over some of the device ports to review what is configured and detected. In this example, you see at the bottom that the lab switch is detected as a client attached to the port:



Below the front ports, hover the mouse over each security service and review the displayed information.

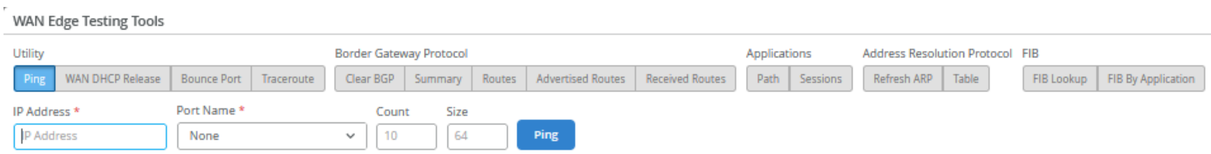


Check out the **Utilities** menu. Then, click **Testing Tools** for more options.

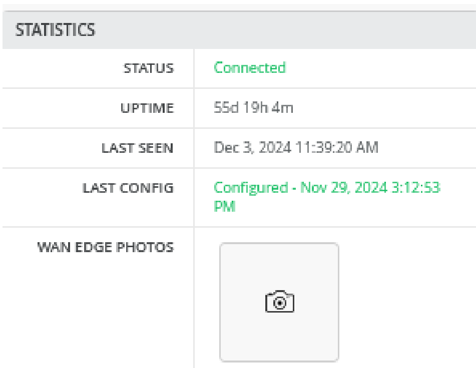


The **Testing Tools** enable:

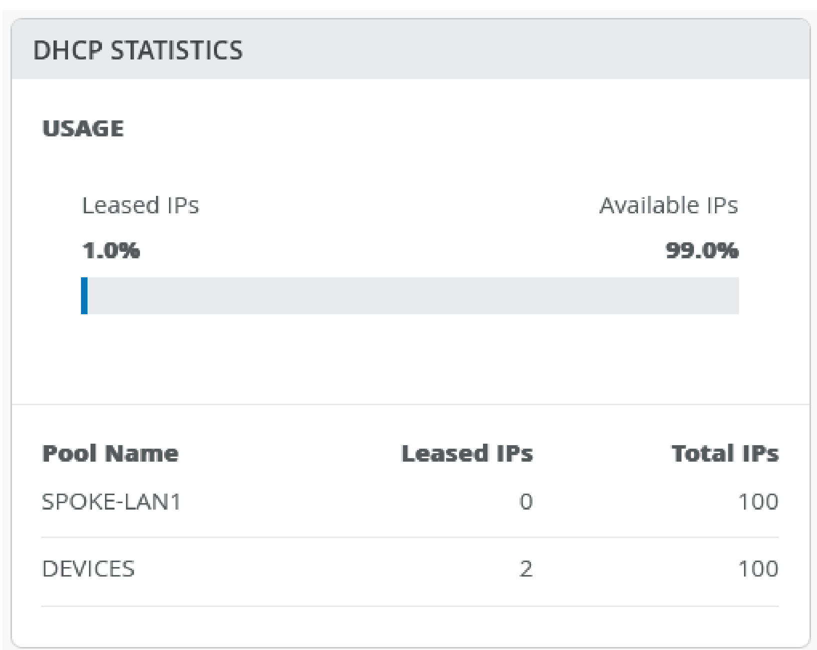
- Simple commands such as ping and traceroute.
- Review the BGP protocol that distributes the routes of the VPN overlay.
- Review Application Path and Session information.
- Review the ARP status.
- Review the FIB for your application traffic.



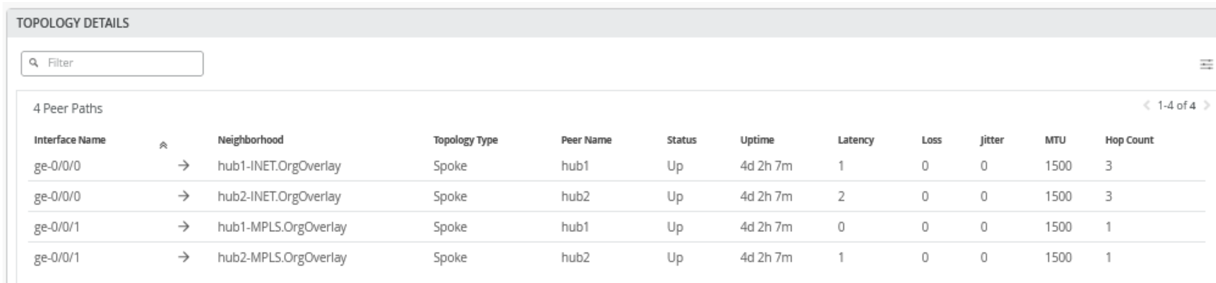
Back to the Device information page, review the **Statistics** pane for information.



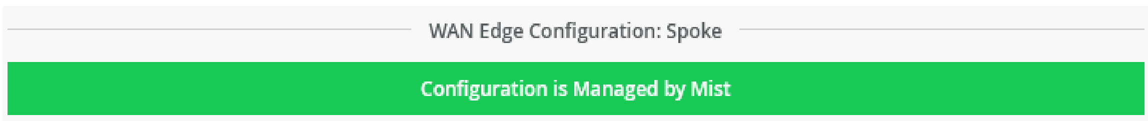
In case you have configured DHCP servers on the WAN router, the **DHCP Statistics** pane displays the very useful information about the leases handed out.



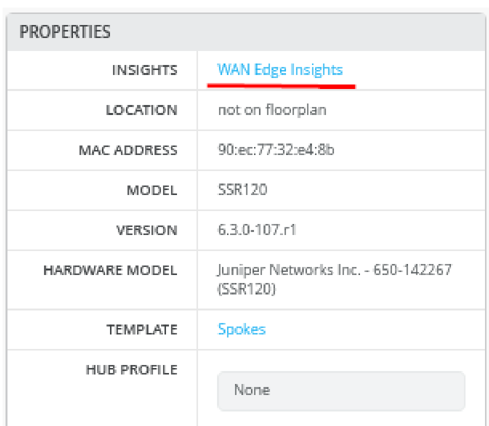
The status of the Secure Vector Routing VPN Overlay Tunnels can be seen in the **Topology Details** pane:



Then, review the device configuration. Usually, it should be inherited by the templates or profiles you have used. You can make individual changes to the configuration to be pushed to the device.

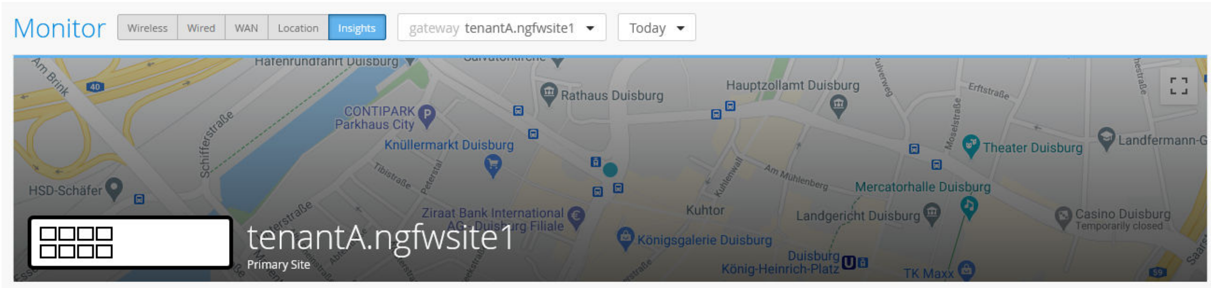


Finally, review the **Properties** pane for information and then click **WAN Edge Insights** for the next level of information about the device.

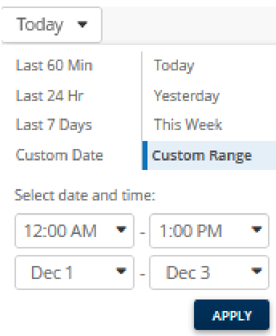


WAN Edge Insights Page

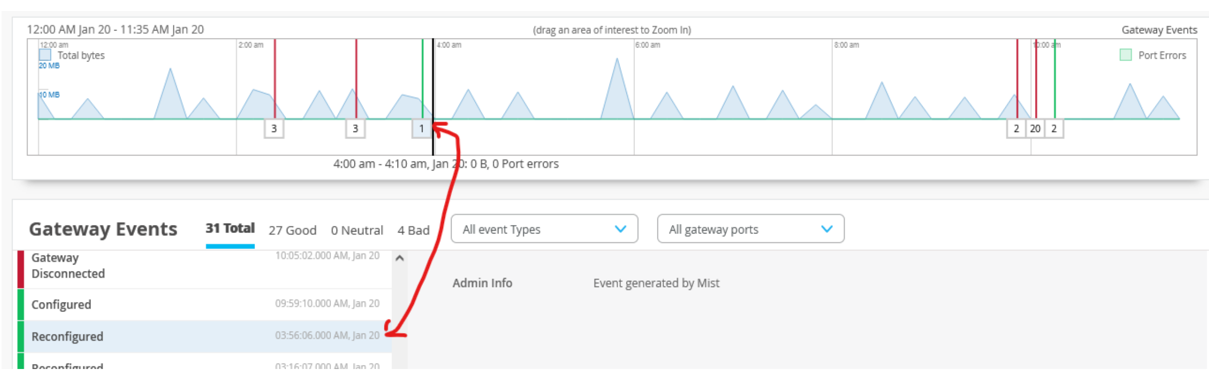
At the top of the WAN Edge Insights page, you see the site's location-based information showing where this gateway is on a map.



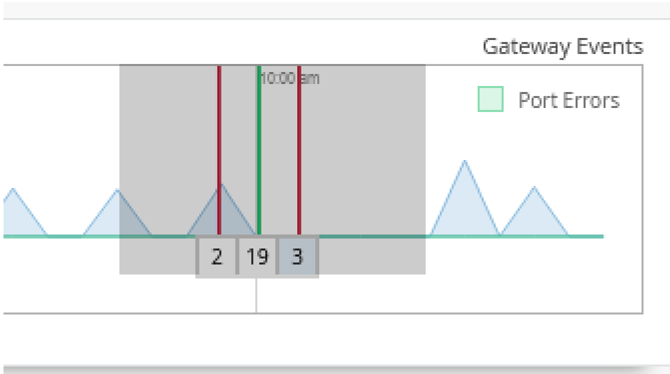
At the top of the page, you can also select the time period for the data you want to view. By default, the time period is set to **Today**.



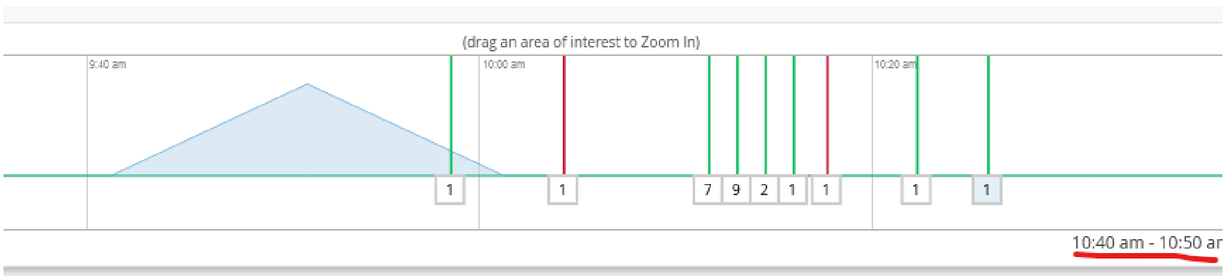
Below the street map, you see the timeline for gateway events (and information about the traffic passing through the device at that time). With your mouse, you can select an event to check, which is selected in the events reports as shown below:



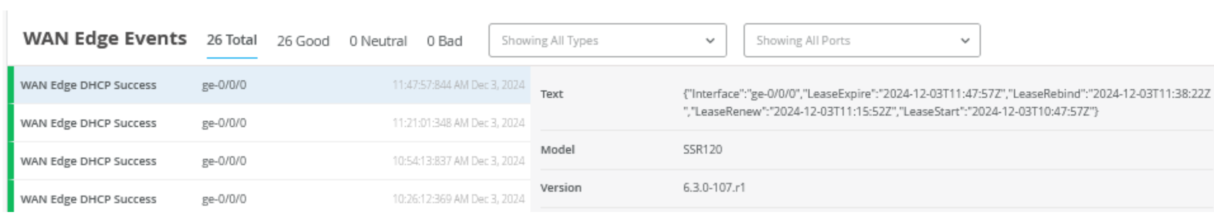
You can also zoom in by selecting an area in the timeline with your mouse cursor. Ensure the selected time period is not too short.



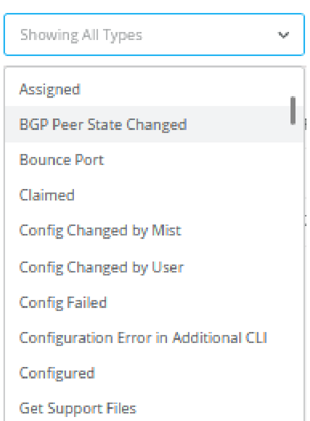
Then, get a more detailed view for the previous time period:



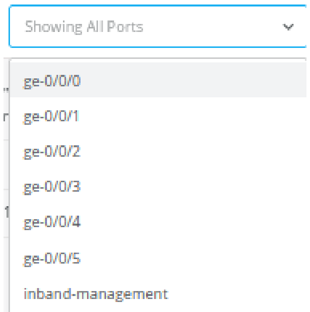
Then, review the Gateway Events pane:



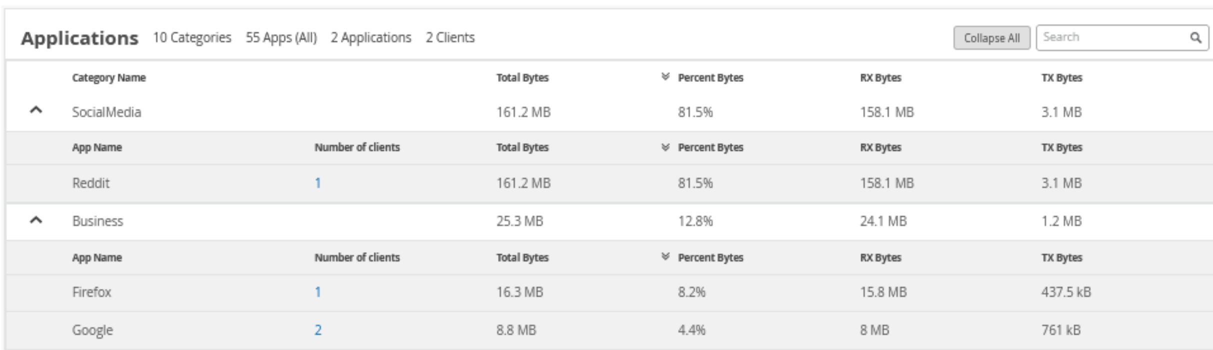
You can filter the events displayed as shown below:



You can filter the events displayed for specific ports as shown below:

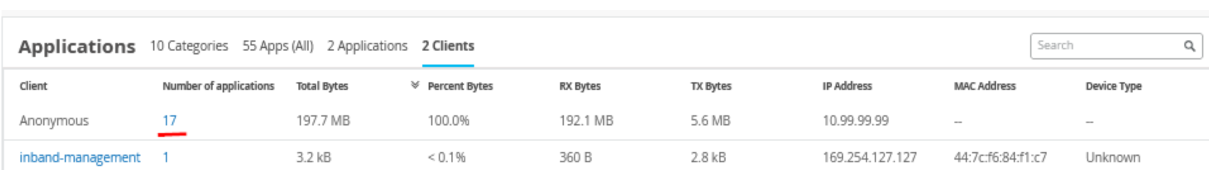


If your device is properly configured and has been sending telemetry data to the Juniper Mist cloud for at least an hour after initial adoption, you should begin to see reports in the **Applications** pane.



	Category Name		Total Bytes	Percent Bytes	RX Bytes	TX Bytes
^	SocialMedia		161.2 MB	81.5%	158.1 MB	3.1 MB
	App Name	Number of clients	Total Bytes	Percent Bytes	RX Bytes	TX Bytes
	Reddit	1	161.2 MB	81.5%	158.1 MB	3.1 MB
^	Business		25.3 MB	12.8%	24.1 MB	1.2 MB
	App Name	Number of clients	Total Bytes	Percent Bytes	RX Bytes	TX Bytes
	Firefox	1	16.3 MB	8.2%	15.8 MB	437.5 kB
	Google	2	8.8 MB	4.4%	8 MB	761 kB

Through the **Clients** tab, you can see bandwidth usage by client.



Client	Number of applications	Total Bytes	Percent Bytes	RX Bytes	TX Bytes	IP Address	MAC Address	Device Type
Anonymous	17	197.7 MB	100.0%	192.1 MB	5.6 MB	10.99.99.99	--	--
inband-management	1	3.2 kB	< 0.1%	360 B	2.8 kB	169.254.127.127	44:7c:f6:84:f1:c7	Unknown

Click on the client to further drill down to see which applications are used.

Applications For Client ✕

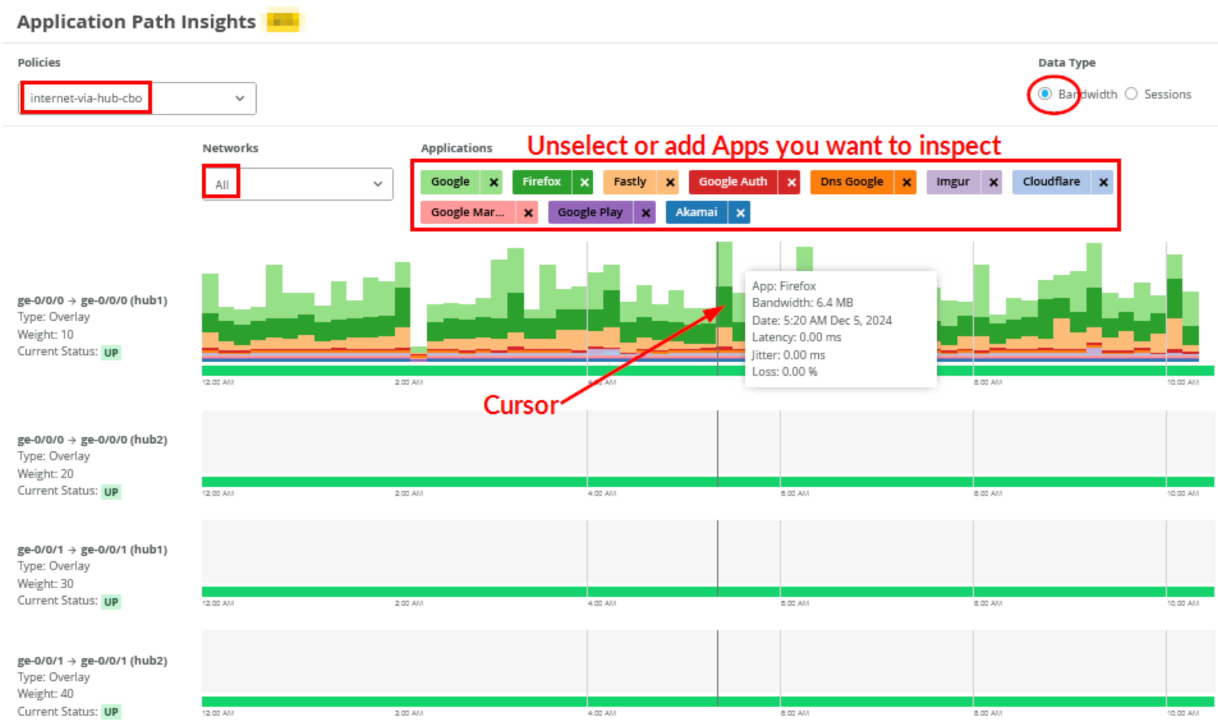
17 Applications associated with **10.99.99.99**

1-17 of **17**

App name	Total Bytes	Percent Bytes	RX Bytes	TX Bytes
Reddit	161.2 MB	81.5%	158.1 MB	3.1 MB
Firefox	16.3 MB	8.2%	15.8 MB	437.5 kB
Google	8.8 MB	4.4%	8 MB	758.3 kB
Fastly	4 MB	2.0%	3.5 MB	482.7 kB
Youtube	3.8 MB	1.9%	3.6 MB	142.2 kB
Wikimedia Foundation	1.3 MB	0.7%	1.2 MB	115.8 kB
Wikipedia	769.2 kB	0.4%	734.6 kB	34.6 kB
DNS Google	571 kB	0.3%	279.1 kB	291.8 kB
Xfinity	427 kB	0.2%	400 kB	27 kB
Google Auth	283.2 kB	0.1%	260.1 kB	23.1 kB
Unclassified	224.1 kB	0.1%	114.2 kB	109.9 kB
Cloudflare	52.3 kB	< 0.1%	29.1 kB	23.2 kB
Google Marketing	23.2 kB	< 0.1%	13.7 kB	9.5 kB
Google Play	20.9 kB	< 0.1%	9.1 kB	11.8 kB
Ubuntu	10 kB	< 0.1%	0	10 kB
Akamai	8 kB	< 0.1%	2.9 kB	5.1 kB
Risky Advertiser	3.5 kB	< 0.1%	840 B	2.7 kB

Next, the new **Application Policies** pane presents bandwidth usage details for each application across the individual paths within your SD-WAN infrastructure:

- **Policy** enables you to set a filter on the configured application policies.
- **Network** enables to review all LAN networks or only one.
- **Applications** enables you to deselect or add applications you are interested.
- **Data Type** enables you to review the application bandwidth, or the amount of session opened.
- **Bubble** enables you to view more details. You must move the cursor over the application in a path to get a bubble.



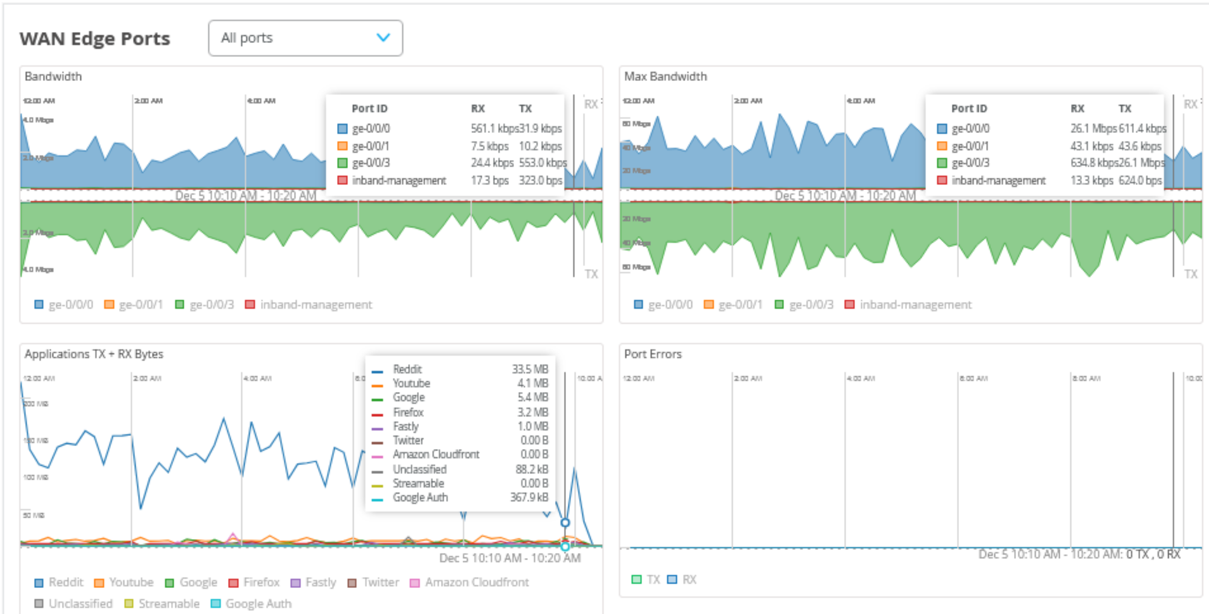
Below is the same view again. But we've chosen to view the session counts:

Next is the **WAN Edge Device** pane with the following charts:

- Control Plane CPU
- Data Plane CPU
- Memory Utilization

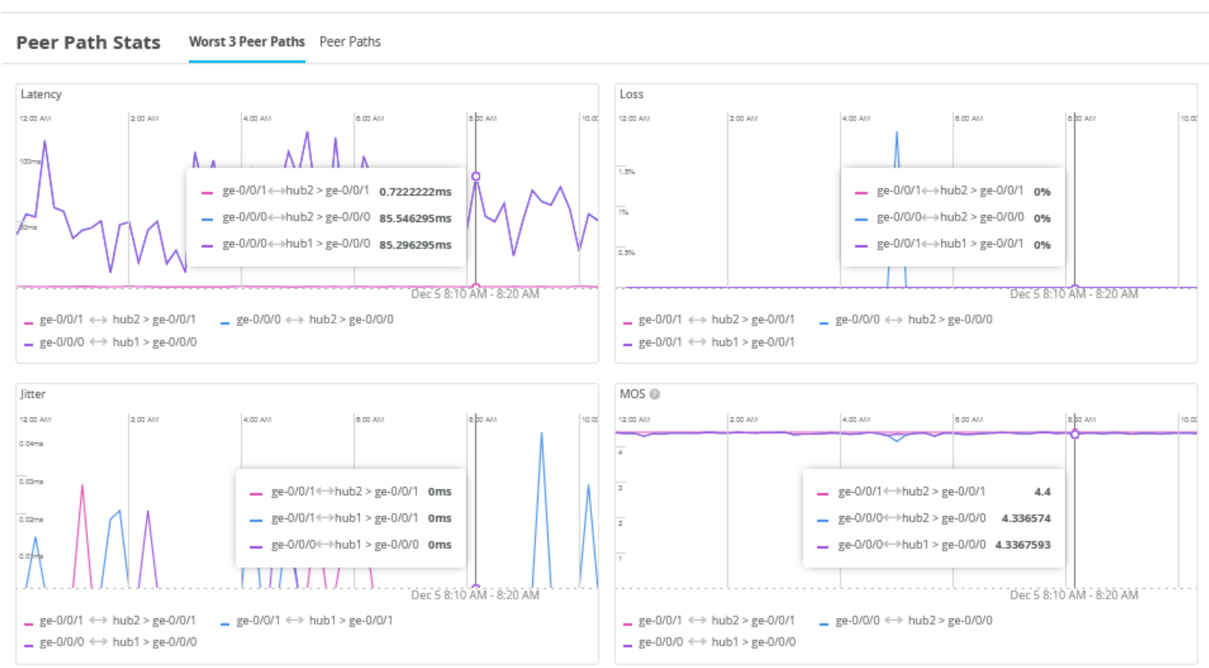
Next is the **WAN Edge Ports** pane with the following charts:

- Bandwidth
- Max Bandwidth
- Applications
- Port Errors

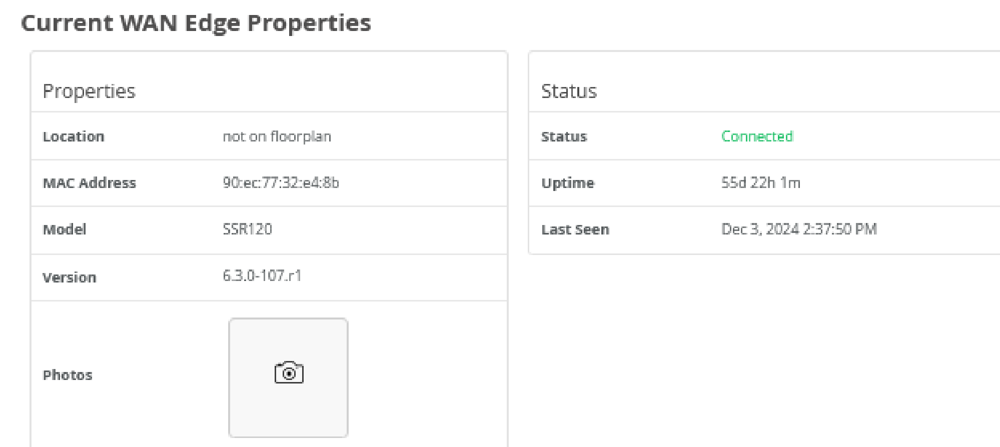


Next is the **Peer Path Stats** pane with the following charts:

- Latency
- Loss
- Jitter
- MOS (Mean Opinion Score)

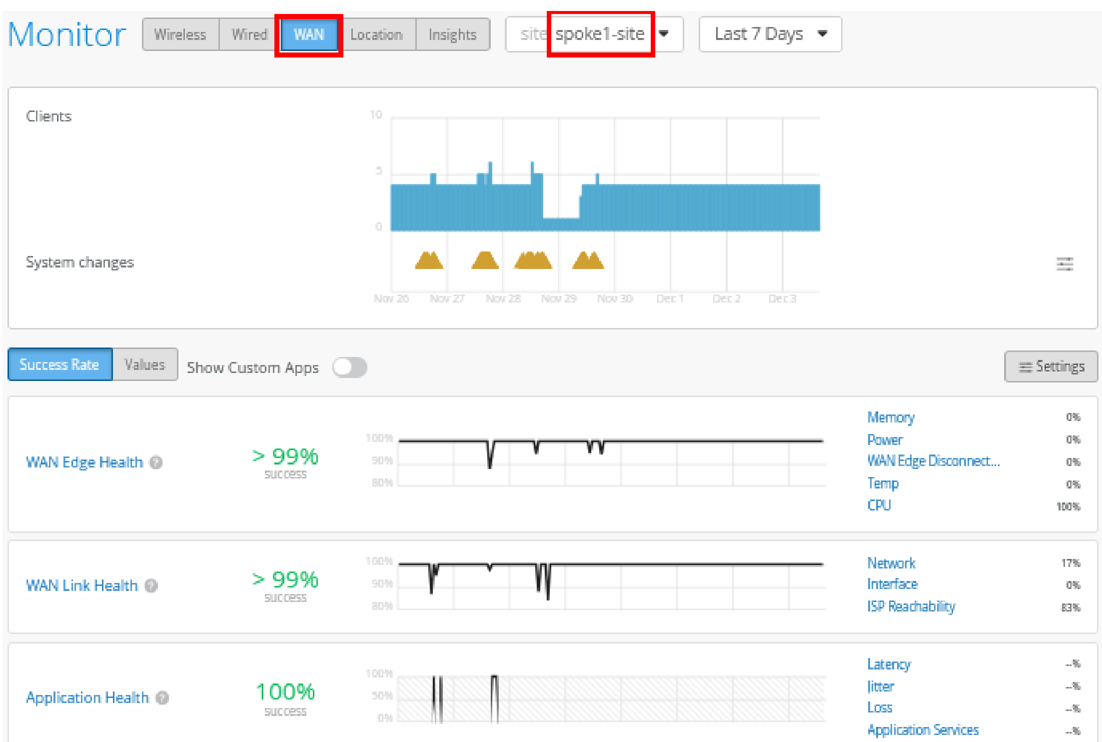


Then, the last pane on this page is **Current WAN Edge Properties**.

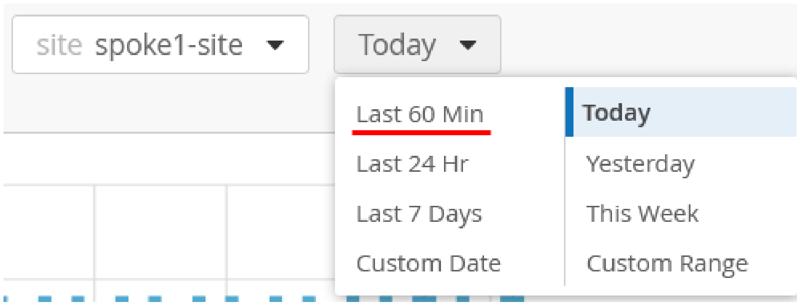


WAN SLE Monitor Page

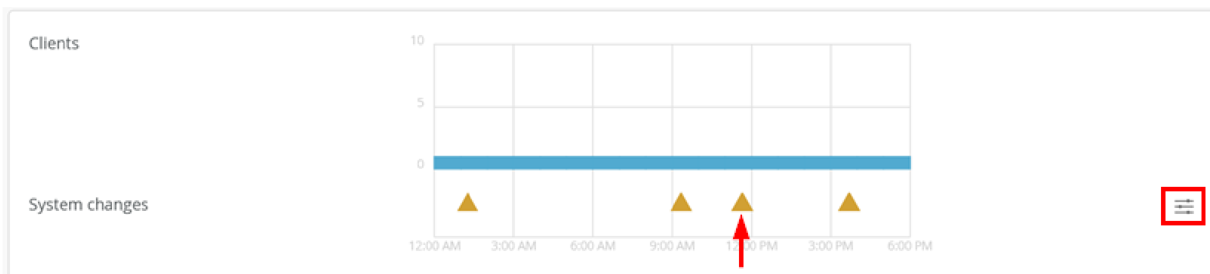
The next level of information is regarding WAN SLE monitoring. To review the information, click **Monitor > Service Levels**. Then, select a site for inspection and select **WAN**.



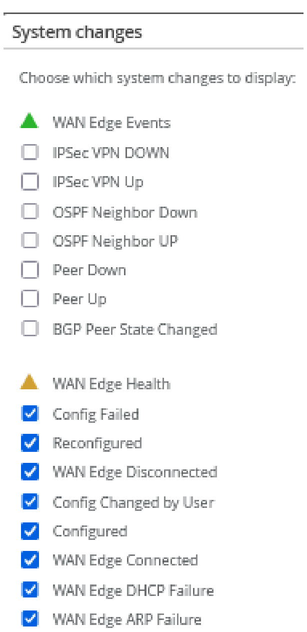
Keep in mind that all WAN SLE metrics are designed for long-term monitoring. They may show limited data immediately after onboarding a device. In a production environment, it's typical to need a week's worth of metrics. You can try adjusting the time range—for example, selecting **Last 60 Min**—but it may still display minimal information at this early stage.



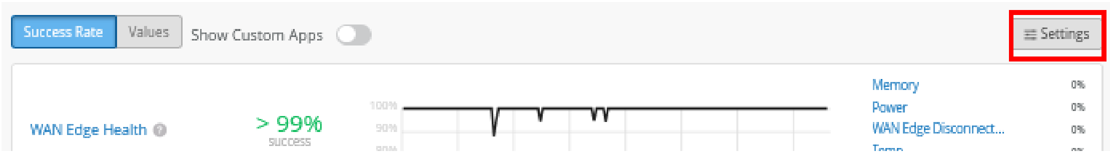
The first pane shows the relationship between the number of connected clients at a given time and any system events that occurred during that period. An amber triangle indicates when a change has taken place. Additionally, take note of the information displayed in the lower-right corner of the pane, which provides further context on reported activity.



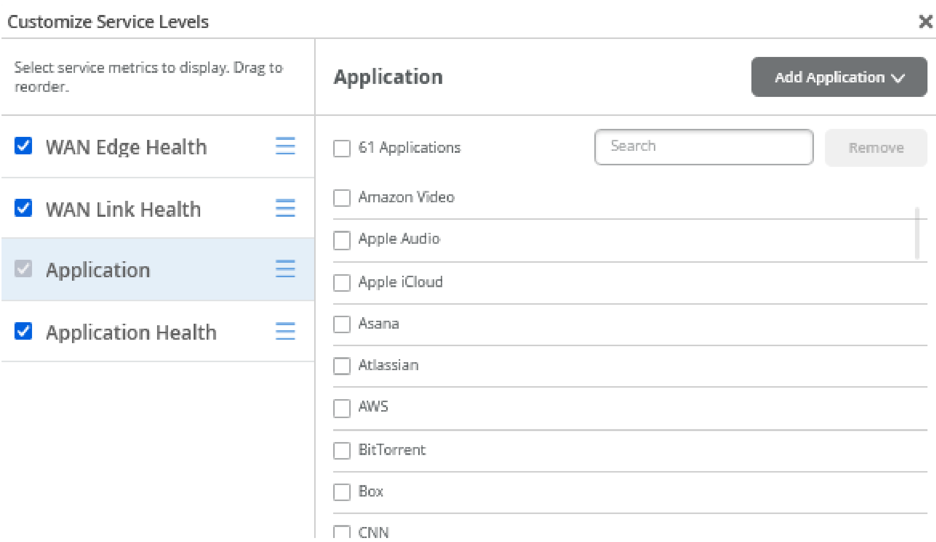
You can select which system changes should be displayed:



Back on the WAN SLE page, make yourself familiar with the **Settings** in the upper-right corner.



You can customize a few settings, but most are adjusted automatically. In contrast to SRX Series Firewalls, the Session Smart Routers do not require application customization and probe configuration. All applications are automatically monitored when they are detected but you can add or remove them in the dialogue window indicated below:

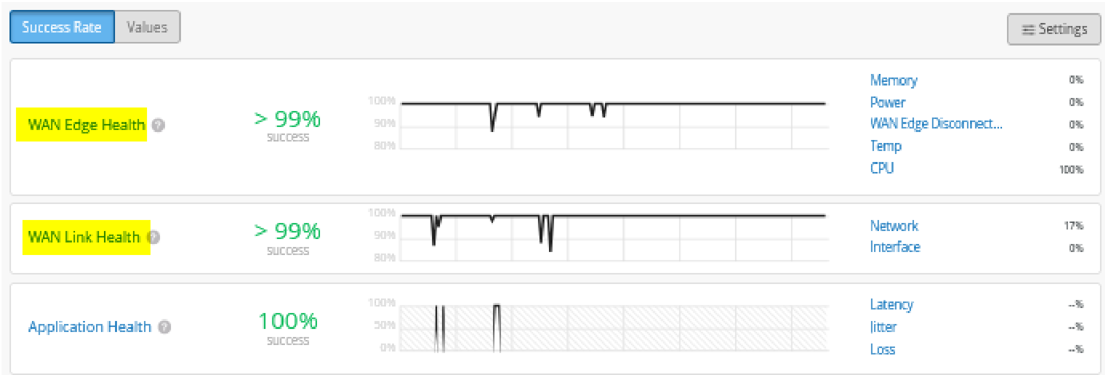


NOTE: It's important to understand that the metrics and reports for WAN Edge Health, WAN Link Health, and Application Health are powered by Mist AI, which uses a TensorFlow-based network. This has several implications:

1. Data-Driven Learning: Like all AI systems, Mist AI requires a significant amount of data to analyze and learn the behavior of your network. For meaningful insights, we recommend waiting at least a week after installing a spoke and generating traffic before reviewing the health metrics.

2. Proactive Health Insights: Unlike traditional monitoring tools that simply display raw data and leave interpretation up to the user, Mist AI evaluates network health and highlights only those areas that are at risk. If no issues are displayed, it indicates that your network is performing well and no immediate review is necessary.

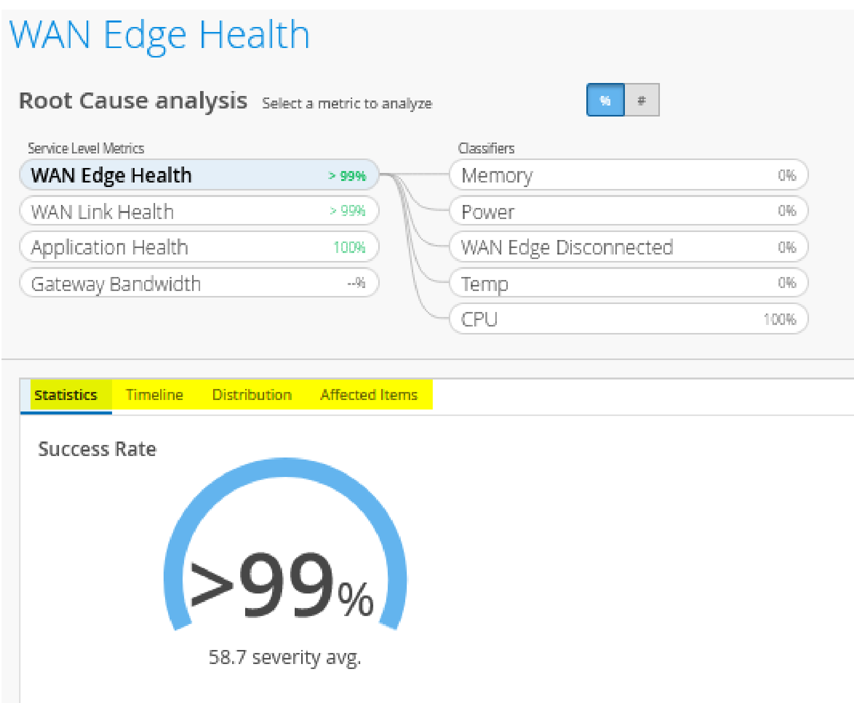
Let's now focus on the reports you can get through WAN Edge Health and WAN Link Health.



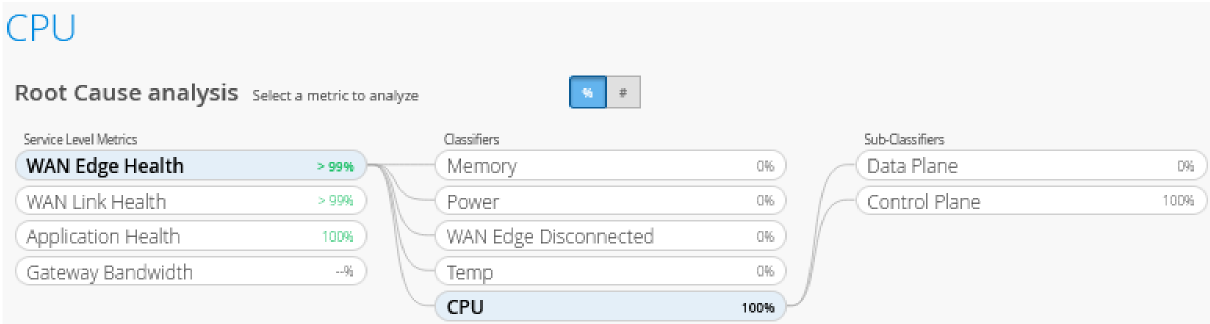
WAN Edge Health reports the health check of the Session Smart Router device deployed with metrics and classifiers such as:

- Memory usage
- Power
- WAN Edge Disconnected
- Temperature
- CPU utilization

Below is an example chart. Use the tabs to explore more detailed, granular information:

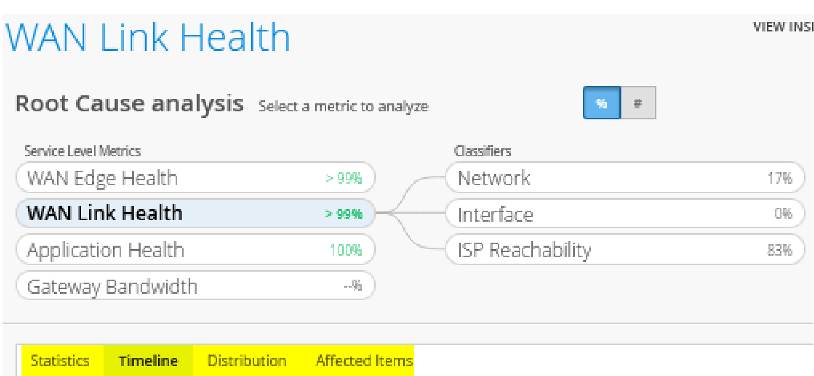


Temperature and CPU utilization have sub-classifiers as shown in the example below:

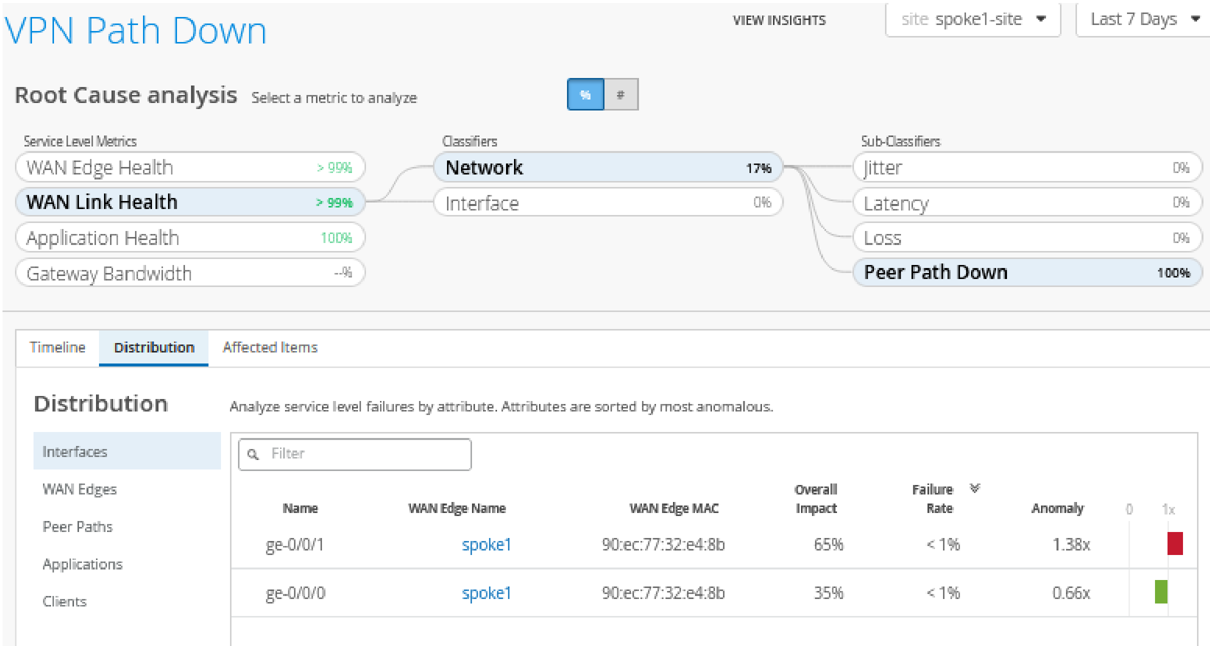


WAN Link Health reports the health status of the Session Smart Router deployed with metrics and classifiers such as:

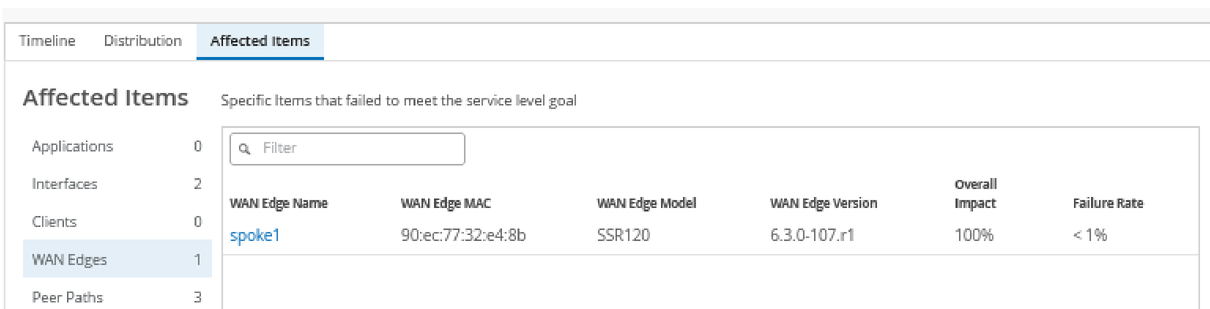
- Network
  - Jitter
  - Latency
  - Loss
  - Peer Path Down
- Interface
  - LTE Signal
  - Congestion
  - Cable Issues
- ISP Reachability
  - ARP
  - DHCP



.



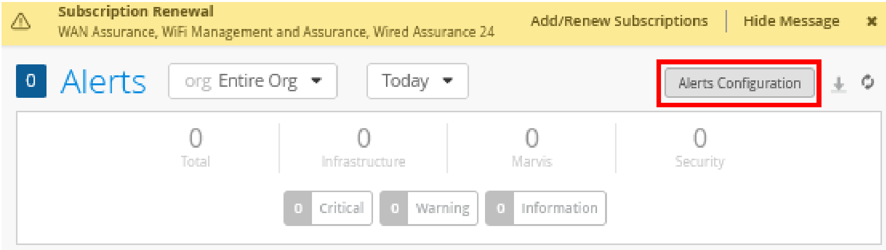
.



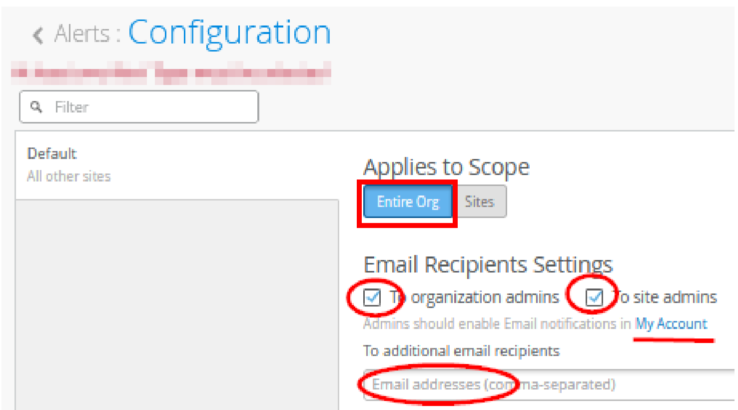
NOTE: Reports on SLEs are only made visible if there is a concern you need to review. If you want charts on raw data without the benefit of an AI based analysis, see the Device page for "WAN Edge Insights Page" on page 240.

Alerts Page

This test case demonstrates how to view gateway alarms and receive them as email notifications for the administrator. To set this up, navigate to **Monitor -> Alerts**, review the current alerts page, and then click on **Alerts Configuration**.

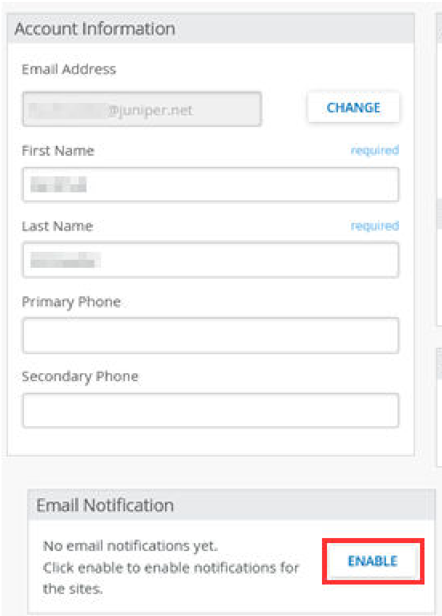


Under **Configuration**, enable the reporting default for Scope=Entire Org, **To Organization admins**, and **To site admins**. You can either add your email address to the **To additional email recipients** field or click **My Account** in the upper-right corner to verify your settings.

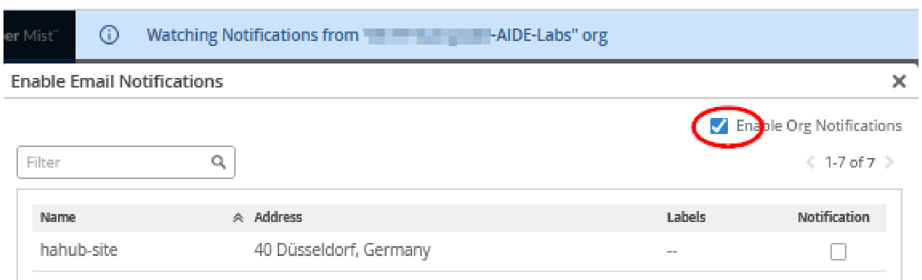


NOTE: By default, administrators do not receive email notifications. To start receiving them, make sure to enable email alerts in your settings.

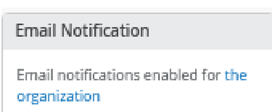
If you have followed the **My Account** link, click **Enable** under **Email Notification**.



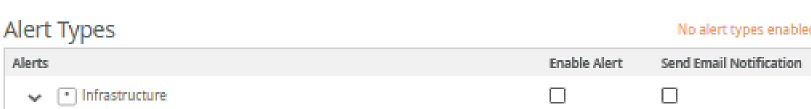
You can enable notifications on a site-by-site basis. But for now, enable the **Enable Org Notifications** option as shown below:

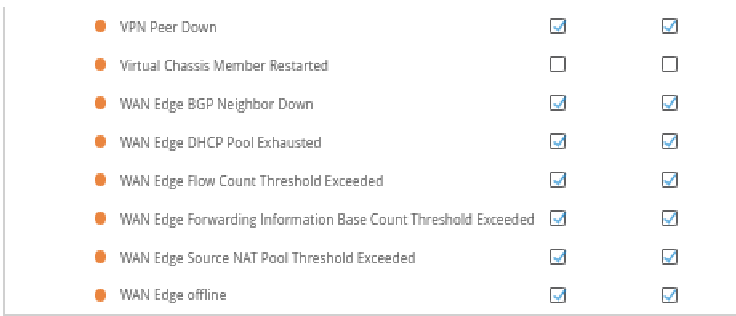


Your account email notification settings look similar to the figure below:



Now, enable the Gateway Alerts and email notifications for Infrastructure as the options shown below:

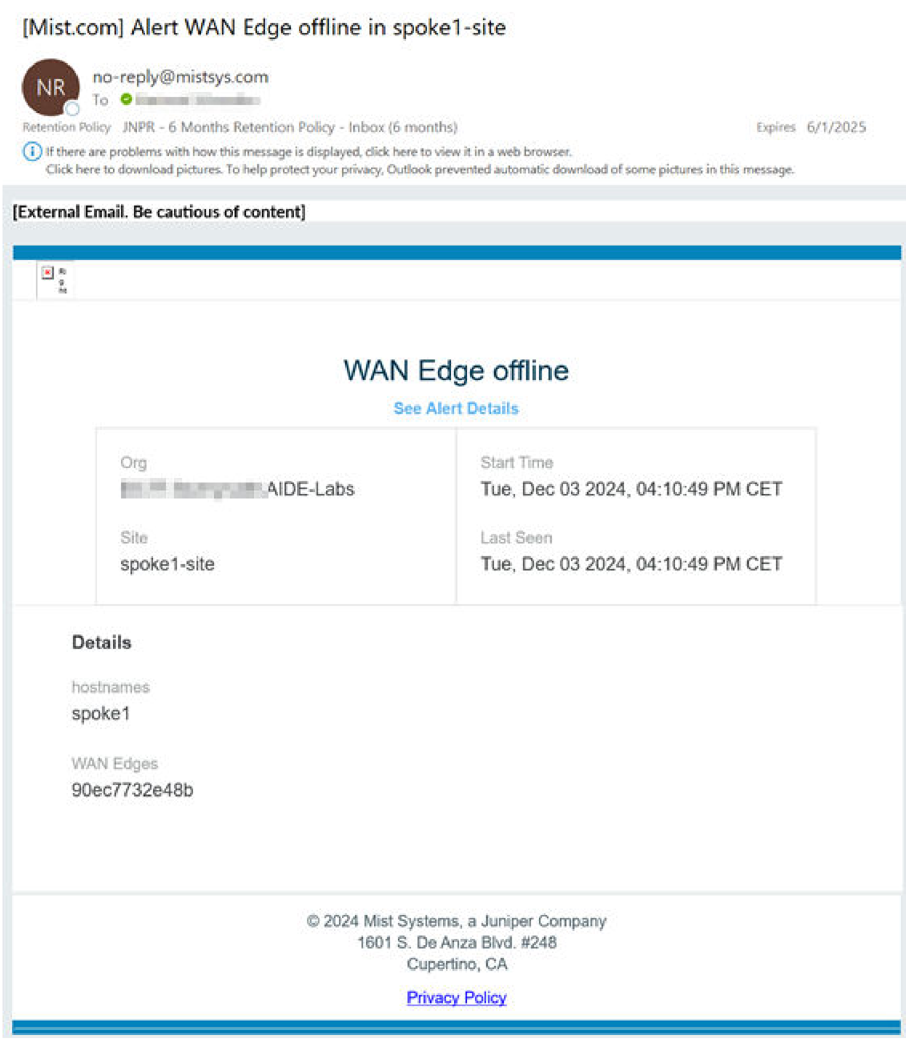




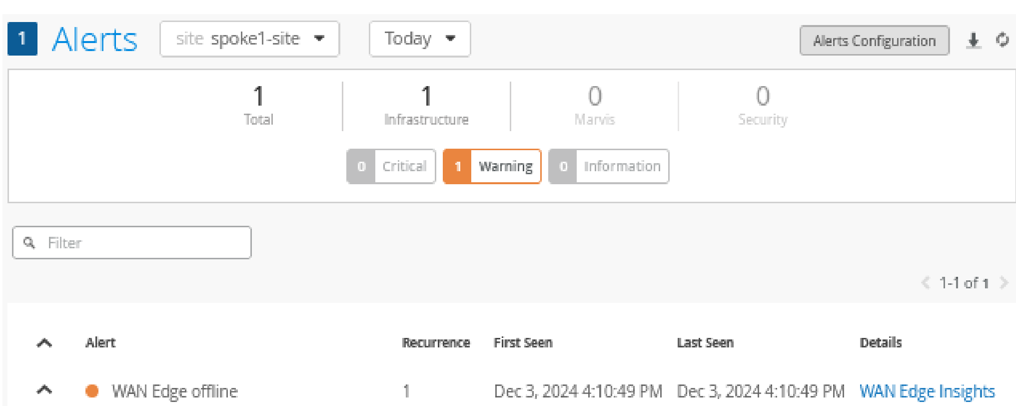
Additionally, we recommend you enable the Marvis WAN Edge alerts and email notifications.

.

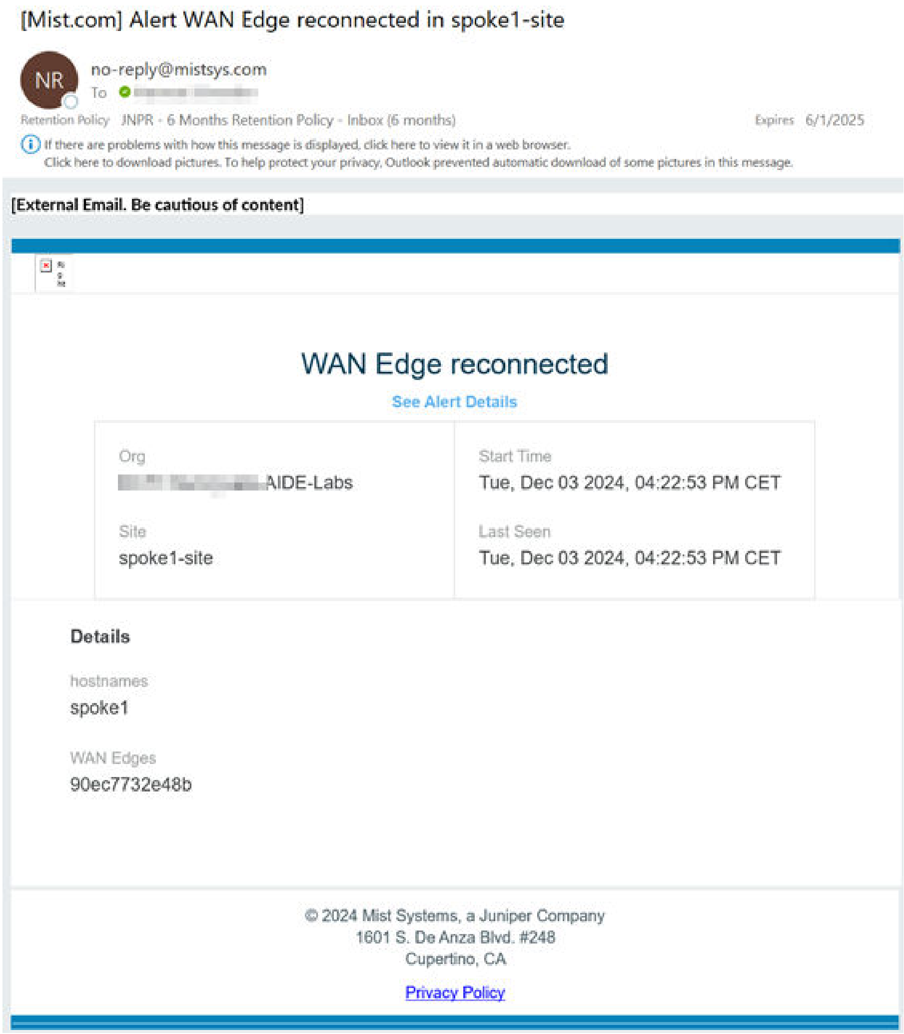
As an example, if a device loses connection to the Juniper Mist cloud, you might receive an email after a couple of minutes. See an example below:



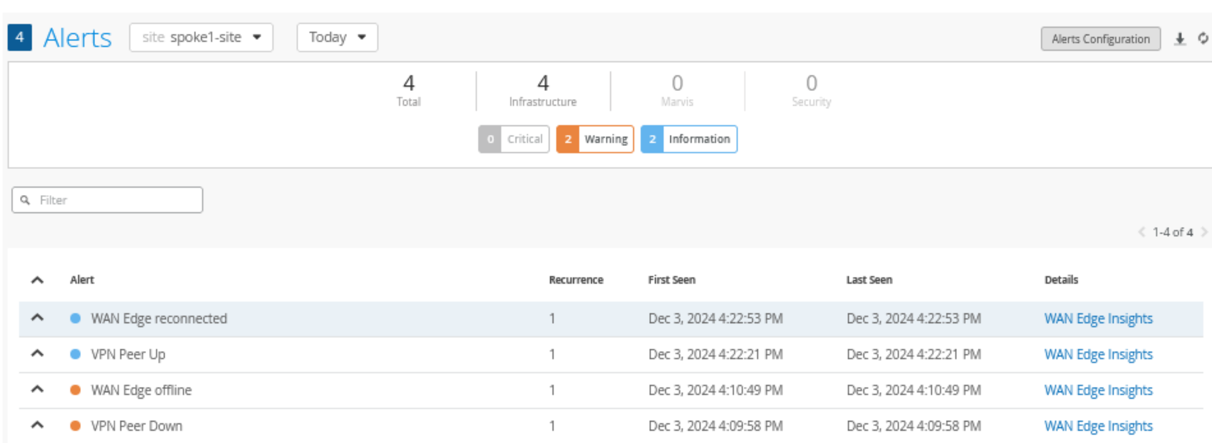
When you click **See Alert Details**, the link redirects you to the **Alerts** page. You can also navigate directly to the **Alerts** page to view the event reported as shown below:



Let's assume that the connection to the Juniper Mist cloud is restored and you get another email with a status change. When such an email arrives, the alert details are similar to those shown below:

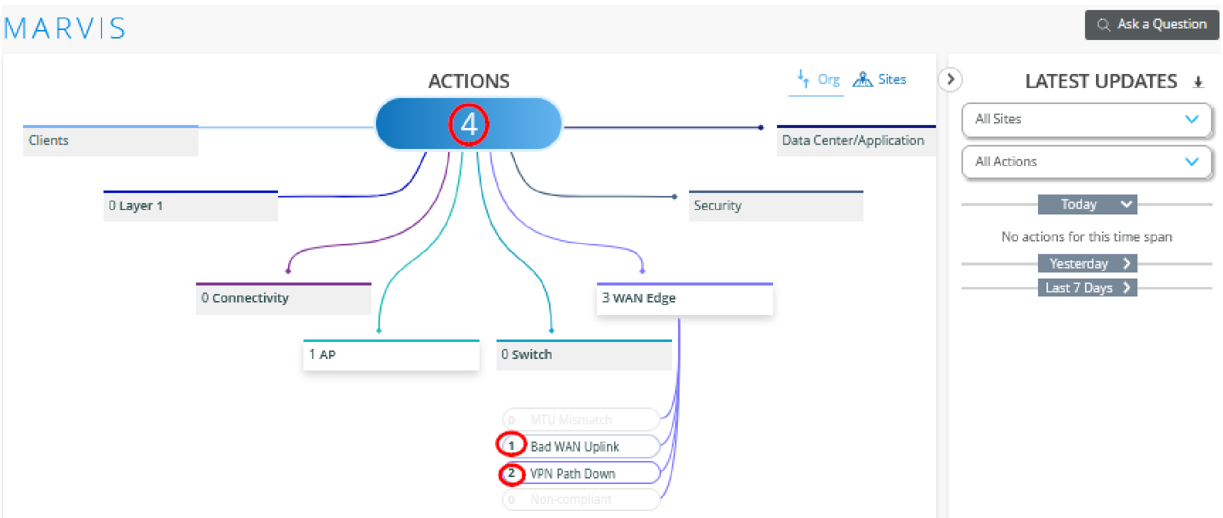


Again, on the **Alerts** page you should see the second event reported.



Marvis Actions

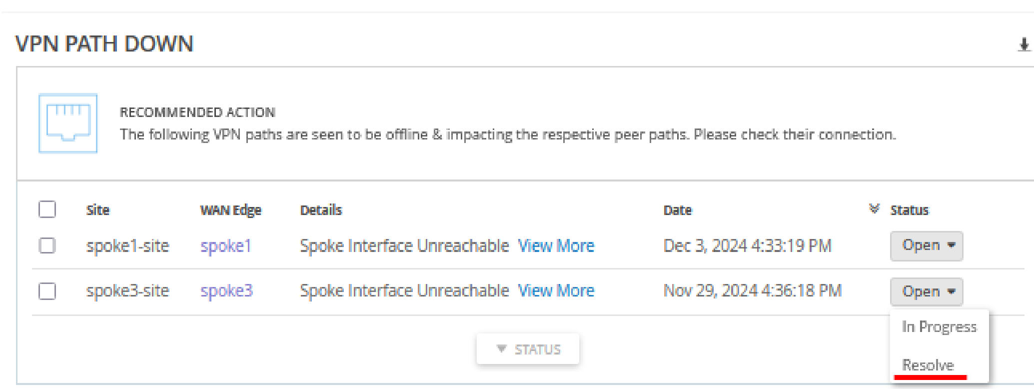
Marvis Actions are reachable through **Marvis** > **Marvis Actions**.



The Marvis Actions related to WAN Edge include:

- MTU Mismatch
- Bad WAN Uplink
- VPN Path Down
- Non compliant

In our example, with a simulated WAN outage, we can inspect the **VPN Path Down** to get more information.



When you select **Resolve** under an alarm's status, you have the option to add details about the resolution for better context and documentation.

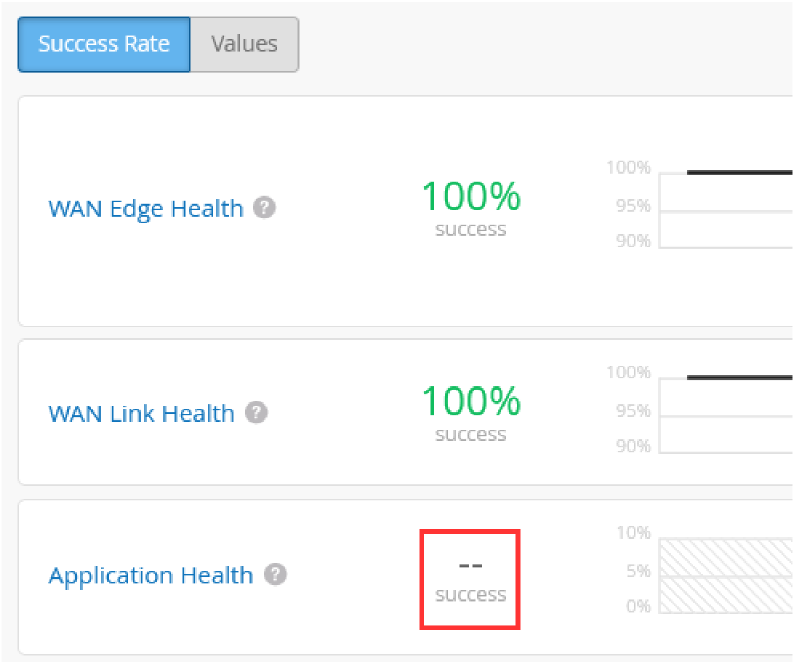
Application SLE

NOTE: We recommend running traffic for at least a week for the Mist AI system to have enough data for analysis.

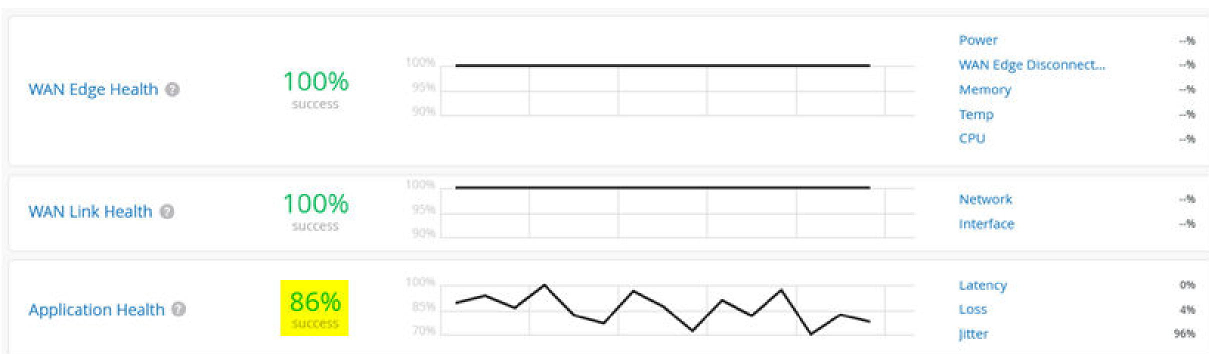
The Application SLE is used to monitor the reachability of applications based on traffic generated by a Session Smart Router. This allows for the automatic collection of monitoring data, which is then sent to the Juniper Mist cloud for analysis and visibility. Unlike SRX Firewalls, which require manual configuration of monitoring probes, Session Smart Routers collect this data automatically.

In lab environments, it may be useful to generate simulated user traffic using scripts. However, in a production setting, it's best to allow actual traffic to flow so you can observe real application usage. This insight helps you fine-tune the required probes based on actual demand.

When monitoring WAN SLEs, ensure that metrics are being populated. In the example screen below, no values are shown yet—likely because sufficient application traffic has not been present. Remember, the system requires sustained traffic over time to collect enough data for meaningful analysis.



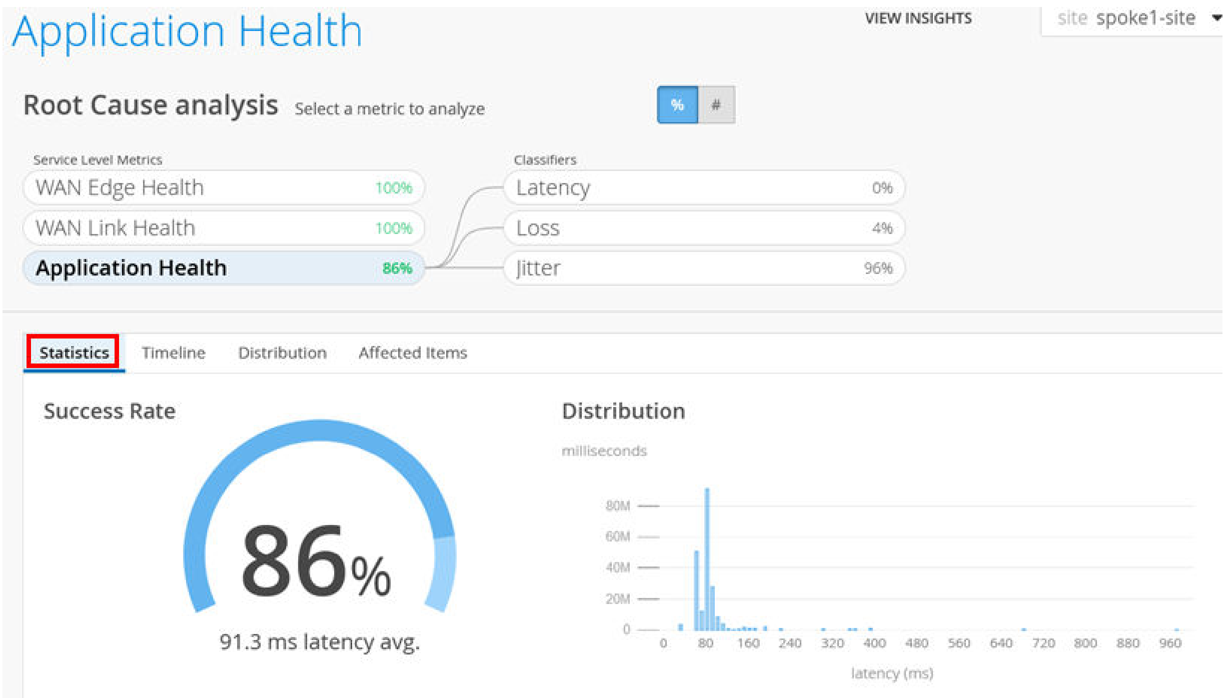
Now, we can check the Application Health SLE. A displayed percentage value means that enough data was collected for analysis.



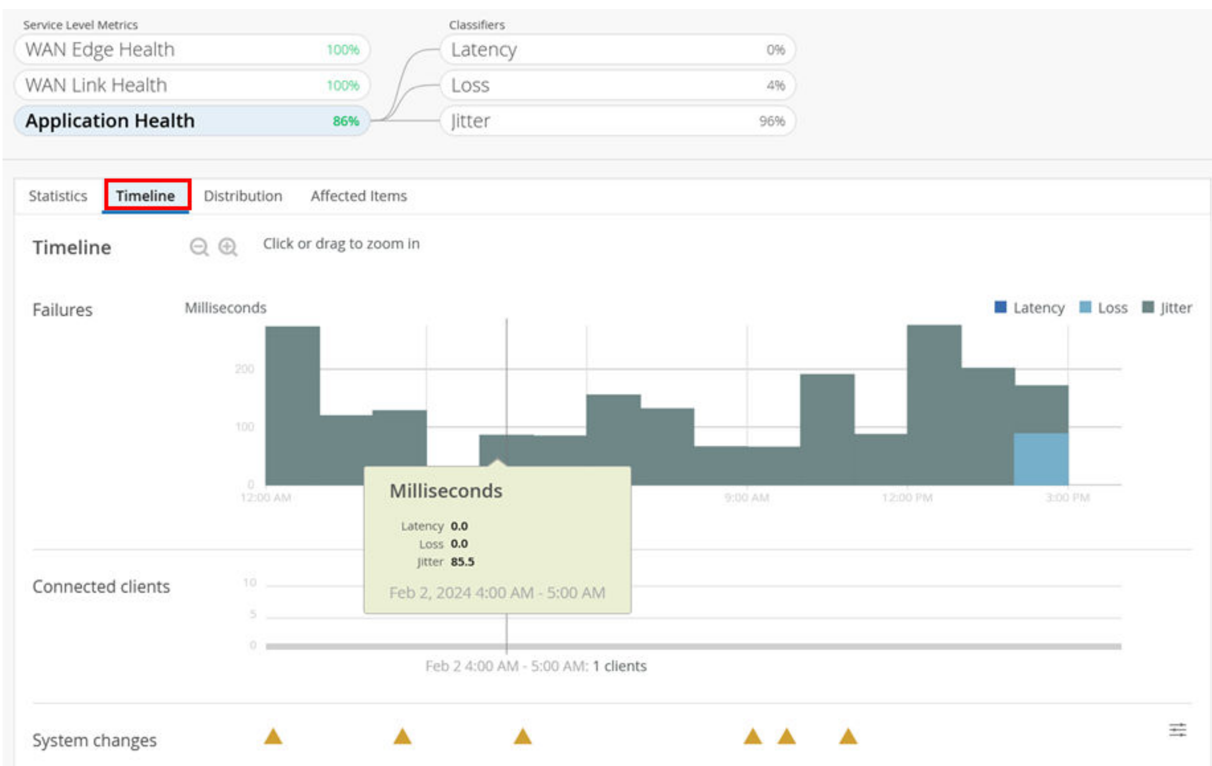
.

In our case, we see 86%. Let's inspect these reports to see who or what is impacted.

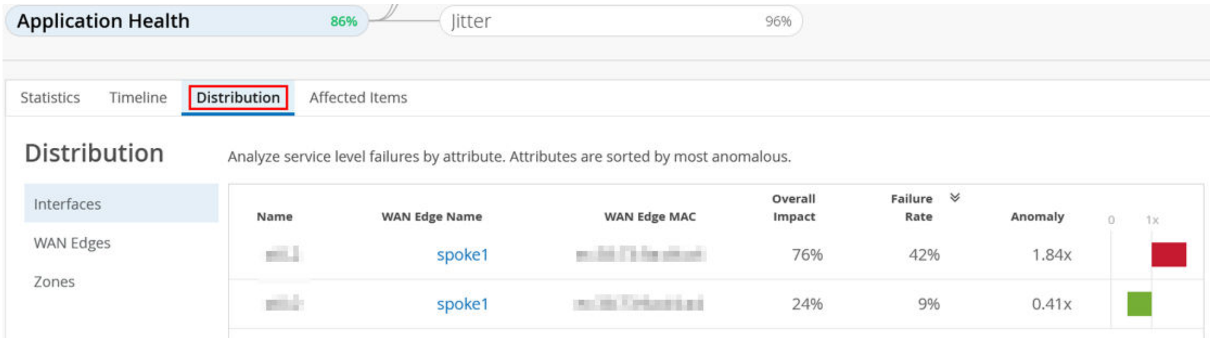
Within the **Application Health** SLE, review the **Statistics** tab to see the distribution of latency values as shown below:



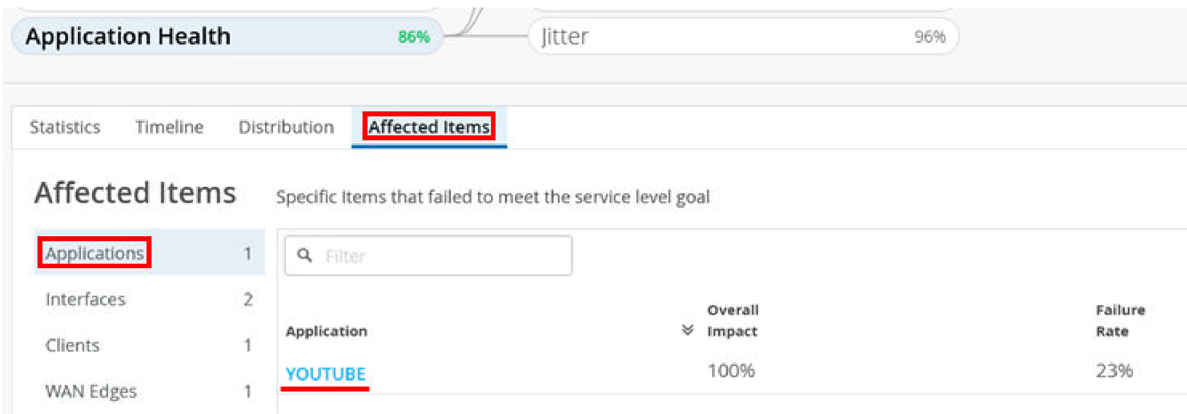
Then, check the **Timeline** tab to see what the impact is and when:



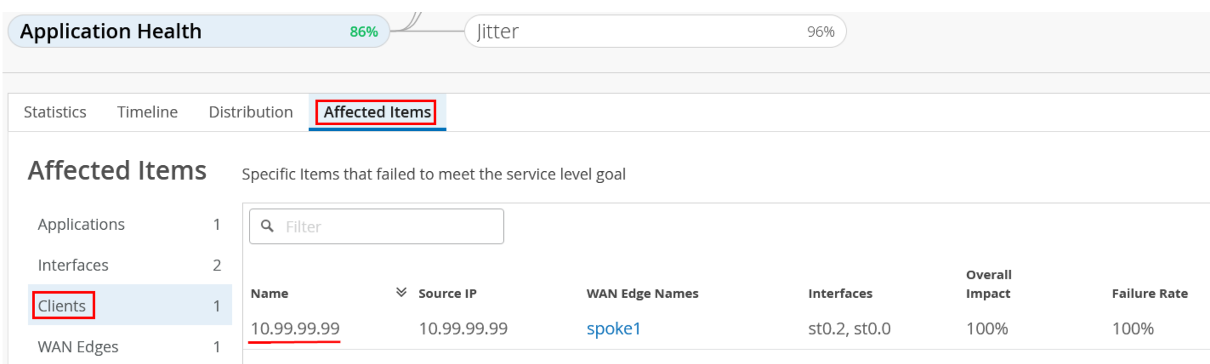
Next, check the **Distribution** tab. Selecting **Interfaces** provides data on the anomaly:



Finally, check **Affected Items** and then **Applications**. In this example, we see issues with YouTube:



Finally, we check the affected users:

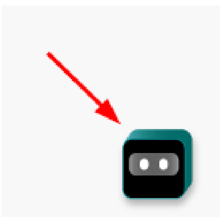


You can also inspect **Interfaces** and **WAN Edges**.

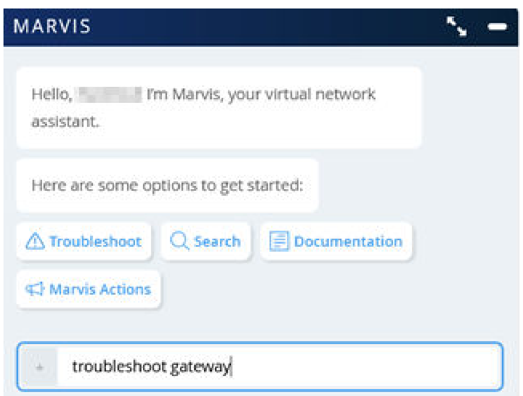
Marvis Conversational Assistant

NOTE: We recommend running traffic for at least a week for the AI system to have enough data for analysis.

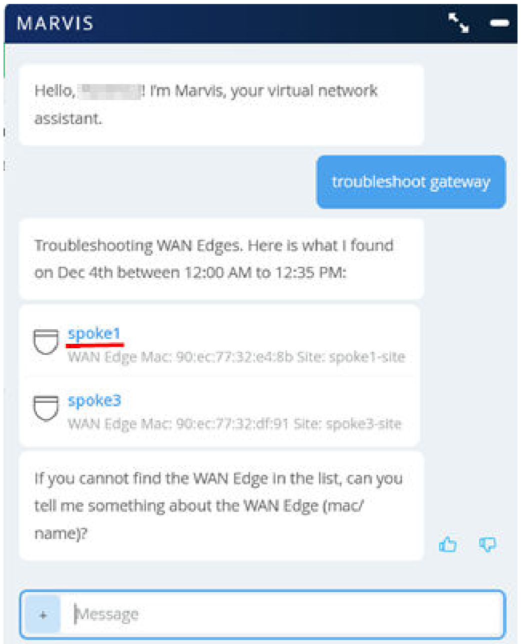
The Marvis Conversational Assistant is in the lower-right corner of your browser window.



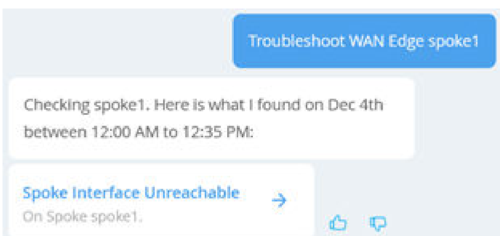
The window that appears has some predefined terms to choose. Enter “troubleshoot gateway” to limit the search to the WAN router:



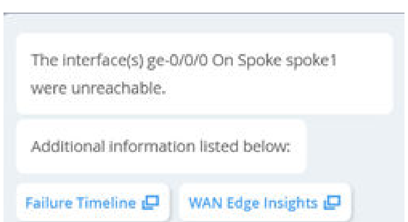
In our case (which may be different in your environment), we get a report about the spokes where we simulated a WAN outage before. Here, we select one of the displayed spokes to get further information:



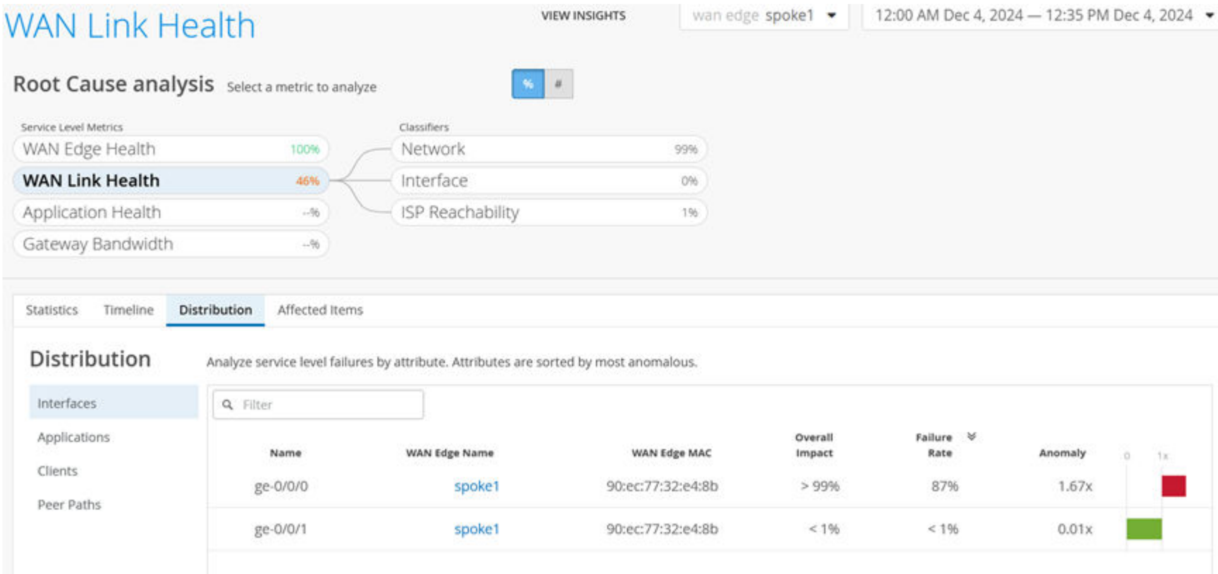
When narrowing down on Spoke1, we see that interfaces became unavailable:



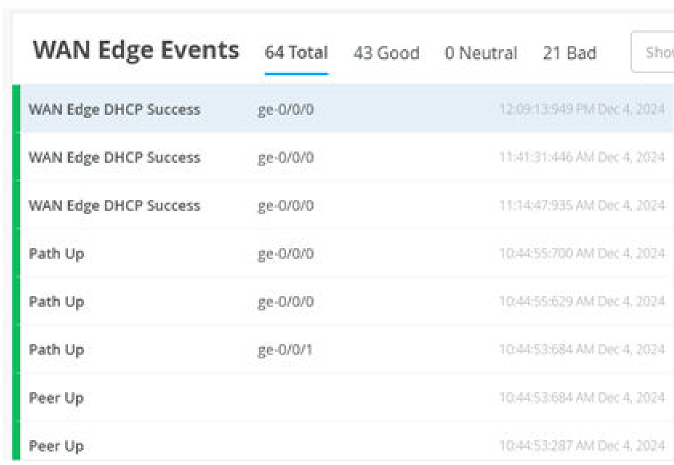
Narrowing down further allows us to select **Failure Timeline** and **WAN Edge Insights**:



When you select **Failure Timeline**, the **WAN Link Health** page opens offering more information:



When you select **WAN Edge Insights**, the Insights page opens and in this case, the Events show when these interfaces came back up and the SVR tunnels towards the hubs were established.



Speed Tests for Session Smart Router

Service Providers (SPs) as well as their end customers install and deploy telecommunication circuits (or paths) to offices, branches, and so on. As Session Smart Routers are deployed at the edge of the customer premises, SPs and customers need to generate traffic to test the speed and performance of these circuits to ensure the quality is being maintained.

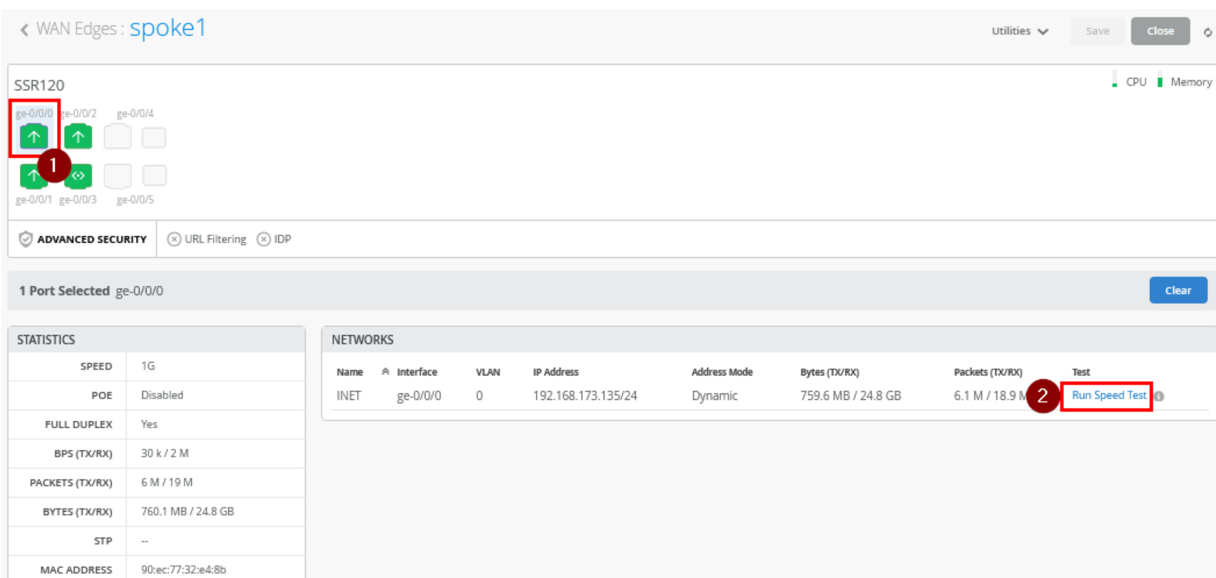
From the Juniper Mist portal, you can run a speed test for a Session Smart Router deployed as a WAN Edge on your network. Speed tests come in handy, for example, when:

- You need to test the speed and performance of the circuit being delivered to the customer.

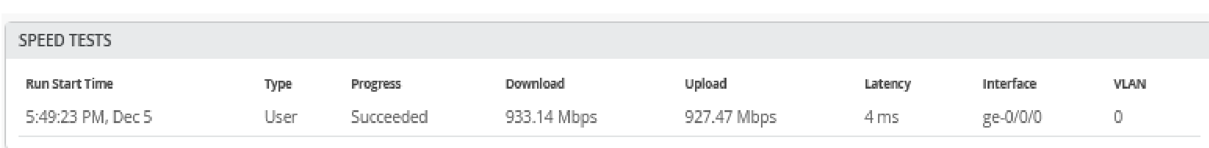
- You need to perform new link qualification to verify that speeds are what the service provider and customer have agreed upon.
- You need to perform on-demand speed tests when you suspect a low link speed is causing link issues.
- You need to run scheduled speed tests to re-test link speeds and ensure performance continues to meet expectations on an ongoing basis.

> **NOTE**: The WAN Edge speed test tool can reliably validate circuit speeds of 1 megabit per second (Mbps) to 1 gigabit per second (Gbps). Circuits exceeding 1Gbps must rely on other tools for validation. The WAN Edge speed test tool does not measure or validate jitter or loss.

In the example below, we select port `ge-0/0/0` as one of the WAN ports to be tested. We then initiate the test command using the **Run Speed Test** link as shown in the figure below:

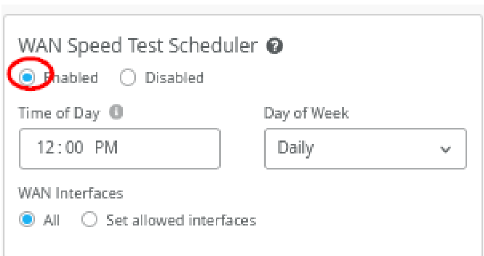


The example results are shared in the figure below:



NOTE: The traffic destinations for this speed test are publicly hosted services that also provide feedback to end users when initiating a test through their browser. This is not a test towards a hub but it's a good test for Internet connectivity overall.

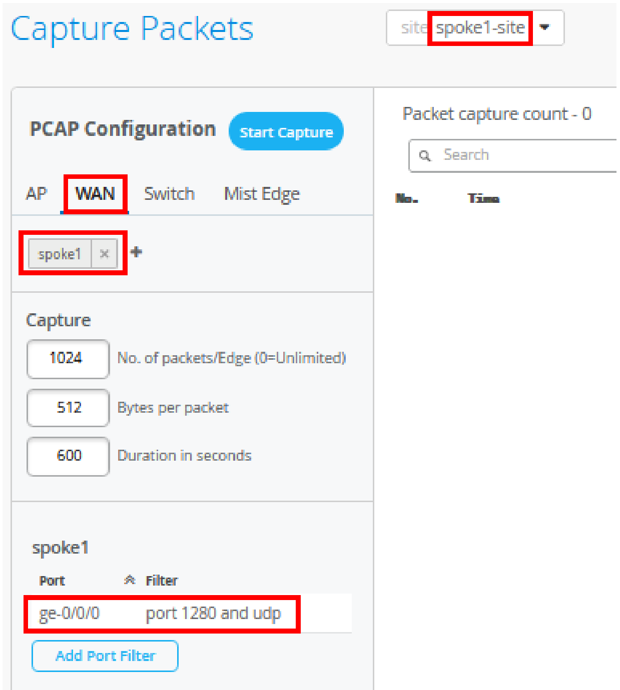
If you want, you can also initiate regular testing schedules. Go to **Organization -> Settings** where you can enable the new **WAN Speed Test Scheduler** and configure the rest.



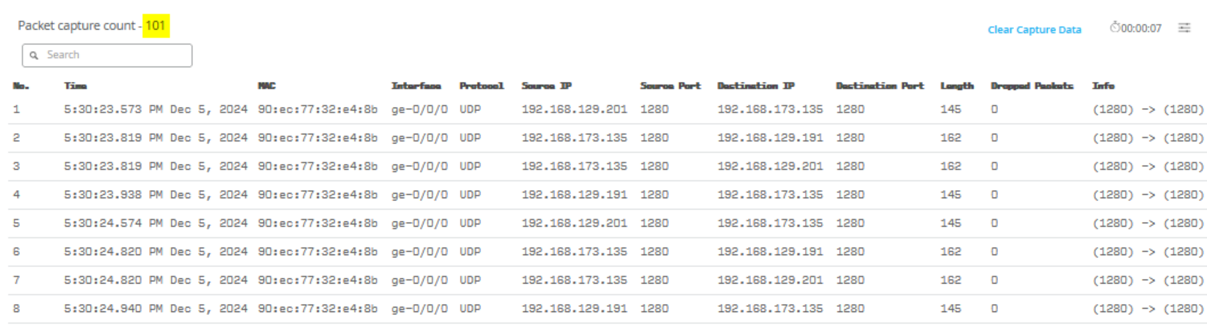
Debugging Using Packet Captures to Collect Remote Traffic Data

Go to **Site -> WAN Edge Packet Capture** and a new pane will open. In our example, we configure the following for packet capture:

- Site=`spoke1-site`
- WAN=`spoke1`
- Capture
    - No. of packets/Edge=`1024` (the default). Do not set this parameter to 0.
    - Bytes per packet=`512`
    - Duration in seconds=`600`
- spoke1
- Port1=`ge-0/0/0`. Our first WAN Interface.
- Filter1=`port 10280 and udp`. We want to capture SVR traffic between spoke and hubs.

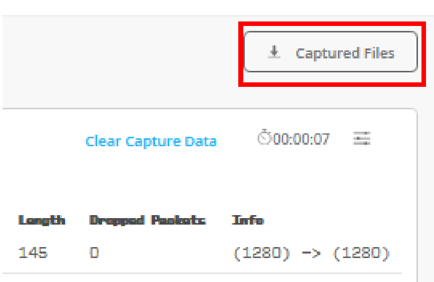


Then, click on **Start Capture**. If your filter captures traffic, it's immediately displayed in the packet capture window as shown in the figure below:

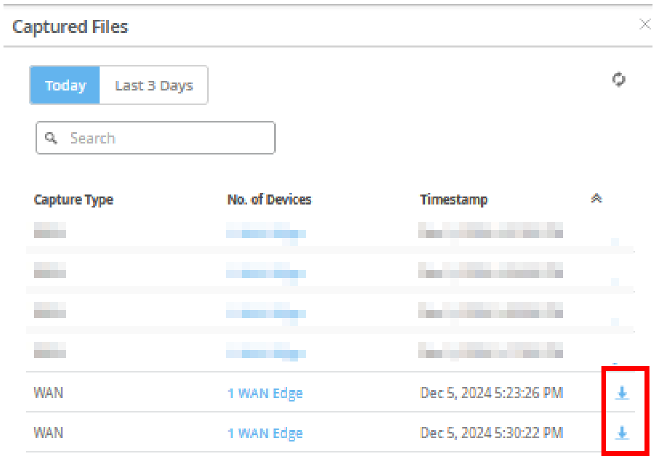


No.	Time	MAC	Interface	Protocol	Source IP	Source Port	Destination IP	Destination Port	Length	Dropped Packets	Info
1	5:30:23.573 PM Dec 5, 2024	90:ec:77:32:e4:8b	ge-0/0/0	UDP	192.168.129.201	1280	192.168.173.135	1280	145	0	(1280) -> (1280)
2	5:30:23.819 PM Dec 5, 2024	90:ec:77:32:e4:8b	ge-0/0/0	UDP	192.168.173.135	1280	192.168.129.191	1280	162	0	(1280) -> (1280)
3	5:30:23.819 PM Dec 5, 2024	90:ec:77:32:e4:8b	ge-0/0/0	UDP	192.168.173.135	1280	192.168.129.201	1280	162	0	(1280) -> (1280)
4	5:30:23.938 PM Dec 5, 2024	90:ec:77:32:e4:8b	ge-0/0/0	UDP	192.168.129.191	1280	192.168.173.135	1280	145	0	(1280) -> (1280)
5	5:30:24.574 PM Dec 5, 2024	90:ec:77:32:e4:8b	ge-0/0/0	UDP	192.168.129.201	1280	192.168.173.135	1280	145	0	(1280) -> (1280)
6	5:30:24.820 PM Dec 5, 2024	90:ec:77:32:e4:8b	ge-0/0/0	UDP	192.168.173.135	1280	192.168.129.191	1280	162	0	(1280) -> (1280)
7	5:30:24.820 PM Dec 5, 2024	90:ec:77:32:e4:8b	ge-0/0/0	UDP	192.168.173.135	1280	192.168.129.201	1280	162	0	(1280) -> (1280)
8	5:30:24.940 PM Dec 5, 2024	90:ec:77:32:e4:8b	ge-0/0/0	UDP	192.168.129.191	1280	192.168.173.135	1280	145	0	(1280) -> (1280)

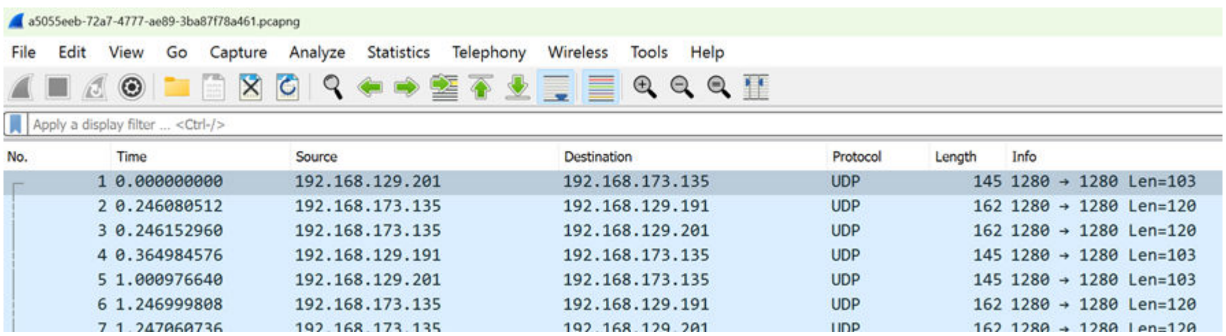
After you have stopped the packet capture, the PCAP file is uploaded (taking ~3 minutes) and you can download the received files.



Now, download the PCAP files.



When you open them in Wireshark you can further analyze the traffic.



Revision History

Table 3: Revision History

Date	Version	Description
July 2025	1.0	Initial publish