Engineering
Simplicity

Campus Fabric IP Clos Using Mist Wired Assurance—Juniper Validated Design (JVD)

Published
2026-02-06

Table of Contents

Campus Fabric IP Clos Using Mist Wired Assurance —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

This document covers how to deploy a campus fabric IP Clos architecture to support a campus networking environment using Juniper Mist™ Wired Assurance. The use case shows how you can deploy a single campus fabric that uses EVPN in the control plane, VXLAN tunnels in the overlay network, and BGP in the underlay with Juniper® Series of High-Performance Access Points integration.

Solution Benefits

IN THIS SECTION

Enterprise networks are undergoing massive transitions to accommodate the growing demand for cloud-ready, scalable, and efficient networks. There's also demand for the plethora of Internet of Things (IoT) and mobile devices. As the number of devices grows, so does network complexity with an ever-greater need for scalability, segmentation, and security. To meet these challenges, you need a network with automation and Artificial Intelligence (AI) for operational simplification. IP Clos networks provide increased scalability and segmentation using a well understood, standards-based approach through EVPN-VXLAN with group-based policies (GBP).

Most traditional campus architectures use single vendor, chassis-based technologies that work well in small, static campuses with few endpoints. However, they are too rigid to support the scalability and changing needs of modern large enterprises. Multi-Chassis Link Aggregation Group (MC-LAG) is a good example of a single vendor technology that addresses the collapsed core deployment model. In this model, two chassis-based platforms are typically in the core of a customer's network and deployed to handle all Layer 2 (L2) and Layer 3 (L3) requirements while providing an active-backup resiliency environment. An MC-LAG does not interoperate between vendors and is limited to two devices. The lack of vendor interoperability creates vendor lock-in.

A Juniper Networks EVPN-VXLAN fabric is a highly scalable architecture that is simple, programmable, and built on a standards-based architecture (https://www.rfc-editor.org/rfc/rfc8365) that is common across campuses and data centers.

The Juniper Networks campus architecture uses an L3 IP-based underlay network and an EVPN-VXLAN overlay network. Broadcast, unknown unicast, and multicast (BUM) traffic is handled natively by EVPN and eliminates the need for Spanning Tree (STP) or Rapid Spanning Tree Protocols (RSTP). A flexible overlay network based on VXLAN tunnels combined with an EVPN control plane efficiently provides L3 or L2 connectivity. This architecture decouples the virtual topology from the physical topology, which improves network flexibility and simplifies network management. Endpoints that require L2 adjacency, such as IoT devices, can be placed anywhere in the network and remain connected to the same logical L2 network.

With an EVPN-VXLAN campus architecture, you can easily add core, distribution, and access layer devices as your business grows without a need for redesigning your network. As EVPN-VXLAN is vendor-agnostic, you can use the existing access layer infrastructure and gradually migrate to access layer switches. This supports EVPN-VXLAN capabilities once the core and access part of the network is deployed. A distribution switch layer between access and core is optional and recommended for scale designs with multiple PoDs.

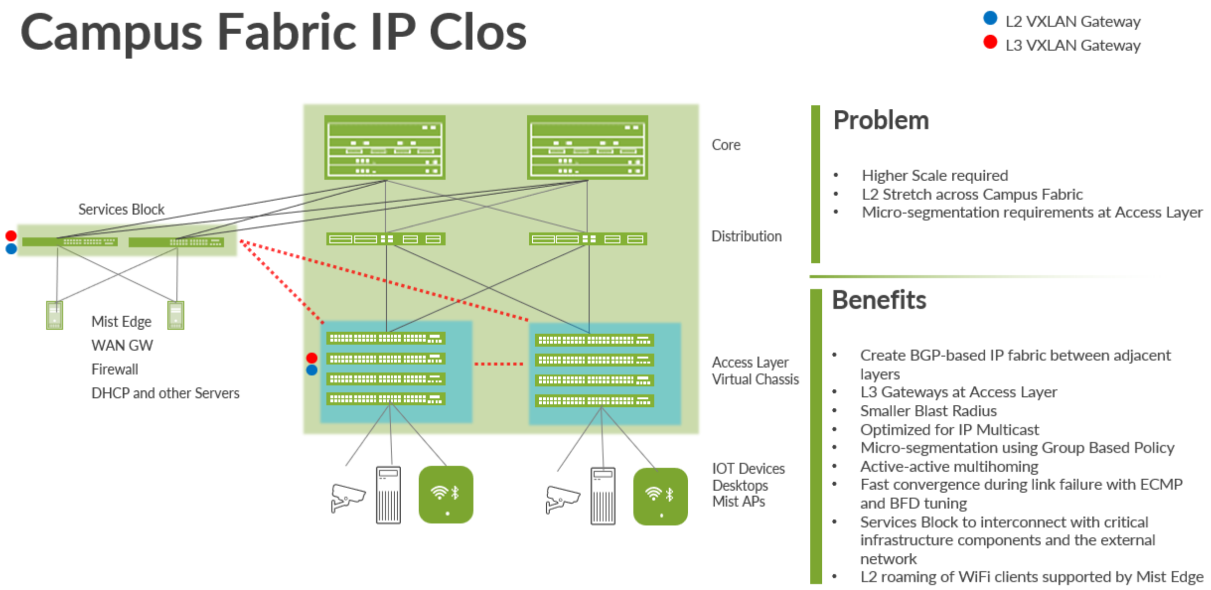
Benefits of Campus Fabric IP Clos

- With the increasing number of devices connecting to the network, you need to scale your campus network rapidly without adding complexity. Many IoT devices have limited networking capabilities and require L2 adjacency across buildings and campuses. Traditionally, this problem was solved by extending virtual LANs (VLANs) between endpoints using data plane-based flood and learning mechanisms inherent with Ethernet switching technologies. The traditional Ethernet switching approach is inefficient because it leverages broadcast and multicast technologies to announce Media Access Control (MAC) addresses. It is also difficult to manage because you need to configure and manually manage VLANs to extend them to new network ports. This problem increases multifold when you take into consideration the explosive growth of mobile and IoT devices.

- Campus fabrics have an underlay topology with a routing protocol that ensures loopback interface reachability between nodes. Devices participating in EVPN-VXLAN function as VXLAN tunnel endpoints (VTEPs) that encapsulate and decapsulate the VXLAN traffic. A VTEP represents a construct within the switching platform that originates and terminates VXLAN tunnels. In addition, these devices route and bridge packets in and out of VXLAN tunnels as required.

- The campus fabric IP Clos extends the EVPN fabric to connect VLANs across multiple buildings or floors of a single building. This is done by stretching the L2 VXLAN network with routing occurring in the access device instead of in the core (Centrally-Routed Bridging (CRB)) or distribution (Edge Routed Bridging (ERB)) devices.

Figure 1: Campus Fabric IP Clos



Campus Fabric IP Clos

● L2 VXLAN Gateway
● L3 VXLAN Gateway

Problem

- Higher Scale required
- L2 Stretch across Campus Fabric
- Micro-segmentation requirements at Access Layer

Benefits

- Create BGP-based IP fabric between adjacent layers
- L3 Gateways at Access Layer
- Smaller Blast Radius
- Optimized for IP Multicast
- Micro-segmentation using Group Based Policy
- Active-active multihoming
- Fast convergence during link failure with ECMP and BFD tuning
- Services Block to interconnect with critical infrastructure components and the external network
- L2 roaming of WiFi clients supported by Mist Edge

An IP Clos network encompasses the distribution, core, and access layers of your topology.

An EVPN-VXLAN fabric solves the problems of previous architectures and provides the following benefits:

- Reduced flooding and learning—Control plane-based L2 and L3 learning reduces the flood and learn issues associated with data plane learning. Learning MAC addresses in the forwarding plane has an adverse impact on network performance as the number of endpoints grows. This is because more management traffic consumes the bandwidth which leaves less bandwidth available for production traffic. The EVPN control plane handles the exchange and learning of MAC addresses through eBGP routing, rather than an L2 forwarding plane.

- Scalability—More efficient control plane-based L2 and L3 learning. For example, in a campus fabric IP Clos, core switches only learn the access layer switches addresses instead of the device endpoint addresses.

- Consistency—A universal EVPN-VXLAN-based architecture across disparate campus and data center deployments enables a seamless end-to-end network for endpoints and applications.
- Group-based policies—With GBP, you can enable microsegmentation with EVPN-VXLAN to provide traffic isolation within and between broadcast domains as well as simplify security policies across a campus fabric.
- Location-agnostic connectivity—The EVPN-VXLAN campus architecture provides a consistent endpoint experience no matter where the endpoint is located. Some endpoints require L2 reachability, such as legacy building security systems or IoT devices. The VXLAN overlay provides an L2 extension across campuses without any changes to the underlay network. Juniper Networks uses optimal BGP timers between the adjacent layers of the campus fabric with Bidirectional Forwarding Detection (BFD) that supports fast convergence in the event of a node or link failure and equal-cost multipath (ECMP). For more information, see Configuring Per-Packet Load Balancing.

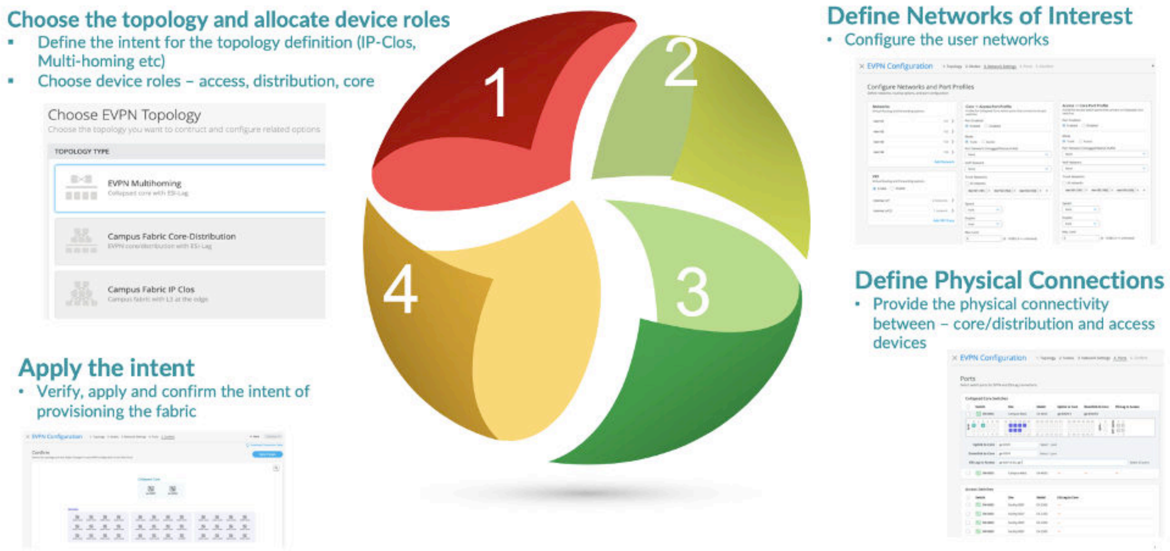
Technical overview

IN THIS SECTION

Campus Fabric IP Clos High-Level Architecture

The campus fabric, with an EVPN-VXLAN architecture, decouples the overlay network from the underlay network. This approach addresses the needs of the modern enterprise network by allowing network administrators to create logical L2 networks across one or more L3 networks. By configuring different routing instances, you can enforce the separation of virtual networks because each routing instance has its own separate routing and switching table.

The Juniper Mist™ portal workflow makes it easy to create campus fabrics.



Choose the topology and allocate device roles

- Define the intent for the topology definition (IP-Clos, Multi-homing etc)
- Choose device roles – access, distribution, core

Define Networks of Interest

- Configure the user networks

Define Physical Connections

- Provide the physical connectivity between – core/distribution and access devices

Apply the intent

- Verify, apply and confirm the intent of provisioning the fabric

1
2
3
4
Choose EVPN Topology
Choose the topology you want to contruct and configure related options
TOPOLOGY TYPE
EVPN Multihoming
Collapsed core with ESI-Lag
Campus Fabric Core-Distribution
EVPN core/distribution with ESI-Lag
Campus Fabric IP Clos
Campus fabric with L3 at the edge
EVPN Configuration
Configure Networks and Port Profiles
EVPN Configuration
Ports
EVPN Configuration
Confirm

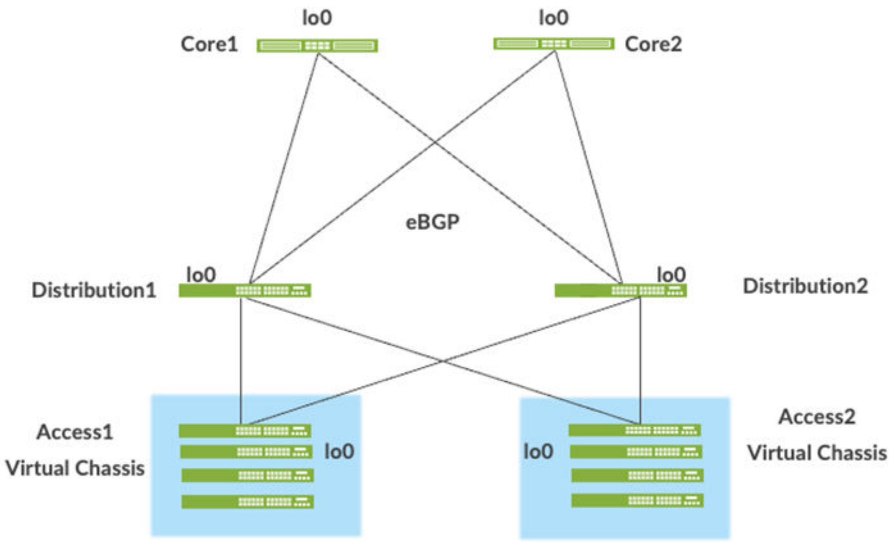
Underlay Network

An EVPN-VXLAN fabric architecture makes the network infrastructure simple and consistent across campuses and data centers. All the core, distribution, and access devices must be connected using an L3 infrastructure. We recommend deploying a Clos-based IP fabric to ensure predictable performance and to enable a consistent, scalable architecture.

You can use any L3 routing protocol to exchange loopback addresses between the access, core, and distribution devices. BGP provides benefits such as better prefix filtering, traffic engineering, and route tagging. We are using eBGP as the underlay routing protocol in this example. Mist automatically provisions Private Autonomous System numbers and all BGP configurations for the underlay and overlay for only the campus fabric. There are options to provide additional BGP speakers to allow you to peer with external BGP peers.

Underlay BGP is used to learn loopback addresses from peers so that the overlay BGP can establish neighbors using the loopback address. The overlay is then used to exchange EVPN routes.

Figure 2: Point-to-Point/31 Links Between Adjacent Layers Running eBGP

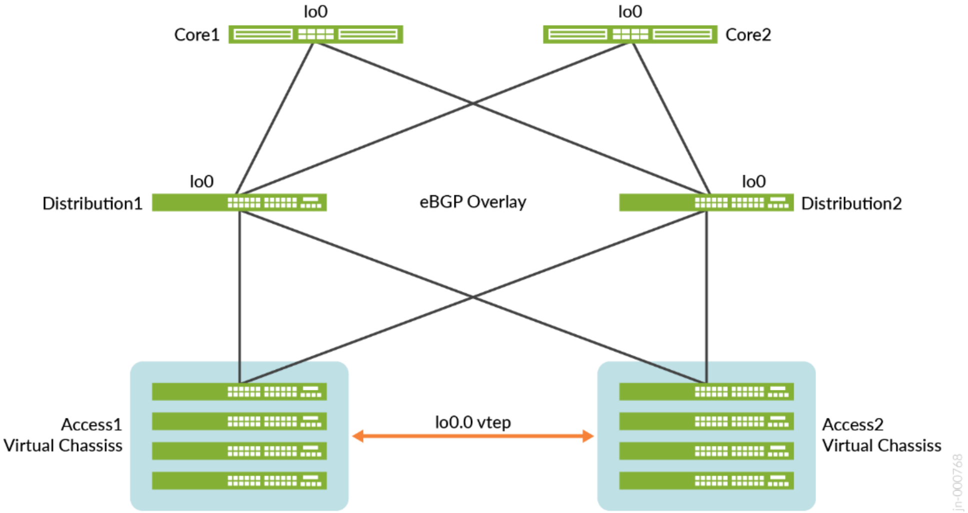


Network overlays enable connectivity and addressing independent of the physical network. Ethernet frames are wrapped in UDP/IP datagrams, which are encapsulated into IP for transport over the underlay. VXLAN enables virtual L2 subnets or VLANs to span underlying physical L3 network.

In a VXLAN overlay network, each L2 subnet or segment is uniquely identified by a Virtual Network Identifier (VNI). A VNI segments traffic the same way that a VLAN ID does. This mapping occurs on the access switches and border gateway, which can reside on the core or service block. As is the case with VLANs, endpoints within the same virtual network can communicate directly with each other.

Endpoints in different virtual networks require a device that supports inter-VXLAN routing, which is typically a router, or a high-end switch known as an L3 gateway. The entity that performs VXLAN encapsulation and decapsulation is called a VTEP. Each VTEP is known as the L2 gateway and is typically assigned with the device's loopback address. This is also where VXLAN (commonly known as VNI) to VLAN mapping takes place.

Figure 3: VXLAN VTEP Tunnels



VXLAN can be deployed as a tunnelling protocol across an L3 IP campus fabric without a control plane protocol. However, the use of VXLAN tunnels alone does not change the flood and learn behavior of the Ethernet protocol.

The two primary methods for using VXLAN without a control plane protocol are static unicast VXLAN tunnels and VXLAN tunnels. These methods are signaled with a multicast underlay and do not solve the inherent flood and learn problem. These methods are also difficult to scale in large, multitenant environments. These methods are not in the scope of this JVD.

Understanding EVPN

EVPN is a BGP extension to distribute endpoint reachability information such as MAC and IP addresses to other BGP peers. This control plane technology uses Multiprotocol BGP (MP-BGP) for MAC and IP address endpoint distribution, where MAC addresses are treated as Type 2 EVPN routes. EVPN enables devices acting as VTEPs to exchange reachability information with each other about their endpoints.

Juniper supported EVPN Standards: https://www.juniper.net/documentation/us/en/software/junos/evpn-vxlan/topics/concept/evpn.html

What is EVPN-VXLAN: https://www.juniper.net/us/en/research-topics/what-is-evpn-vxlan.html

The benefits of using EVPNs include:

- MAC address mobility
- Multitenancy

- Load balancing across multiple links
- Fast convergence
- High availability
- Scale
- Standards-based interoperability

EVPN provides multipath forwarding and redundancy through an all-active model. The access layer can connect to two or more distribution devices and forward traffic using all the links. If an access link or distribution device fails, traffic flows from the access layer toward the distribution layer using the remaining active links. For traffic in the other direction, remote distribution devices update their forwarding tables to send traffic to the remaining active distribution devices connected to the multihomed Ethernet segment.

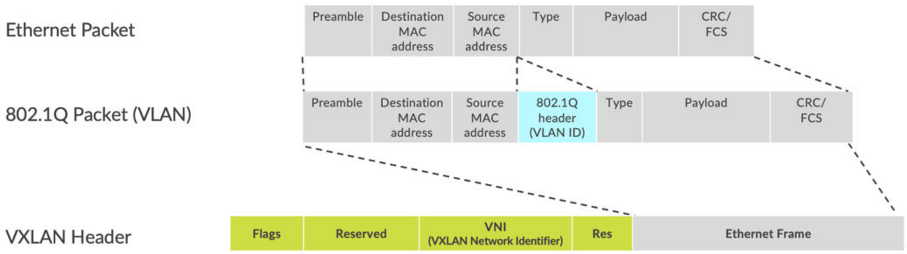
The technical capabilities of EVPN include:

- Minimal flooding—EVPN creates a control plane that shares end-host MAC addresses between VTEPs.
- Multihoming—EVPN supports multihoming for client devices. A control protocol like EVPN that enables synchronization of endpoint addresses between the access switches is needed to support multihoming because traffic traveling across the topology needs to be intelligently moved across multiple paths.
- Aliasing—EVPN leverages all-active multihoming when connecting devices to the access layer of a campus fabric. The connection of the multihomed access layer switches is called ESI-LAG, while the access layer devices connect to each access switch using standard LACP.
- Split horizon—Split horizon prevents the looping of BUM traffic in a network. With split horizon, a packet is never sent back over the same interface it was received on, which prevents loops.

Overlay Network (Data Plane)

VXLAN is the overlay data plane encapsulation protocol that tunnels Ethernet frames between network endpoints over the underlay network. A device that performs VXLAN encapsulation and decapsulation for the network is referred to as a VTEP. Before a VTEP sends a frame into a VXLAN tunnel, it wraps the original frame in a VXLAN header that includes a Virtual Network Identifier (VNI). The VNI maps the packet to the original VLAN at the ingress switch. After applying a VXLAN header, the frame is encapsulated into a UDP/IP datagram for transmission to the remote VTEP over the IP fabric, where the VXLAN header is removed and the VNI-to-VLAN translation happens at the egress switch.

Figure 4: VXLAN Header



VLAN	VXLAN
4K	16M

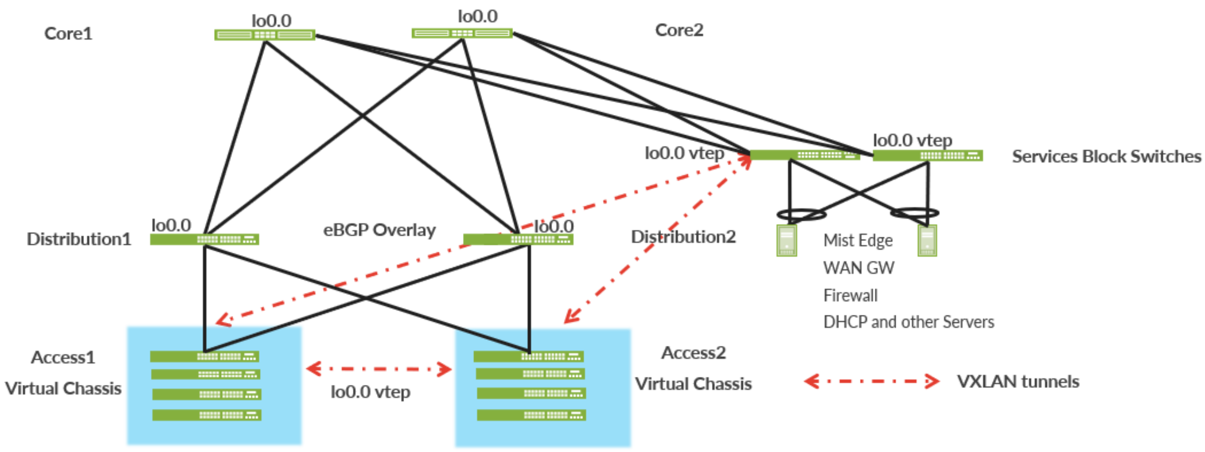
VTEPs are software entities tied to the devices' loopback address that source and terminate VXLAN tunnels. VXLAN tunnels in an IP Clos fabric are provisioned on the following:

- Access switches to extend services across the campus fabric IP Clos.
- Core switches, when acting as a border router, interconnect the campus fabric with the outside network.
- Service block devices that interconnect the campus fabric with the outside network.

Overlay Network (Control Plane)

Multiprotocol Border Gateway Protocol (MP-BGP) with EVPN signaling acts as the overlay control plane protocol. Adjacent layer switches set up eBGP peers using their loopback addresses with next hops announced by the underlay BGP sessions. For example, core and distribution devices establish eBGP sessions between each other while the access and distribution devices establish eBGP sessions between each other. When there is an L2 forwarding table update on any switch participating in the campus fabric, it sends a BGP update message with the new MAC route to other devices in the fabric. Those devices then update their local EVPN database and routing tables.

Figure 5: EVPN VXLAN Overlay Network with a Service Block



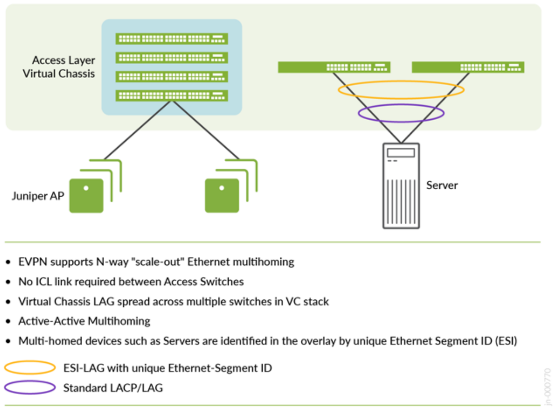
Resiliency and Load Balancing

We support BFD as part of the BGP protocol implementation. This provides fast convergence in the event of a device or link failure without relying on the routing protocol's timers. Juniper Mist configures BFD with minimum intervals of 1000ms in the underlay and overlay respectively. Load balancing, per packet by default, is supported across all links within the campus fabric using equal-cost multipath (ECMP) routing enabled at the forwarding plane.

Ethernet Segment Identifier (ESI)

When devices such as servers and access points are multihomed to two or more switches at the access layer in a campus fabric, an ESI-LAG is formed on the access layer devices. This ESI is a 10-octet integer that identifies the Ethernet segment amongst all access layer switches participating in the ESI. MP-BGP is the control plane protocol used to coordinate this information. An ESI-LAG enables link failover in the event of a bad link, supports active-active load balancing, and is automatically assigned by Juniper Mist.

Figure 6: Device Resiliency and Load Balancing

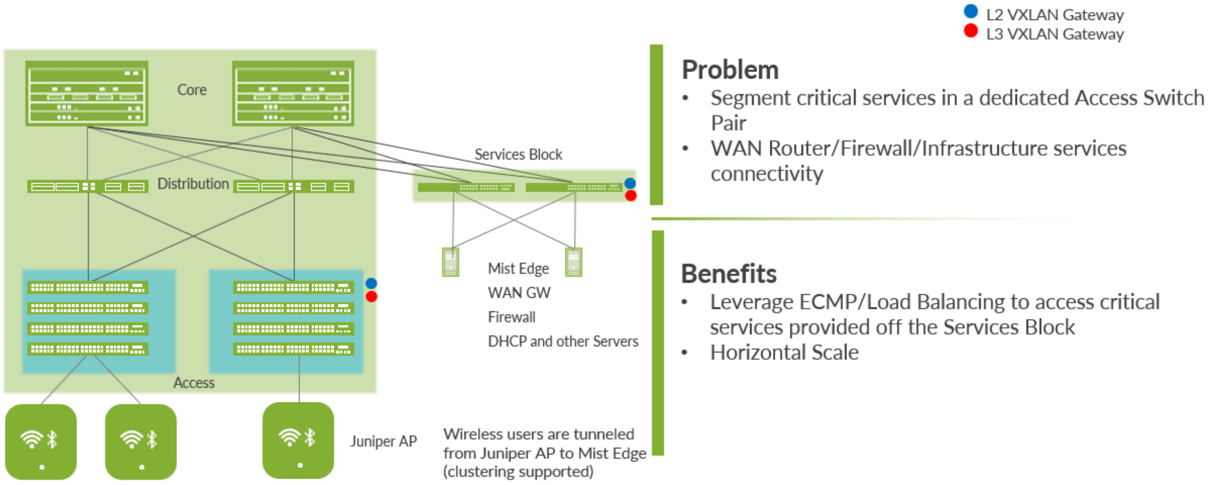


- EVPN supports N-way "scale-out" Ethernet multihoming
- No ICL link required between Access Switches
- Virtual Chassis LAG spread across multiple switches in VC stack
- Active-Active Multihoming
- Multi-homed devices such as Servers are identified in the overlay by unique Ethernet Segment ID (ESI)

Service Block

You need to position critical infrastructure services off of a dedicated pair of Juniper Networks switches. This can include WAN and firewall connectivity, RADIUS, and DHCP servers, for example. If you need to deploy a lean core, the dedicated service block mitigates the need for the core to support encapsulation and de-encapsulation of VXLAN tunnels, multiple routing instances, and additional L3 routing protocols. The service block border capability is supported directly off of the core layer or as a dedicated pair of switches.

Figure 7: Service Block



Problem

- Segment critical services in a dedicated Access Switch Pair
- WAN Router/Firewall/Infrastructure services connectivity

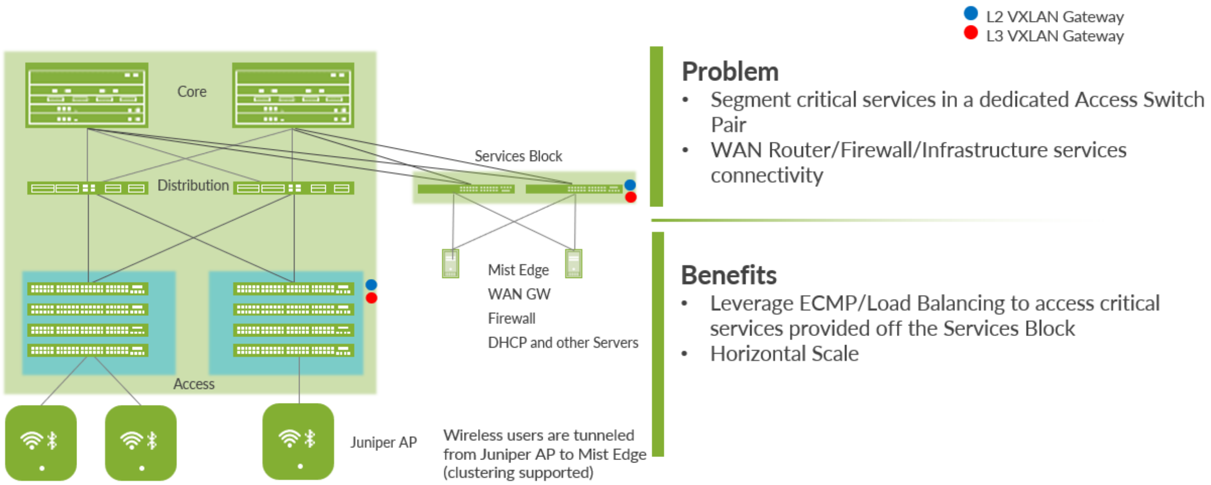
Benefits

- Leverage ECMP/Load Balancing to access critical services provided off the Services Block
- Horizontal Scale

Access Layer

The access layer provides network connectivity to end-user devices, such as personal computers, VoIP phones, printers, IoT devices, as well as connectivity to wireless access points. The EVPN-VXLAN network extends all the access layer switches.

Figure 8: Endpoint Access



Problem

- Segment critical services in a dedicated Access Switch Pair
- WAN Router/Firewall/Infrastructure services connectivity

Benefits

- Leverage ECMP/Load Balancing to access critical services provided off the Services Block
- Horizontal Scale

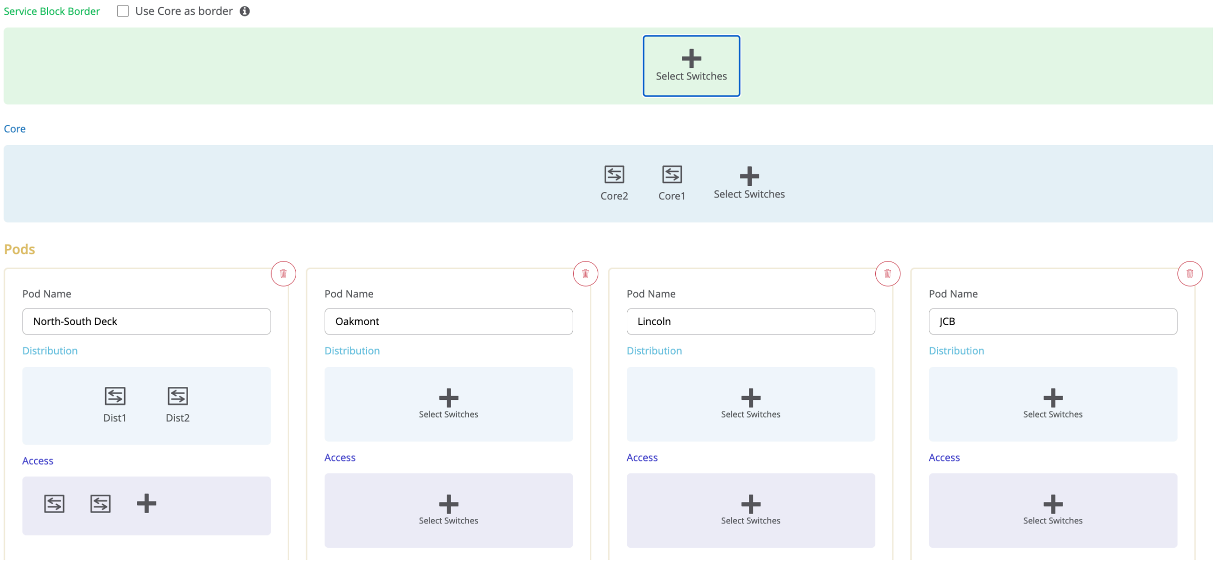
In this example, each access switch or Virtual Chassis is multihomed to two or more distribution switches. Juniper's Virtual Chassis reduces the number of ports required on distribution switches and optimizes the availability of fiber throughout the campus. The Virtual Chassis supports up to 10 member switches (depending on the switch model) and is managed as a single device. See https://www.juniper.net/documentation/us/en/software/junos/vcf-best-practices-guide/vcf-best-practices-guide.pdf.

With EVPN running as the control plane protocol, any access switch or Virtual Chassis device can enable active-active multihoming to the distribution layer. EVPN provides a standards-based multihoming solution that scales horizontally across any number of access layer switches.

Single or Multiple PoD Design

Juniper Mist campus fabric supports deployments with only one point of delivery (PoD) (formally called Site-Design) or multiple PoDs. The organizational deployment shown below, targets enterprises who need to align with a multi-PoD structure:

Figure 9: Campus Fabric Multi PoD Design



Juniper Access Points

In our network, we choose Juniper access points (APs) as our preferred AP devices. They are designed from the ground up to meet the stringent networking needs of the modern cloud and smart device era. Juniper Mist delivers unique capabilities for both wired and wireless LAN:

- Wired and Wireless Assurance—Juniper Mist is enabled with wired and wireless assurance. Once configured, Service-Level Expectations (SLE) for key wired and wireless performance metrics such as throughput, capacity, roaming, and uptime are monitored in the Juniper Mist platform. This JVD uses Juniper Mist Wired Assurance cloud services.

- Marvis® Virtual Network Assistant—An integrated AI engine that provides rapid wired and wireless troubleshooting, trending analysis, anomaly detection, and proactive problem remediation.

Juniper Mist Edge

For large campus networks, Juniper Mist™ Edge provides seamless roaming through on-premises tunnel termination of traffic to and from the Juniper APs. Juniper Mist Edge extends select microservices to the customer premises while using the Juniper Mist cloud and its distributed software architecture for scalable and resilient operations, management, troubleshooting, and analytics. Juniper Mist Edge is deployed as a standalone appliance with multiple variants for different-size deployments.

Evolving IT departments look for a cohesive approach for managing wired, wireless, and wan networks. This full-stack approach simplifies and automates operations, provides end-to-end troubleshooting, and ultimately evolves into the Self-Driving Network™. The integration of the Juniper Mist platform in this JVD addresses both full-stack deployments and automation.

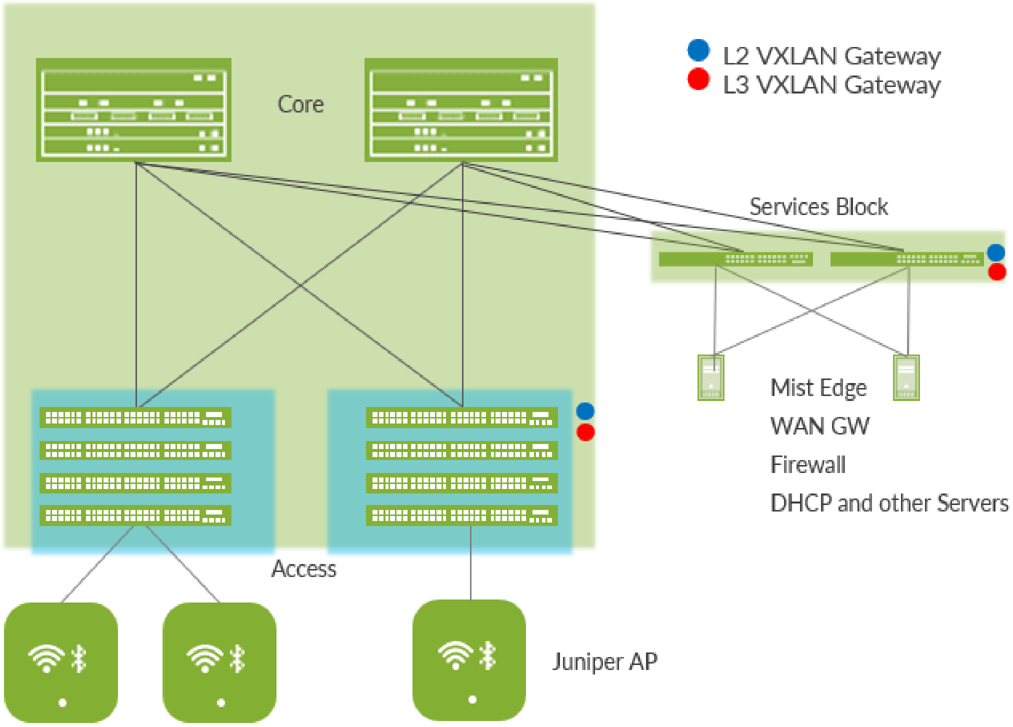
Campus Fabric IP Clos Deployment Types

Juniper Mist's Wired Assurance supports 3-stage and 5-stage IP Clos deployments depending on scale and PoD design.

3-Stage IP Clos

The 3-stage IP Clos is targeted for deployments that do not require a distribution layer and have smaller-scale requirements.

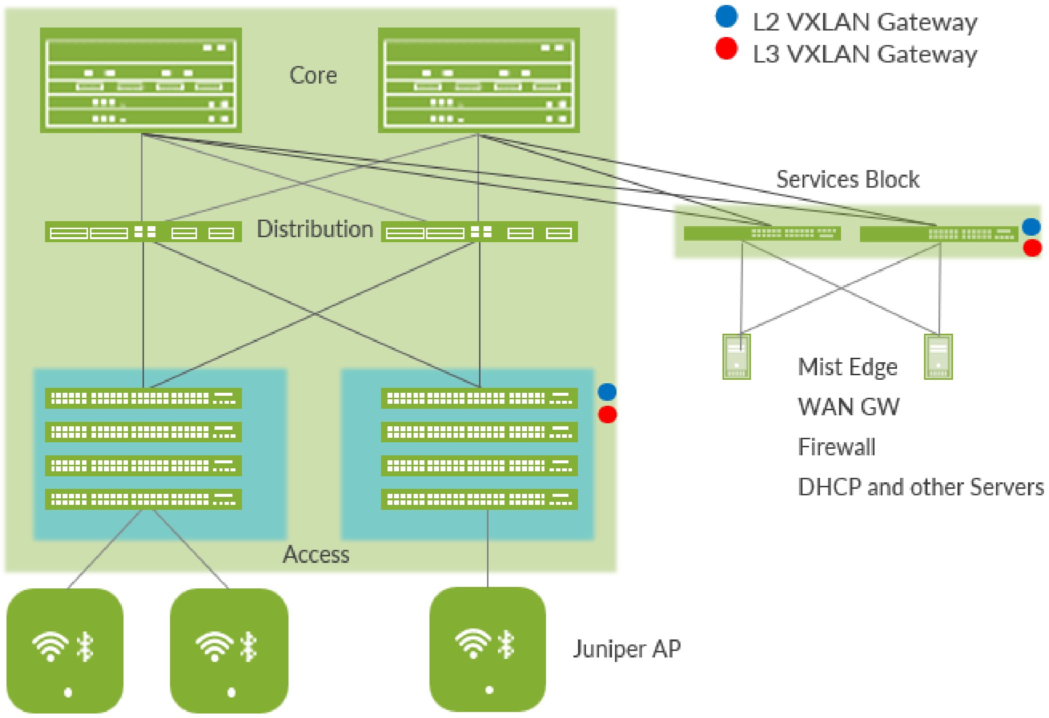
Figure 10: 3-Stage IP Clos



5-Stage IP Clos

The 3-stage IP Clos is the recommended deployment with access, distribution and core layer. It is also recommended when using multiple PoDs that then peer using the core switches (not shown in the below figure).

Figure 11: 5-Stage IP Clos



Supported Platforms for Campus Fabric IP Clos

Table 1 on page 16 lists the supported platforms for campus fabric IP Clos deployments.

Table 1: Supported Platforms for Campus Fabric IP Clos Deployments

Supported Platforms	
Campus Fabric IP Clos Deployment	Supported Platforms
Access layer	EX4100 EX4300-MP EX4400
Distribution layer	EX4400-24X EX4650 QFX5120 QFX5130 QFX5700 EX92xx
Core layer	EX4650 QFX5120 QFX5130 QFX5700 QFX10000 EX92xx
Service block	EX4400-24X EX4650 QFX5120 QFX5130 QFX5700 EX92xx

NOTE: A hardware limitation on the Juniper Networks® EX4300-MP Switch does not allow it to be used for VXLAN-GBP. Consider the Juniper Networks® EX4100 or Juniper Networks® EX4400 Switches for such a feature.

Juniper Mist Wired Assurance

Juniper Mist Wired Assurance is a cloud service that brings automated operations and service levels to the campus fabric for switches, IoT devices, access points, servers, and printers. It is about simplifying every step of the way, starting from Day 0 for seamless onboarding and auto-provisioning through Day 2 and beyond for operations and management. Juniper Networks® EX Series Switches provide Junos streaming telemetry that enable the insights for switch health metrics and anomaly detection, as well as Mist AI™ capabilities.

Mist's AI engine and virtual network assistant, Marvis, further simplifies troubleshooting while streamlining helpdesk operations by monitoring events and recommending actions. Marvis is one step towards the Self-Driving Network, turning insights into actions and transforming IT operations from reactive troubleshooting to proactive remediation.

Juniper Mist cloud services are 100% programmable using open APIs for full automation and/or integration with your operational support systems. For example, IT applications such as ticketing systems and IP management systems.

Juniper Mist delivers unique capabilities for the WAN, LAN, and wireless networks:

- A UI or API-driven configuration at scale.
- Service-level expectations (SLEs) for key performance metrics such as throughput, capacity, roaming, and uptime.
- Marvis—An integrated AI engine that provides rapid troubleshooting of full-stack network issues, trending analysis, anomaly detection, and proactive problem remediation.
- A single management system.
- License management.
- Premium analytics for long term trending and data storage.

To learn more about Juniper Mist Wired Assurance, see the following datasheet: https://www.juniper.net/content/dam/www/assets/datasheets/us/en/cloud-services/juniper-mist-wired-assurance-datasheet.pdf

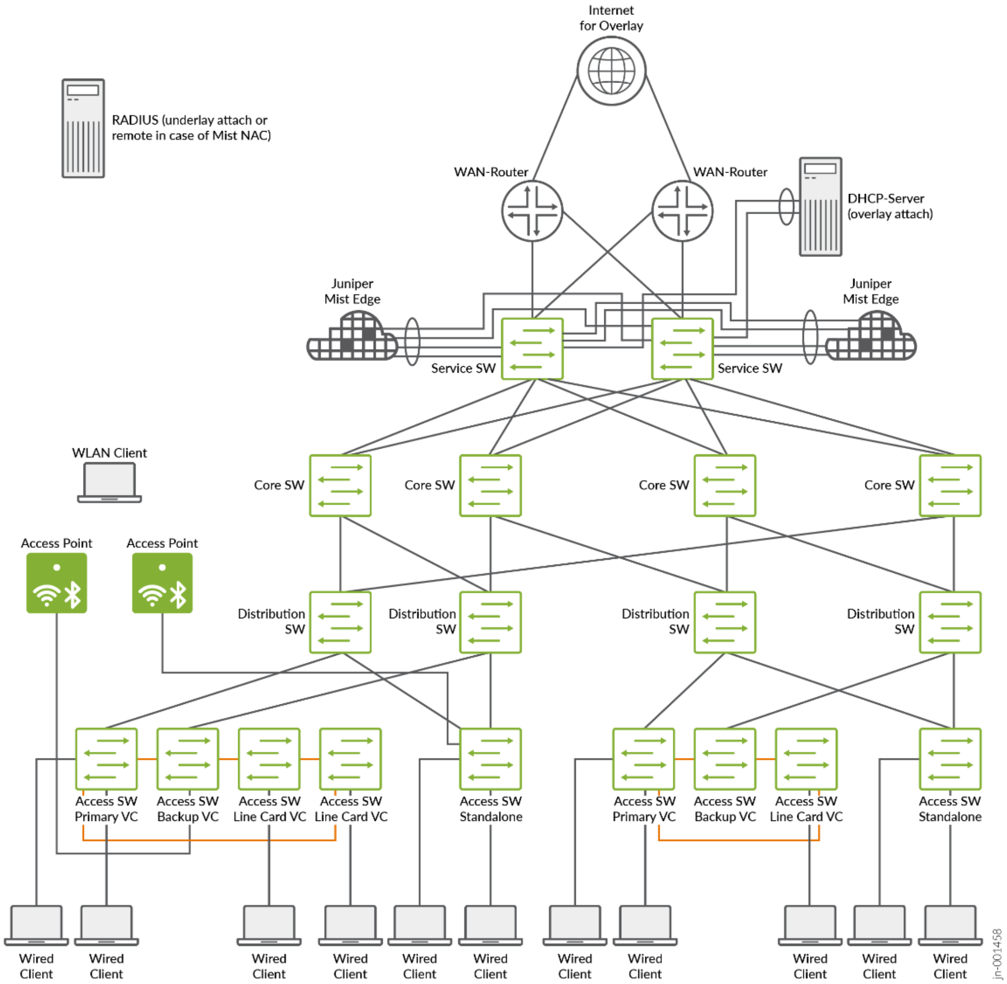
Validation Framework

IN THIS SECTION

Test Bed

In the diagram below, you will see the suggested topology used for the phase 3 lab evaluating an IP Clos fabric design with multiple PoDs including 4 core switches as shown in Figure 12 on page 19 .

Figure 12: Topology 1 JVD Lab Design Phase3



Using Topology 1, one can evaluate the following major campus fabric IP Clos features:

- Five-Stage IP Clos multi-site fabric with recommended link redundancy:
  - Four redundant core switches acting as a super spine (interleaved mesh to spine layer).
  - Pod1/Building1:
    - Two redundant distribution switches acting as spine.
    - One 4-member Virtual Chassis access switch acting as leaf.
    - One standalone access switch acting as Leaf.
  - Pod2/Building2:
    - Two redundant distribution switches acting as spine.

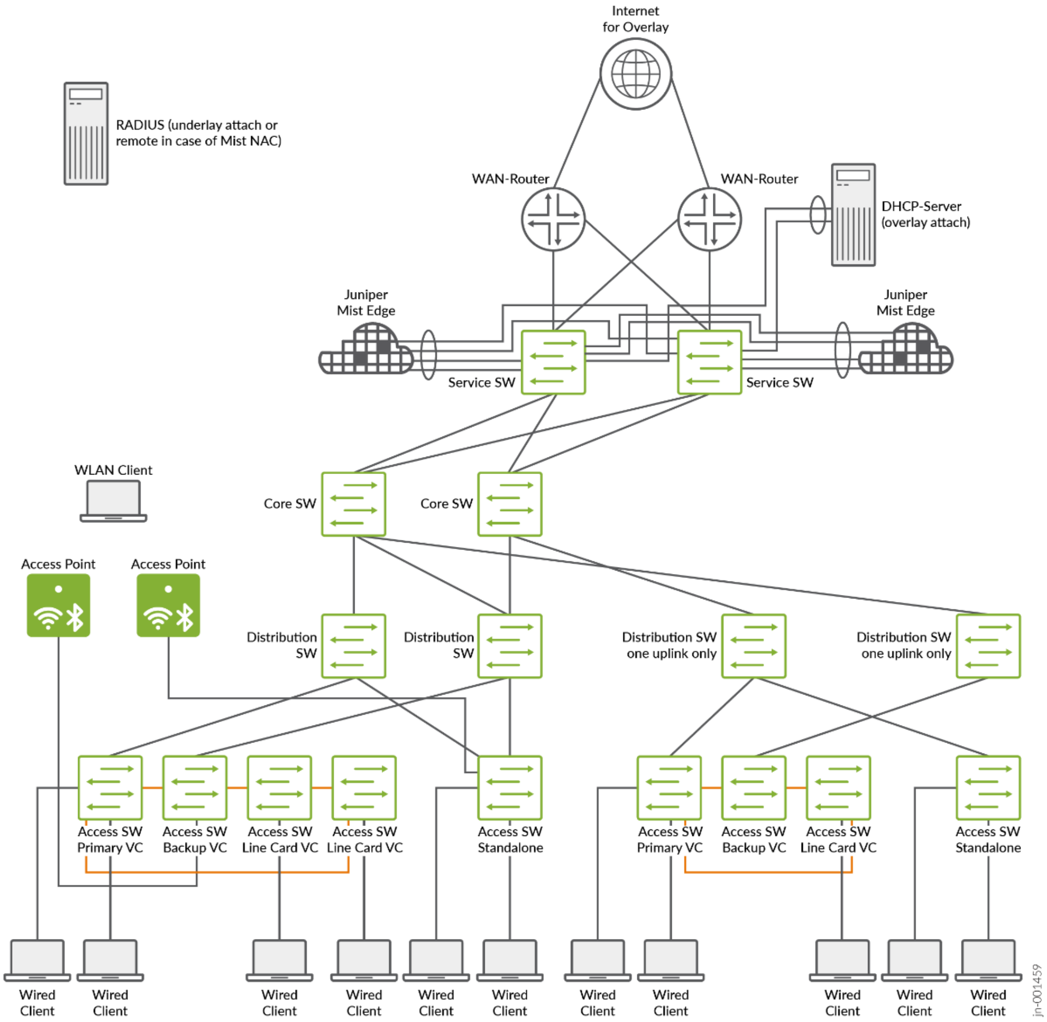
        - One 3-member Virtual Chassis access switch acting as leaf.
        - One standalone access switch acting as leaf.
- Service block function via:
    - Integrated to existing core switches (default) acting as service leaf and core at the same time.
    - As a separate and dedicated pair of service switches acting as service leaf.
    - Attached WAN routers via L2 or L3 exit.
    - Attached servers via ESI-LAG redundant links.
    - Attached Juniper Mist Edges
- WAN router integration
    - L2 fabric exit
        - ESI-LAG based trunks.
    - L3 fabric exit
        - OSPF as routing protocol.
        - eBGP as routing protocol (used for this JVD)
    - Attached to
        - Dedicated service block switches.
    - Redundant WAN router design
        - Two Juniper SRX firewalls in a cluster configuration (or MX routers).
- Wi-Fi access points
    - Locally attached to the access switches with Power over Ethernet (PoE).
    - Various Wi-Fi clients.
    - Basic Wi-Fi roaming.
- Juniper Mist Edge
    - Each Juniper Mist Edge is only attached (through a standard LAG) to one service block switch.
    - Ability to tunnel traffic to Juniper Mist Edge to break out at the service block switch.
    - Only one Juniper Mist Edge at a time shall terminate the tunnels to reduce MAC moves.
    - Fast roaming when using Juniper Mist Edge.

- Overlay server attached to the service block functionality.
  - DHCP server.
  - Two Juniper Mist Edge devices.
  - Other services.
- RADIUS server
  - Server location
    - Local server attached to the underlay network.
    - Remote Juniper Mist Access Assurance via public cloud.
  - Authentication for the following clients
    - Wired clients attached to access switches.
    - Wi-Fi clients using the access points.
  - Authentication based on clients:
    - MAC address.
    - 802.1X EAP authentication.
  - Dynamic authorization profiles
    - Single VLAN assign.
    - Multiple VLANs assigned.
    - GBP tags assigned using Juniper-Switching-Filter.
- Testing Fabric Features such as:
  - DHCP relay
  - Protect RE filter
  - DHCP snooping
  - Storm control
  - MAC address limit with aging
  - DNS
  - NTP
  - IPv6 overlay

- IPv6 underlay
- GBP tags assigned using static configuration

A second topology was created and used to evaluate a design with multiple PoDs with 2 core switches as shown in Figure 13 on page 22, when the full redundancy of recommended links cannot be achieved. This design provides a reduced amount of links between floors and buildings for cost reasons.

Figure 13: Topology 2 JVD Lab Design Phase3



Using the second topology, you can evaluate the following campus fabric IP Clos features:

- Five-Stage IP Clos multisite fabric without recommended link redundancy:
  - Two redundant core switches acting as super spines.

- PoD1/Building1 (recommended link redundancy):
  - Two redundant distribution switches acting as spines.
  - One 4-member Virtual Chassis access switch acting as a leaf.
  - One standalone access switch acting as a leaf.
- PoD2/Building2 (without link redundancy):
  - Two distribution switches acting as spines (each with only one uplink).
  - One 3-member Virtual Chassis access switch acting as a leaf.
  - One standalone access switch acting as a leaf (with only one uplink).

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

In the appendix section of this JVD, we share information on exactly how some of the tests were performed. Contact your Juniper account representative to obtain the full details 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 in mind. Please also consult the separate Test Report for more information. The testing of this Phase3 was executed with a focus on the following:

- Testing with Junos OS version 24.2R2.
- Testing with multiple PoDs and 4 cores in Topology 1 as in Figure 12 on page 19.
- Testing with multiple PoDs and 2 cores with reduced redundancy requirements between core and distribution switches in Topology 2 as shown in Figure 13 on page 22.
- Testing with IPv6 as underlay.
- Testing with IPv6 as overlay.
- Testing with Juniper Mist Edge integration for Wi-Fi scale.

The scale testing for this design was done with:

- Up to 20 virtual routing and forwarding instances (VRFs).
- Up to 500 VLANs (across all VRFs).
- Review the Test Report Brief for details on the verified IPv4 and IPv6 client scaling numbers, which vary based on the access switch models used.

Test Non-Goals

Following are the non-goals for the current Phase3 JVD qualification:

- Testing this fabric with redundant WAN routers. This is already described in a separate JVD extension in common for all fabrics.
- Extensive testing of DHCP relay. This is already done in a separate JVD extension in common for all fabrics.
- VXLAN-GBP testing was done in a separate JVD extension for this fabric with the same Junos version.
- Large scale node testing. This is a lab limitation.
- 3-Stage IP Clos testing. The focus is on 5-Stage IP Clos fabrics with PoDs for scale out.
- 5-Stage IP Clos single site testing. This was already done in Phase1.

Recommendations

The following simple guidelines will help you to successfully implement a campus fabric IP Clos design in your network.

- Review the JVD extension for WAN router integration.
    - For this fabric type, we recommend using the L3 eBGP integration approach.
- All fabric networks should be configured in the following way to avoid inconsistency:
    - First, create them as part of your switch template for a site.
    - Then, import the created networks as part of the campus fabric dialogue and assign to VRFs.
    - Even if the system allows you a local network creation on a switch, do not use this option.
- Do not manually configure VRFs locally on any switch. The fabric usually does this automatically on an as-needed basis.
    - The current exception to this rule is L2 WAN router integration via transport VLAN. Please review the JVD extension for WAN router integration and follow the example in the appendix.
- When using DHCP relay configuration for the fabric:
    - Please review the JVD extension which covers DHCP relay configuration.
    - Configure a "Loopback per-VRF subnet" pool range.
    - Include the pool range as sharing host routes with your WAN router as the loopback IP addresses get assigned as /32 across all of the VRFs shared on the fabric.
    - Only use the fabric dialogue for configuring DHCP relay and no local configuration directly on a switch.
- When designing and using Virtual Chassis:
    - Virtual Chassis can only be used at the access switch layer of a campus fabric environment:
    - When designing a Virtual Chassis, it is not advised to use the maximum number of supported members listed in the Virtual Chassis Overview (Juniper Mist). A good rule of thumb is to use roughly half of the stated maximum. This helps prevent bandwidth oversubscription on the VCPs that form the ring between the chassis members.
    - Create and assign separate templates for Virtual Chassis systems that have the same number of members. Avoid applying identical port configurations to Virtual Chassis setups of different sizes. This approach allows the system to apply configuration changes directly, without repeatedly checking whether the ports defined in the template actually exist on the local Virtual Chassis.

  - All Virtual Chassis configurations should be done through the Juniper Mist cloud and the Modify Virtual Chassis dialogue. Additional CLI or CLI commands should not be used for managing a Virtual Chassis.

- Consider Juniper Mist Edge integration when you have more than 2,000 wireless clients.
  - Each Juniper Mist Edge should connect to both service block functions simultaneously, and this connection should be made through a LAG. On the campus fabric side, a corresponding ESI-LAG will be configured to match it.
  - Design the cluster redundancy so that traffic remains anchored to a single Juniper Mist Edge under normal conditions, switching to another only when a failover is required.
  - Assign VLANs for wireless clients only at service block functions where a Juniper Mist Edge is integrated and do not also stretch or reuse them at the access switches. Further VLAN configuration details for high scale are given in the "Design for high scale of wired and wireless clients with EVPN Fabrics" on page 170 section.
  - Review the "APPENDIX: Mist Edge integration into IP-Clos Fabric (Optional)" on page 165 section for more information about Juniper Mist Edge integration.

- Unassigned access ports should be configured with a quarantine VLAN or disabled ports using a template. Please review the example here.
  - If possible, use a different VRF for the quarantine VLAN to isolate this traffic.
  - Best practice is also enabling “STP Edge” in the quarantine port profile.

- When deciding how to manage port configurations dynamically:
  - Using RADIUS or a NAC system to assign VLANs and filters is the recommended method, particularly for customers using Juniper Mist Access Assurance.
  - Dynamic Port Configuration is considered a less preferred option.

- When using Dynamic Port Configuration:
  - Avoid matching by MAC address if the device supports LLDP.
  - Don’t match by MAC address if ports are enabled with dot1x.
  - The use of a filter-id should be avoided. In most cases, this is unnecessary when ports are 802.1X-enabled and a dynamic VLAN can be assigned through RADIUS.
  - Avoid a high number of port flaps for a DPC-configured port.
  - Refer switch insights to ascertain the individual configuration is applied.

- Traffic towards a third-party RADIUS Server is expected to use inet.0 via the management port, same as the management traffic towards the Juniper Mist cloud, e.g. underlay. This allows you to fine-tune the MTU for the UDP Packets send towards such a service in case it is needed.

When utilizing IPv6 in overlay, it is currently mandatory that you also select IPv6 in Underlay in the EVPN campus fabric creation dialogue. The IPv6 underlay selection in the EVPN campus fabric creation dialogue is supported in the following fabric types: EVPN Multihoming, ERB and IP Clos. Due to a Junos limitation, CRB is an exception to this rule and IPv4 underlay-based EVPN Fabric management is permanently enabled. Also, the usage of dynamic IPv6 assignment using SLACC is not recommended.

Revision History

Table 3: Revision History

Date	Version	Description
November 2025	Phase 3 (this JVD)	Junos OS version 24.2R2 IPv6 support for overlay and underlay Reduced link-redundancy between core and distribution switches Juniper Mist Edge integration EVPN type 2 and type 5 routes coexistence scale testing
May 2024	Phase 2	Junos OS version 22.4R3 Multiple PoDs Feature combination testing EVPN Type 2 scale testing
September 2023	Phase 1	Initial publish Junos OS version 22.2R3-S2.8 Single site 5-Stage Fabric

APPENDIX: Example IP Clos Fabric Creation

IN THIS SECTION

In this chapter, we demonstrate the creation of an IP Clos fabric managed by Juniper Mist cloud. This is a small lab with physical devices as a single site fabric to begin with.

It is possible to build a virtual campus fabric to explore the major functionality without Virtual Chassis and VXLAN-GBP testing capabilities. If you are interested, please review the following Network Configuration Example.

Campus Fabric IP Clos Lab Components

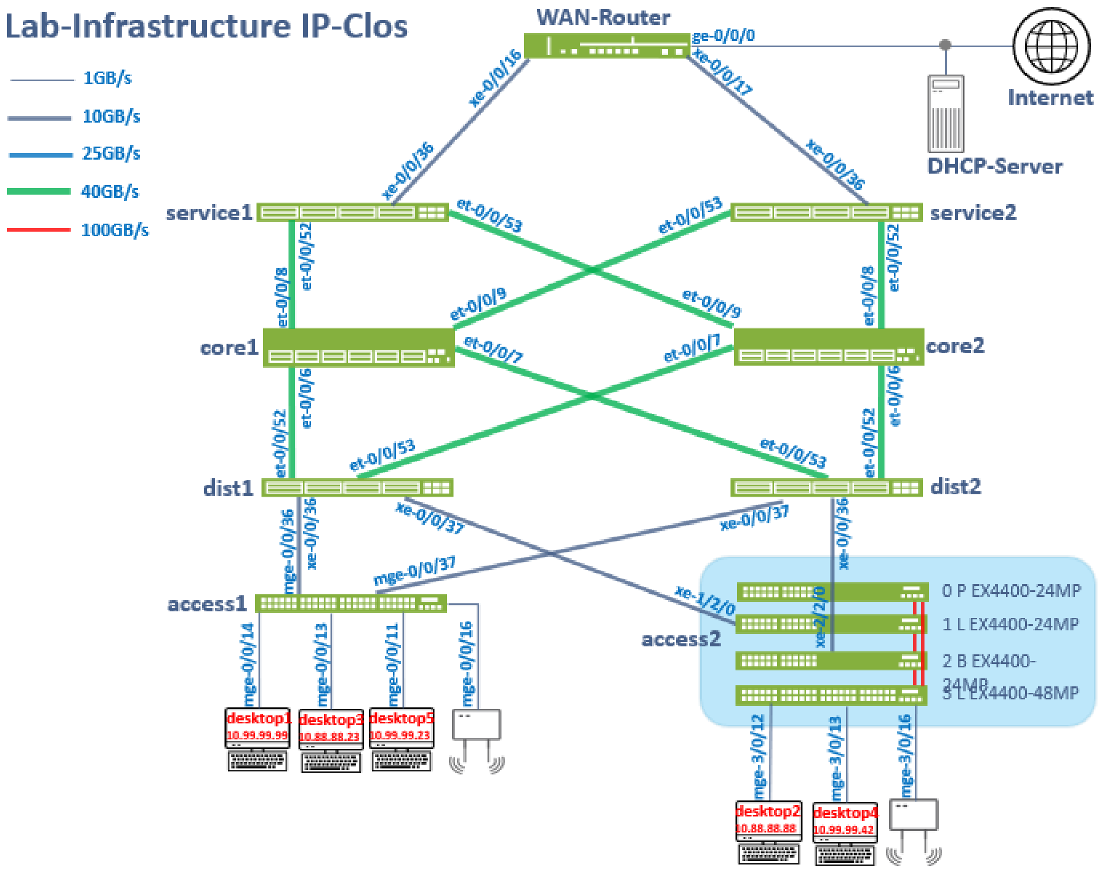
This configuration example uses the following devices:

- Two Juniper Networks® QFX10002 Switches as core devices.
- Two Juniper Networks® QFX5120 Switches as distribution devices.
- One Juniper Networks® EX4400-MP Switch as a standalone access device.
- Four EX4400-MP Switches as Virtual Chassis access devices.
- Two Juniper Networks® QFX5120 Switches as service block devices.
- One Juniper Networks® SRX1500 as a WAN router. Integration to fabric utilizes eBGP.
- A Linux-based DHCP server for the fabric. Externally attached to the fabric.

- Juniper® Series of High-Performance Access Points.
- Various Linux desktops that act as wired/wireless clients.

The following Lab-topology was used:

Figure 14: Lab Topology



> **NOTE**: This picture does not show the required out-of-band management network that each core, distribution, service block and access switch requires to be managed by the Juniper Mist cloud (and for functions like RADIUS authentication). You need to design for that as well.

Juniper Mist Wired Assurance

Juniper Mist Wired Assurance, through the Juniper Mist portal, can be used to centrally manage all Juniper switches. Juniper Mist Wired Assurance gives you full visibility into the devices that comprise your network's access layer. The portal provides a user interface to access your architecture through the AI-driven cloud services with your Juniper Mist account. You can monitor, measure, and get alerts on key compliance metrics on the wired network. This includes switch version and PoE compliance, switch-AP affinity, and VLAN insights.

Juniper switch onboarding to the Juniper Mist cloud: https://www.juniper.net/documentation/us/en/quick-start/hardware/cloud-ready-switches/topics/topic-map/step-1-begin.html

Juniper Mist Wired Assurance, through the portal, is used to build a campus fabric IP Clos from the ground up. This includes the following:

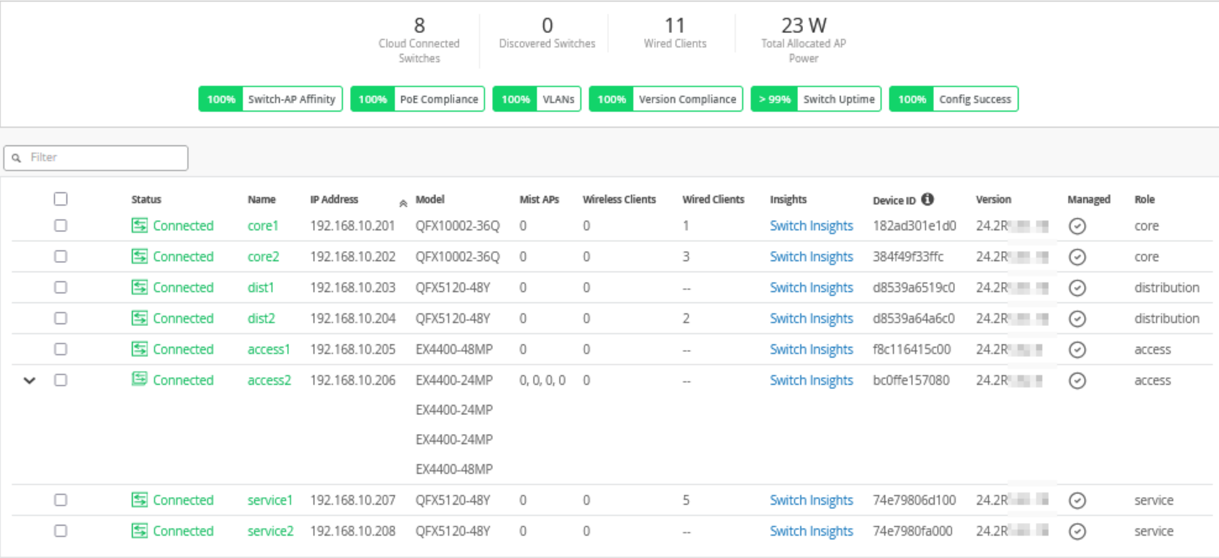
- Assignment of point-to-point (P2P) links between the core, distribution, service block and access layers.
- Assignment of unique BGP AS numbers per device participating in the underlay and overlay.
- The creation of VRF instances allows you to logically segment traffic. This also includes the assignment of new or existing VLANs to each representative VRF.
- IP addressing of each L3 gateway integrated routing and bridging (IRB) interface assigned to the access layer.
- IP addressing of each loopback interface.
- Configuration of routing policies for underlay and overlay connectivity.
- Optimized maximum transmission unit (MTU) settings for P2P underlay, L3 IRB, and ESI-LAG bundles.
- Downloadable connection table (CSV format) that can be used by those involved in the physical buildout of the campus fabric.
- Graphical interface depicting all devices with BGP peering and physical link status.

For more information on Juniper Mist Wired Assurance, see: https://www.mist.com/documentation/category/wired-assurance/

Juniper Mist Wired Assurance Switches

You must validate that each device participating in the campus fabric has been adopted or claimed and assigned to a site. The switches are named for their respective layers in the fabric to facilitate building and operating the fabric.

Figure 15: Switch Inventory



8 Cloud Connected Switches | 0 Discovered Switches | 11 Wired Clients | 23 W Total Allocated AP Power

100% Switch-AP Affinity | 100% PoE Compliance | 100% VLANs | 100% Version Compliance | > 99% Switch Uptime | 100% Config Success

Filter

	☐	Status	Name	IP Address	Model	Mist APs	Wireless Clients	Wired Clients	Insights	Device ID	Version	Managed	Role
	☐	Connected	core1	192.168.10.201	QFX10002-36Q	0	0	1	Switch Insights	182ad301e1d0	24.2R[redacted]	✓	core
	☐	Connected	core2	192.168.10.202	QFX10002-36Q	0	0	3	Switch Insights	384f49f33ffc	24.2R[redacted]	✓	core
	☐	Connected	dist1	192.168.10.203	QFX5120-48Y	0	0	--	Switch Insights	d8539a6519c0	24.2R[redacted]	✓	distribution
	☐	Connected	dist2	192.168.10.204	QFX5120-48Y	0	0	2	Switch Insights	d8539a64a6c0	24.2R[redacted]	✓	distribution
	☐	Connected	access1	192.168.10.205	EX4400-48MP	0	0	--	Switch Insights	f8c116415c00	24.2R[redacted]	✓	access
⌄	☐	Connected	access2	192.168.10.206	EX4400-24MP EX4400-24MP EX4400-24MP EX4400-48MP	0, 0, 0, 0	0	--	Switch Insights	bc0ffe157080	24.2R[redacted]	✓	access
	☐	Connected	service1	192.168.10.207	QFX5120-48Y	0	0	5	Switch Insights	74e79806d100	24.2R[redacted]	✓	service
	☐	Connected	service2	192.168.10.208	QFX5120-48Y	0	0	--	Switch Insights	74e7980fa000	24.2R[redacted]	✓	service

Templates

A key feature of switch management through the Juniper Mist cloud is to use templates and a hierarchical model to group the switches and make bulk updates. Templates provide uniformity and convenience, while the hierarchy (site and switch) provides both scale and granularity.

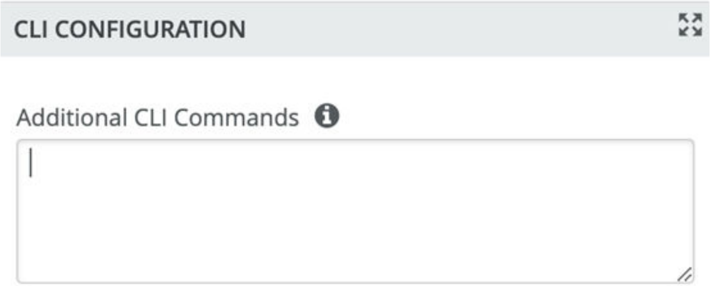
Templates and the hierarchical model means that you can create a template configuration and then all the devices in each group inherit the template settings. When a conflict occurs, for example, when there are settings at both the site and organizational levels that apply to the same device, the narrower settings (in this case, the site settings) override the broader settings defined at the organization level.

Individual switches, at the bottom of the hierarchy, can inherit all or part of the configuration defined at the organization level, and again at the site level. Of course, individual switches can also have their own unique configurations.

You can include individual CLI commands at any level of the hierarchy, which are then appended to all the switches in that group on an "AND" basis—that is, individual CLI settings are appended to the existing configuration (existing settings might be replaced or appended).

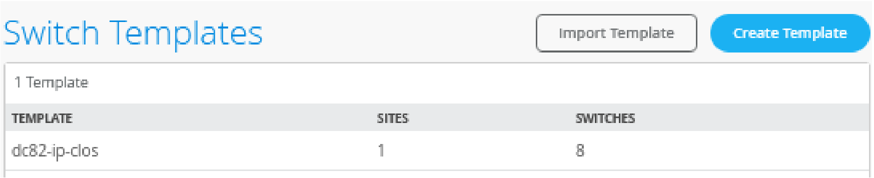
NOTE: If you run CLI commands for items not native to the portal, this configuration data is applied last; overwriting existing configuration data within the same stanza. You can access the CLI command option from the switch template or individual switch configuration.

Figure 16: Adding additional CLI



Under organization and switch templates, we use the following template:

Figure 17: Switch Templates



1 Template

TEMPLATE	SITES	SWITCHES
dc82-ip-clos	1	8

We provide a copy of the following template in JSON format for importing into your own system for verification:

```
{
  "additional_config_cmds": [],
  "networks": {
    "VLAN1033": {
      "vlan_id": "1033",
      "subnet": "10.33.33.0/24",
      "subnet6": ""
    },
    "VLAN1088": {
      "vlan_id": "1088",
      "subnet": "10.88.88.0/24",
      "subnet6": ""
    },
    "VLAN1099": {
      "vlan_id": "1099",
      "subnet": "10.99.99.0/24",
      "subnet6": ""
    }
  },
  "port_usages": {
    "vlan1099-no-auth": {
      "mode": "access",
      "disabled": false,
      "port_network": "VLAN1099",
      "voip_network": null,
      "stp_edge": false,
      "port_auth": null,
      "allow_multiple_supplicants": null,
      "enable_mac_auth": null,
      "mac_auth_only": null,
      "guest_network": null,
      "bypass_auth_when_server_down": null,
      "dynamic_vlan_networks": null,
      "stp_p2p": false,
      "stp_no_root_port": false,
      "mac_auth_protocol": null,
      "reauth_interval": null,
      "all_networks": false,
      "networks": null,
      "speed": "auto",
      "duplex": "auto",
      "mac_limit": 0,
      "persist_mac": false,
      "poe_disabled": false,
      "enable_qos": false,
      "storm_control": {},
      "mtu": null,
      "description": "",
      "disable_autoneg": false
    },
    "vlan1088-no-auth": {
      "disabled": false,
      "mode": "access",
      "port_network": "VLAN1088",
      "voip_network": null,
      "stp_edge": false,
      "port_auth": null,
      "allow_multiple_supplicants": null,
      "enable_mac_auth": null,
      "mac_auth_only": null,
      "guest_network": null,
      "bypass_auth_when_server_down": null,
      "dynamic_vlan_networks": null,
      "stp_p2p": false,
      "stp_no_root_port": false,
      "mac_auth_protocol": null,
      "reauth_interval": null,
      "all_networks": false,
      "networks": null,
      "speed": "auto",
      "duplex": "auto",
      "mac_limit": 0,
      "persist_mac": false,
      "poe_disabled": false,
      "enable_qos": false,
      "storm_control": {},
      "mtu": null,
      "description": "",
      "disable_autoneg": false
    },
    "vlan1099-eap-auth": {
      "mode": "access",
      "disabled": false,
      "port_network": "VLAN1099",
      "voip_network": null,
      "stp_edge": false,
      "port_auth": "dot1x",
      "allow_multiple_supplicants": false,
      "enable_mac_auth": false,
      "mac_auth_only": false,
      "guest_network": null,
      "bypass_auth_when_server_down": false,
      "dynamic_vlan_networks": null,
      "stp_p2p": false,
      "stp_no_root_port": false,
      "mac_auth_protocol": null,
      "reauth_interval": "65000",
      "all_networks": false,
      "networks": null,
      "speed": "auto",
      "duplex": "auto",
      "mac_limit": 0,
      "persist_mac": false,
      "poe_disabled": false,
      "enable_qos": false,
      "storm_control": {},
      "mtu": null,
      "description": "",
      "disable_autoneg": false
    },
    "dynamic": {
      "mode": "dynamic",
      "rules": []
    },
    "vlan1088-eap-auth": {
      "disabled": false,
      "mode": "access",
      "port_network": "VLAN1088",
      "voip_network": null,
      "stp_edge": false,
      "port_auth": "dot1x",
      "allow_multiple_supplicants": false,
      "enable_mac_auth": false,
      "mac_auth_only": false,
      "guest_network": null,
      "bypass_auth_when_server_down": false,
      "dynamic_vlan_networks": null,
      "stp_p2p": false,
      "stp_no_root_port": false,
      "mac_auth_protocol": null,
      "reauth_interval": "65000",
      "all_networks": false,
      "networks": null,
      "speed": "auto",
      "duplex": "auto",
      "mac_limit": 0,
      "persist_mac": false,
      "poe_disabled": false,
      "enable_qos": false,
      "storm_control": {},
      "mtu": null,
      "description": "",
      "disable_autoneg": false
    },
    "access-point": {
      "mode": "trunk",
      "disabled": false,
      "port_network": "VLAN1033",
      "voip_network": null,
      "stp_edge": false,
      "port_auth": null,
      "allow_multiple_supplicants": null,
      "enable_mac_auth": null,
      "mac_auth_only": null,
      "guest_network": null,
      "bypass_auth_when_server_down": null,
      "dynamic_vlan_networks": null,
      "stp_p2p": false,
      "stp_no_root_port": false,
      "mac_auth_protocol": null,
      "reauth_interval": null,
      "all_networks": true,
      "networks": [],
      "speed": "auto",
      "duplex": "auto",
      "mac_limit": 0,
      "persist_mac": false,
      "poe_disabled": false,
      "enable_qos": false,
      "storm_control": {},
      "mtu": null,
      "description": "",
      "disable_autoneg": false
    },
    "vlan1099-mab-auth": {
      "mode": "access",
      "disabled": false,
      "port_network": "VLAN1099",
      "voip_network": null,
      "stp_edge": false,
      "port_auth": "dot1x",
      "allow_multiple_supplicants": true,
      "enable_mac_auth": true,
      "mac_auth_only": true,
      "guest_network": null,
      "bypass_auth_when_server_down": false,
      "dynamic_vlan_networks": null,
      "stp_p2p": false,
      "stp_no_root_port": false,
      "mac_auth_protocol": "pap",
      "reauth_interval": "65000",
      "all_networks": false,
      "networks": null,
      "speed": "auto",
      "duplex": "auto",
      "mac_limit": 0,
      "persist_mac": false,
      "poe_disabled": false,
      "enable_qos": false,
      "storm_control": {},
      "mtu": null,
      "description": "",
      "disable_autoneg": false
    },
    "vlan1088-mab-auth": {
      "disabled": false,
      "mode": "access",
      "port_network": "VLAN1088",
      "voip_network": null,
      "stp_edge": false,
      "port_auth": "dot1x",
      "allow_multiple_supplicants": true,
      "enable_mac_auth": true,
      "mac_auth_only": true,
      "guest_network": null,
      "bypass_auth_when_server_down": false,
      "dynamic_vlan_networks": null,
      "stp_p2p": false,
      "stp_no_root_port": false,
      "mac_auth_protocol": "pap",
      "reauth_interval": "65000",
      "all_networks": false,
      "networks": null,
      "speed": "auto",
      "duplex": "auto",
      "mac_limit": 0,
      "persist_mac": false,
      "poe_disabled": false,
      "enable_qos": false,
      "storm_control": {},
      "mtu": null,
      "description": "",
      "disable_autoneg": false
    }
```

```
  },
  "disabled_system_defined_port_usages": [],
  "extra_routes": {},
  "extra_routes6": {},
  "switch_mgmt": {
    "config_revert_timer": 10,
    "root_password": "",
    "local_accounts": {},
    "protect_re": {
      "enabled": false
    },
    "tacacs": {
      "enabled": false
    }
  },
  "mist_nac": {
    "enabled": true,
    "network": null
  },
  "radius_config": {
    "auth_servers": [],
    "acct_servers": [],
    "auth_servers_timeout": 5,
    "auth_servers_retries": 3,
    "fast_dot1x_timers": false,
    "acct_interim_interval": 0,
    "auth_server_selection": "ordered",
    "coa_enabled": false,
    "coa_port": ""
  },
  "vrf_config": {
    "enabled": false
  },
  "remote_syslog": {
    "enabled": false
  },
  "snmp_config": {
    "enabled": false
  },
  "dhcp_snooping": {
    "enabled": false
  },
  "bgp_config": null,
  "routing_policies": {},
  "dns_servers": [
    "8.8.8.8",
    "9.9.9.9"
  ],
  "dns_suffix": [],
  "ntp_servers": [
    "192.168.10.1"
  ],
  "acl_policies": [],
  "port_mirroring": {},
  "switch_matching": {
    "enable": true,
    "rules": []
  },
  "name": "dc82-ip-clos"
}
```

Topology

Juniper Mist Wired Assurance provides the configuration for LAN and loopback IP addressing for each core, distribution, service block and access device once the device's management IP address is reachable. Each device is provisioned with a /32 loopback address and /31 point-to-point interfaces that interconnect the switch devices within the campus fabric IP-Clos down to the access layer as well in opposite to other designs.

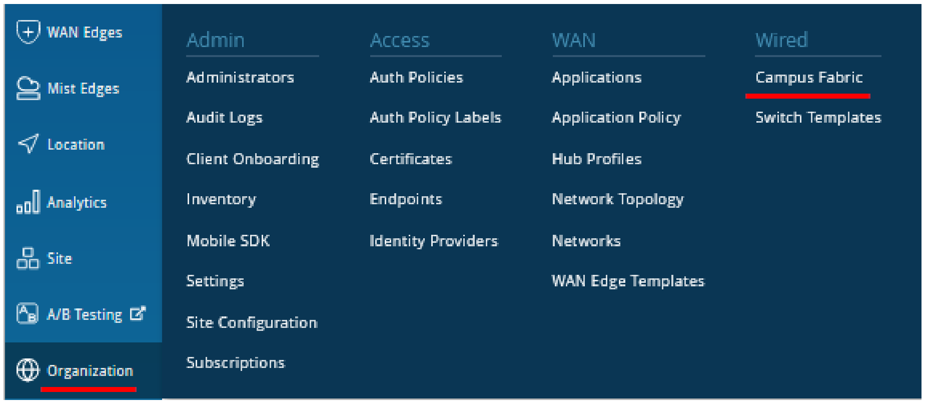
The WAN router can be provisioned through the portal but is separate from the campus fabric workflow. The WAN router has an eBGP peering enabled and exchanges routes with the service block switches of the fabric. WAN routers can be standalone or built as a high availability cluster. In this document, a single SRX Firewall is used as the WAN router.

NOTE: There is a JVD extension available covering more details on WAN router integration especially for production grade installations. What is shown here does not have the required redundancy for production as it is just a single WAN router.

Create the Campus Fabric

1. From **Organization** on the left-hand side of the portal, select **Campus Fabric**.

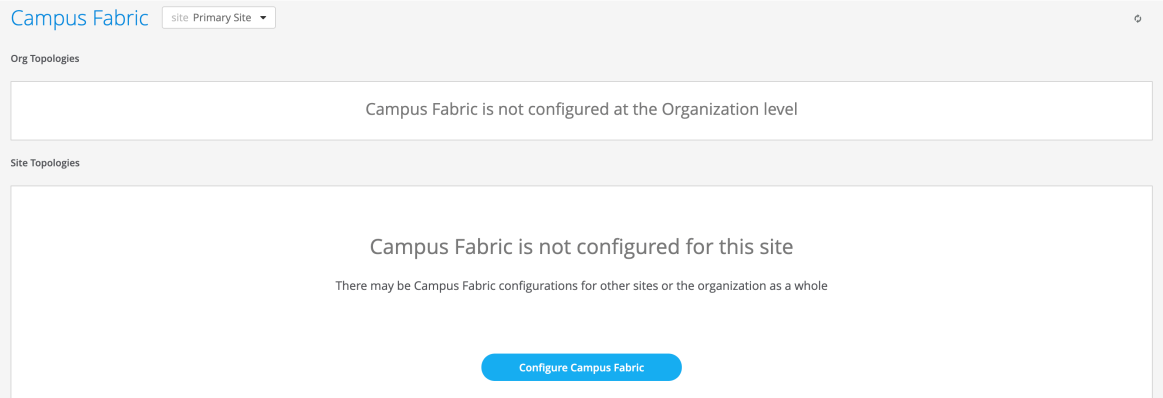
Figure 18: Campus Fabric Creation



Juniper Mist provides the option of deploying a campus fabric at the organizational or site level noted in the upper-left corner of the campus fabric menu shown below. Both designs now allow you to build fabrics with just a single PoD or multiple PoDs based on customer requirements to connect multiple buildings.

In our example here, the fabric was built on the site level with a single PoD only.

Figure 19: Fabric Site Level Creation



Campus Fabric is not configured at the Organization level

Campus Fabric is not configured for this site

There may be Campus Fabric configurations for other sites or the organization as a whole

Choose the Campus Fabric Topology

2. Select the **Campus Fabric IP Clos** option below and configure the following:

   a. Topology Type=`Campus Fabric IP Clos`

b. Topology Name=`ip-clos`

c. BGP Local AS=`65001` (this is the default setting)

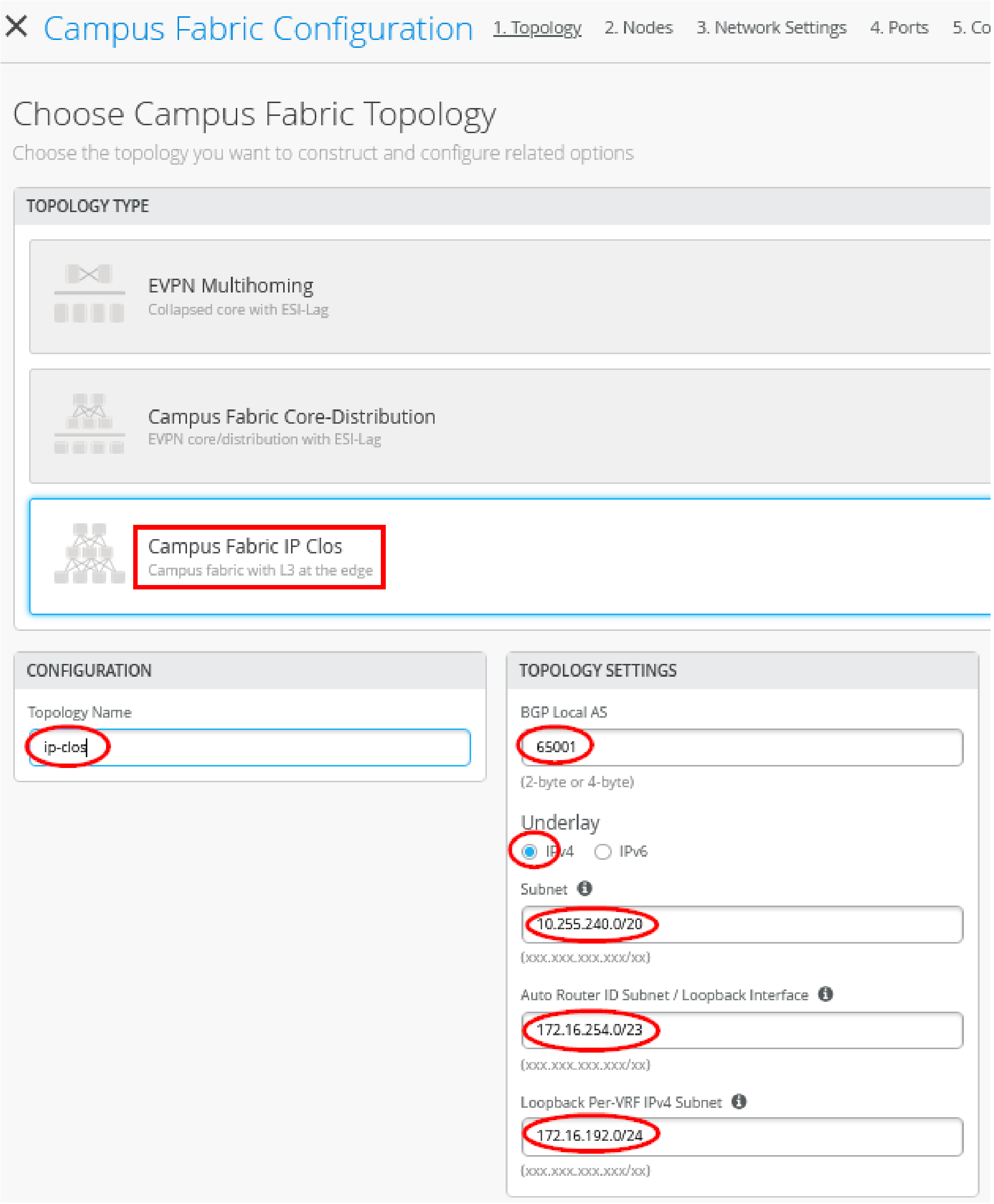
d. Underlay=`IPv4` (this is the default setting)

e. Subnet=`10.255.240.0/20` (this is the default setting)

f. Auto Route ID Subnet / Loopback Interface=`172.16.254.0/23` (this is the default setting)

g. Loopback per VRF subnet=`172.16.192.0/24` (this is the default setting)

Figure 20: IP Clos Fabric Creation



Choose Campus Fabric Topology

Choose the topology you want to construct and configure related options

TOPOLOGY TYPE

EVPN Multihoming
Collapsed core with ESI-Lag

Campus Fabric Core-Distribution
EVPN core/distribution with ESI-Lag

Campus Fabric IP Clos
Campus fabric with L3 at the edge

CONFIGURATION

Topology Name: ip-clos

TOPOLOGY SETTINGS

BGP Local AS: 65001
(2-byte or 4-byte)

Underlay: ● IPv4 ○ IPv6

Subnet: 10.255.240.0/20
(xxx.xxx.xxx.xxx/xx)

Auto Router ID Subnet / Loopback Interface: 172.16.254.0/23
(xxx.xxx.xxx.xxx/xx)

Loopback Per-VRF IPv4 Subnet: 172.16.192.0/24
(xxx.xxx.xxx.xxx/xx)

Topology Settings

- BGP Local AS—Represents the starting point of private BGP AS numbers that are automatically allocated per device. You can use whatever private BGP AS number range suits your deployment, routing policy is provisioned by Juniper Mist to ensure the AS numbers are never advertised outside of the fabric.

- Subnet—Represents the pool of IP addresses used for point-to-point links between devices. You can use whatever range suits your deployment. Juniper Mist breaks this subnet into /31 subnet addressing per link. This number can be modified to suit the specific deployment scale. For example, /24 provides up to 128 P2P /31 subnets.

- Auto Router ID Subnet—Represents the pool of IP addresses associated with each device's loopback address. Each device will automatically get a loopback IP address of /32 assigned from this pool. You can use whatever range suits your deployment. VXLAN tunnelling using a VTEP is associated with this address. The loopback IP addresses assigned here are only visible in the underlay transport network. The definition of these underlay loopback IP addresses is critical for the operation of the EVPN-VXLAN fabric to function at all.
- Loopback per-VRF-subnet—Represents a second pool of loopback IP addresses which are each associated with an L3 VRF and switch of the overlay fabric network. It is designed for scale-out services in the overlay network where some services, like DHCP relay, share a single IP address external to the fabric. This is the case for anycast fabrics like ERB and IP Clos. If those L3 VRFs use a dedicated loopback IP address per VRF and switch, it is easy to send return answers to an originating VRF/switch.

NOTE: We recommend default settings for all options unless they conflict with other networks attached to the campus fabric. The P2P links between each layer utilize /31 addressing to conserve addresses.

Select Campus Fabric Nodes

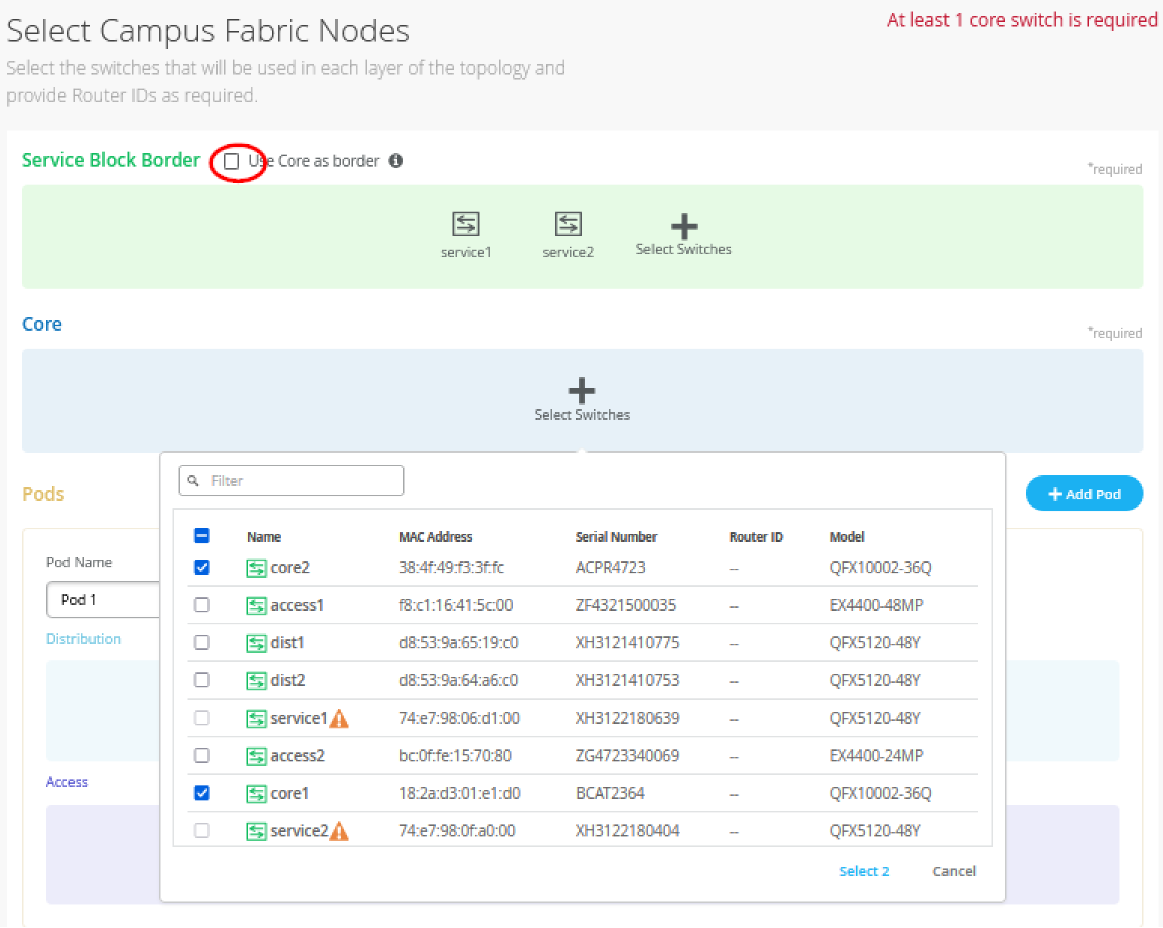
3. Select devices to participate in each layer of the campus fabric IP Clos. We recommend that you validate each device's presence in the site switch inventory prior to the creation of the campus fabric.

   The next step is to assign the switches to the layers. Since the switches were named relative to target layer functionality, they can be quickly assigned to their roles.

   The service block router is where the campus fabric interconnects external devices such as firewalls, routers, or critical devices. For example, DHCP and RADIUS servers. Devices to which external services connect to the campus fabric are known as border leafs. If you want to connect these services or devices to the campus fabric.

   IP Clos in a separate device or pair of devices, clear the **Use Core as border** option and select the **Select Switches** option to choose the devices as this is the case for our topology.

Figure 21: Service Block Switch Configuration



Select Campus Fabric Nodes

Select the switches that will be used in each layer of the topology and provide Router IDs as required.

At least 1 core switch is required

Service Block Border □ Use Core as border *required

Core *required

Pods

Pod Name: Pod 1

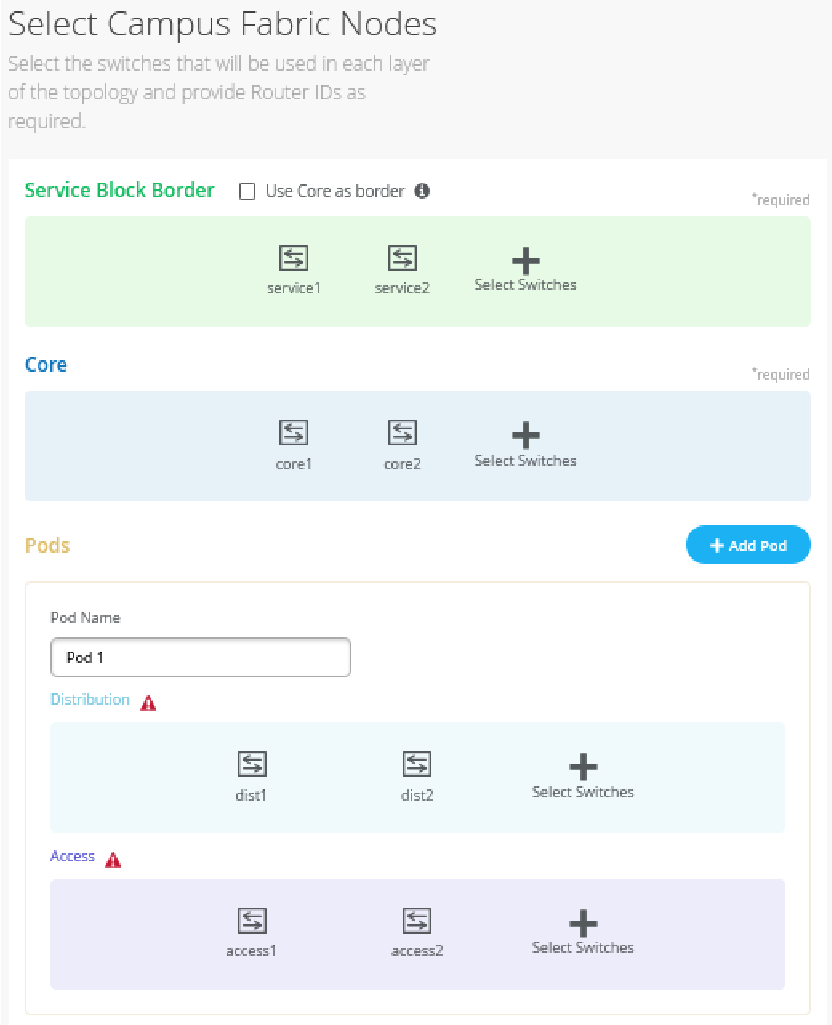
Filter

▣	Name	MAC Address	Serial Number	Router ID	Model
☑	core2	38:4f:49:f3:3f:fc	ACPR4723	--	QFX10002-36Q
□	access1	f8:c1:16:41:5c:00	ZF4321500035	--	EX4400-48MP
□	dist1	d8:53:9a:65:19:c0	XH3121410775	--	QFX5120-48Y
□	dist2	d8:53:9a:64:a6:c0	XH3121410753	--	QFX5120-48Y
□	service1 ⚠	74:e7:98:06:d1:00	XH3122180639	--	QFX5120-48Y
□	access2	bc:0f:fe:15:70:80	ZG4723340069	--	EX4400-24MP
☑	core1	18:2a:d3:01:e1:d0	BCAT2364	--	QFX10002-36Q
□	service2 ⚠	74:e7:98:0f:a0:00	XH3122180404	--	QFX5120-48Y

NOTE: Placing the service block router on a dedicated pair of switches (or single switch) alleviates the encapsulation and de-capsulation of VXLAN headers from the core layer. If you want to combine this capability within the core devices, you must select the **Use Core as border** option.

The final configuration of all nodes of the fabric at their topology should look like the figure below.

Figure 22: Fabric Configuration of All Nodes



Select Campus Fabric Nodes

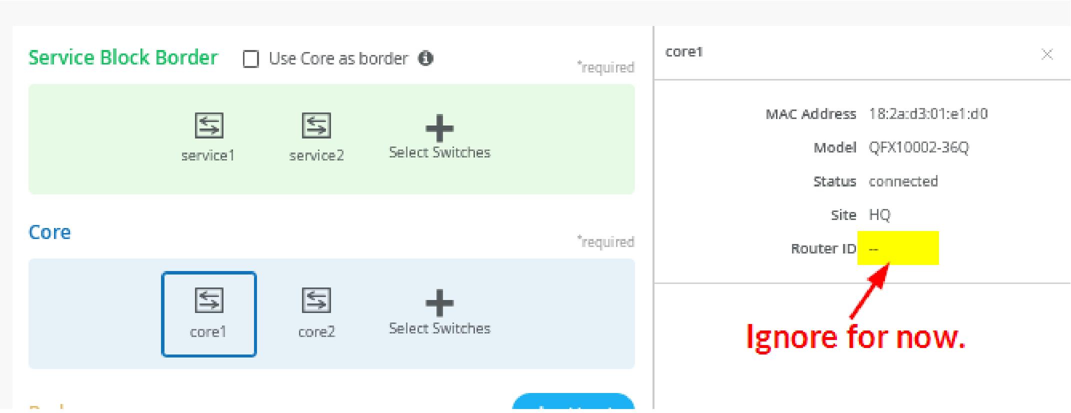
Select the switches that will be used in each layer of the topology and provide Router IDs as required.

Service Block Border ☐ Use Core as border

Pod Name: Pod 1

4. Once all layers have selected the appropriate devices, you must provide an underlay loopback IP address for each device. This loopback interface is associated with a logical construct called a VTEP; used to source the VXLAN tunnel. The campus fabric IP Clos has VTEPs for VXLAN tunnelling on all nodes of the fabric. When defining an auto router ID subnet prefix, the underlay loopback IP address and router ID assignments happen automatically. There is no way to manually assign them as it was possible in older versions of previous documentation.

Figure 23: Fabric Node has No Initial Router ID Assigned Yet.



core1

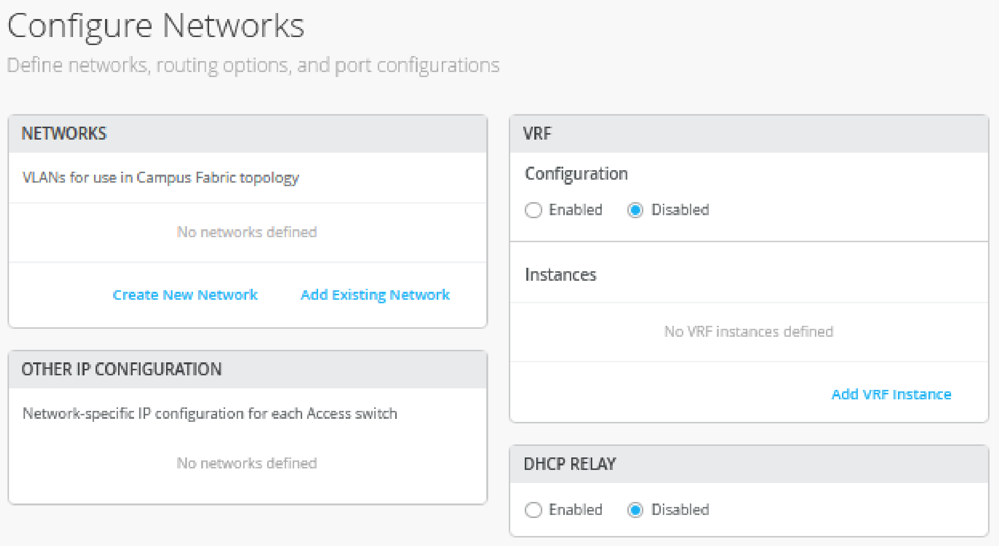
MAC Address 18:2a:d3:01:e1:d0
Model QFX10002-36Q
Status connected
Site HQ
Router ID --

Ignore for now.

Configuring Networks

5. Enter the network information such as VLANs and VRF options. VLANs are mapped to VNIs and can optionally be mapped to VRFs to provide a way to logically separate traffic such as device traffic from corporate traffic.

Figure 24: Configure Networks



Configure Networks

Define networks, routing options, and port configurations

NETWORKS

VLANs for use in Campus Fabric topology

No networks defined

Create New Network   Add Existing Network

OTHER IP CONFIGURATION

Network-specific IP configuration for each Access switch

No networks defined

VRF

Configuration

○ Enabled ● Disabled

Instances

No VRF instances defined

Add VRF Instance

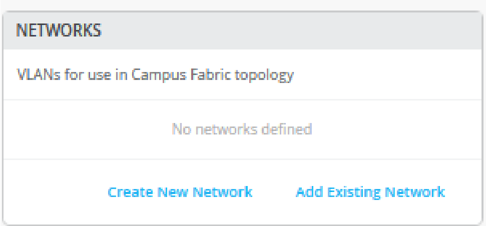
DHCP RELAY

○ Enabled ● Disabled

Networks

6. VLANs can be created or imported under this section including the IP subnet and default gateway per each VLAN. The Shared Elements section of the campus fabric template includes the networks section mentioned above where VLANs are created.

Figure 25: Default Known Networks



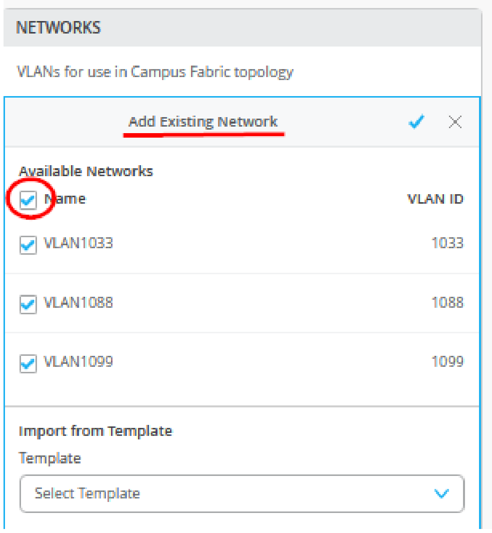
NETWORKS

VLANs for use in Campus Fabric topology

No networks defined

7. Back to the campus fabric build, select **Add Existing Network** using the existing template which includes L2 VLAN information. All VLAN and IP information is inherited from the template.

Figure 26: Network Import from Template



NETWORKS

VLANs for use in Campus Fabric topology

Available Networks

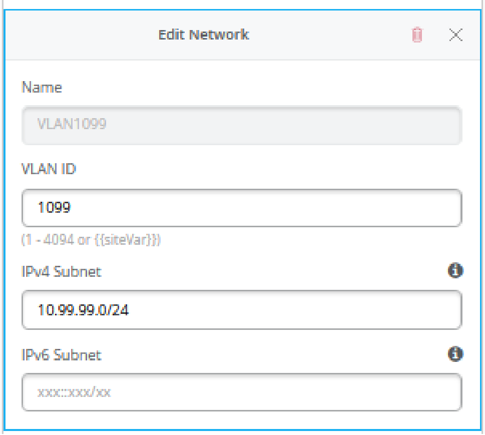
Name	VLAN ID
☑ VLAN1033	1033
☑ VLAN1088	1088
☑ VLAN1099	1099

Import from Template

Template: Select Template

Networks can be edited, newly added or added from an existing template:

Figure 27: Edit a Network



Edit Network

Name: VLAN1099

VLAN ID: 1099

(1 - 4094 or {{siteVar}})

IPv4 Subnet: 10.99.99.0/24

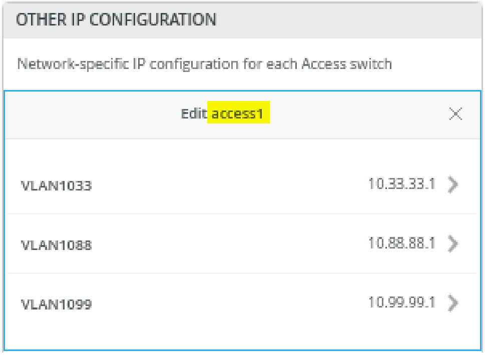
IPv6 Subnet: xxx::xxx/xx

Other IP Configuration

Juniper Mist Wired Assurance provides automatic IP addressing for IRB interfaces for each of the VLANs. Port profiles and port configurations then associate the VLAN with specified ports. In this case, we selected campus fabric IP Clos at the onset of the campus fabric build. This fabric type uses anycast addressing for all devices participating in the L3 subnet. In this case, all access switches are configured with the same IP address for each L3 subnet.

More information on anycast gateways can be found here: https://www.juniper.net/documentation/us/en/software/junos/evpn-vxlan/topics/concept/evpn-mclag-irb-gateway-anycast-address.html

Figure 28: Anycast GW on Access Switch 1



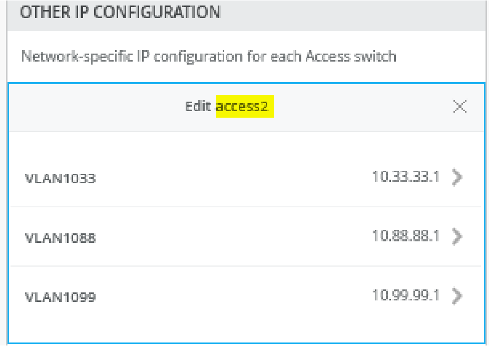
OTHER IP CONFIGURATION

Network-specific IP configuration for each Access switch

Edit access1

VLAN1033	10.33.33.1
VLAN1088	10.88.88.1
VLAN1099	10.99.99.1

Figure 29: Anycast GW on Access2 VC Switch



OTHER IP CONFIGURATION

Network-specific IP configuration for each Access switch

Edit access2

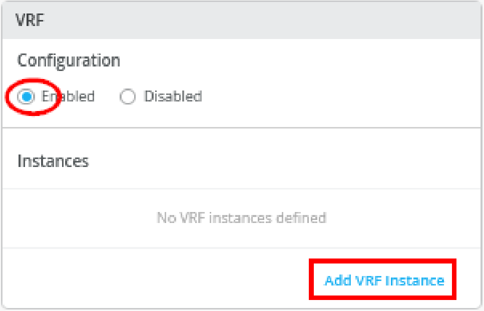
VLAN1033	10.33.33.1
VLAN1088	10.88.88.1
VLAN1099	10.99.99.1

By default, all VLANs are placed in the default VRF. The VRF option allows you to group common VLANs into the same VRF or separate VRFs depending on traffic isolation requirements. This example includes three VRFs or routing instances: corp-it, developers, and guest-wifi.

NOTE: In our example, we use a VLAN per VRF as that is the minimum requirement. You can always add multiple VLANs to a particular VRF but by default the traffic between these VLANs is always allowed and not controlled by firewalls. The IP Clos fabric type when using EX4100 or EX4400 Switches as access switches does allow more granular control of the traffic inside and between VLANs at the same VRF by using VXLAN-GBP. In this case, it’s better to design for a single, global VRF of the fabric where all VLANs are located.

8. Here, you build the first corp-it VRF and select the pre-defined vlan 1099.

Figure 30: Enable VRF



VRF

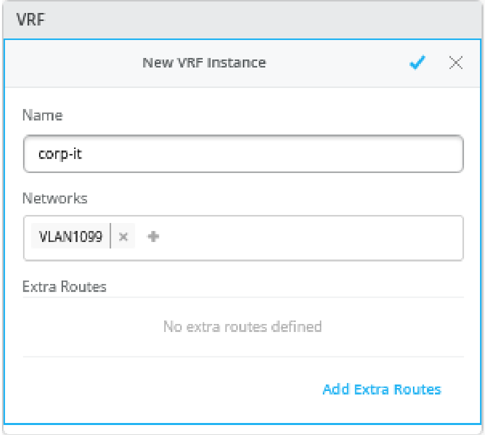
Configuration

● Enabled ○ Disabled

Instances

No VRF instances defined

Figure 31: Assign First Network to VRF1



VRF

New VRF Instance

Name: corp-it

Networks: VLAN1099

Extra Routes

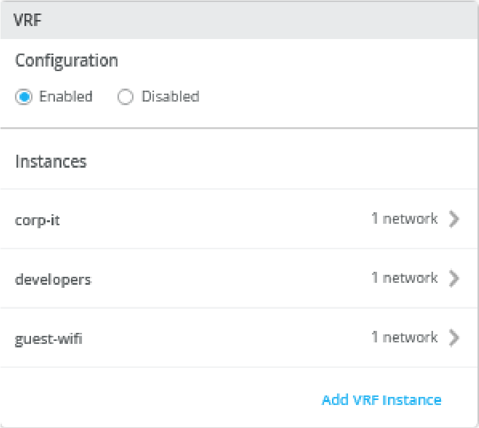
No extra routes defined

9. Create two additional VRFs:

   a. VRF2=`developers` with Network=`VLAN1088`

   b. VRF3=`guest-wifi` with Network=`VLAN1033`

Figure 32: 3 VRFs with At Least One Network Each



VRF

Configuration

● Enabled ○ Disabled

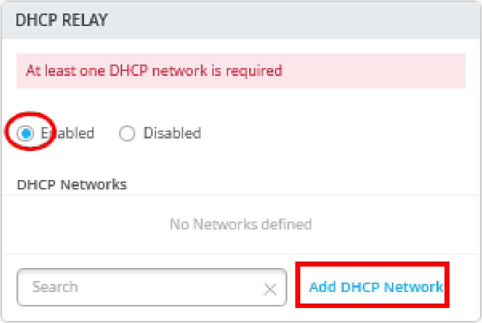
Instances

corp-it	1 network
developers	1 network
guest-wifi	1 network

10. Assign to each VLAN our external DHCP server using DHCP relay.

> **NOTE**: Information about DHCP relay usage and proper design with campus fabrics is available via the following JVD extension.

Figure 33: Enable DHCP Relay



DHCP RELAY

At least one DHCP network is required

● Enabled ○ Disabled

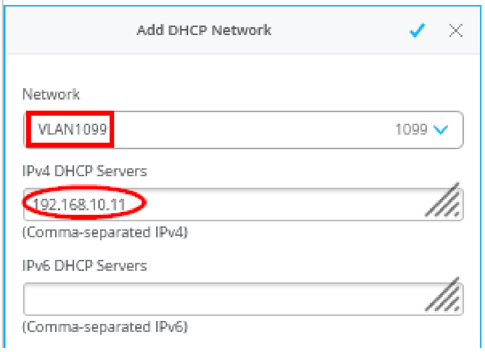
DHCP Networks

No Networks defined

Assign as DHCP relay:

- Network=`VLAN1099`
- IPv4 DHCP Servers=`192.168.10.11` (In a production network, use two servers as recommended)

Figure 34: DHCP Relay for First Network



Add DHCP Network

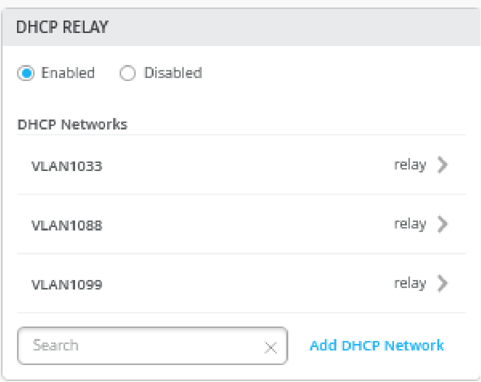
Network: VLAN1099 1099

IPv4 DHCP Servers: 192.168.10.11
(Comma-separated IPv4)

IPv6 DHCP Servers:
(Comma-separated IPv6)

Assign the two remaining networks to the same DHCP server=`192.168.10.11`

Figure 35: DHCP Relay for All Networks Configured



DHCP RELAY

● Enabled ○ Disabled

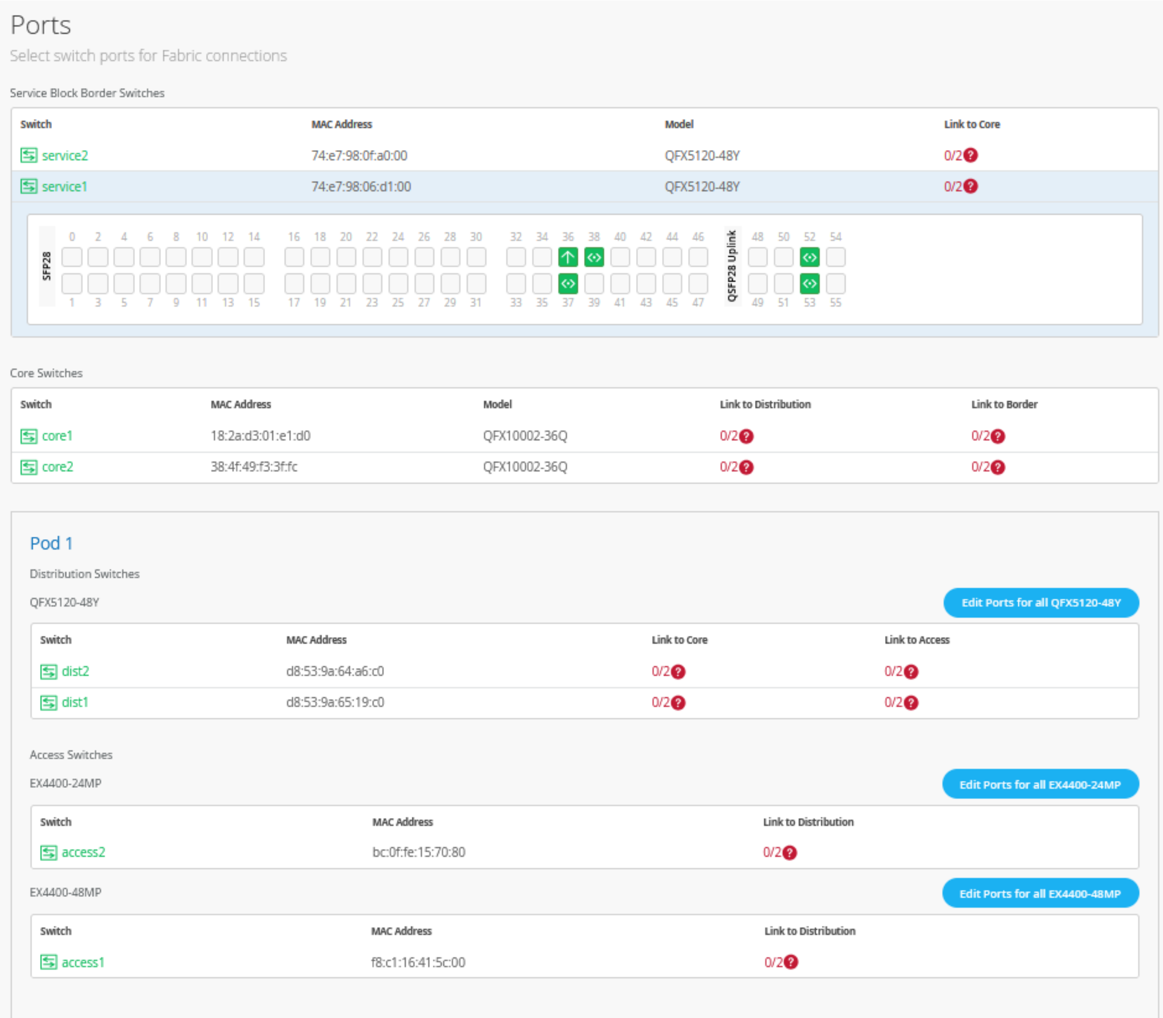
DHCP Networks

VLAN1033	relay
VLAN1088	relay
VLAN1099	relay

Configure Campus Fabric Ports

11. The final step is the selection of physical ports among service, core, distribution and access switches.

Figure 36: Port Overview



Ports

Select switch ports for Fabric connections

Service Block Border Switches

Switch	MAC Address	Model	Link to Core
service2	74:e7:98:0f:a0:00	QFX5120-48Y	0/2
service1	74:e7:98:06:d1:00	QFX5120-48Y	0/2

Core Switches

Switch	MAC Address	Model	Link to Distribution	Link to Border
core1	18:2a:d3:01:e1:d0	QFX10002-36Q	0/2	0/2
core2	38:4f:49:f3:3f:fc	QFX10002-36Q	0/2	0/2

Pod 1

Distribution Switches

QFX5120-48Y

Edit Ports for all QFX5120-48Y

Switch	MAC Address	Link to Core	Link to Access
dist2	d8:53:9a:64:a6:c0	0/2	0/2
dist1	d8:53:9a:65:19:c0	0/2	0/2

Access Switches

EX4400-24MP

Edit Ports for all EX4400-24MP

Switch	MAC Address	Link to Distribution
access2	bc:0f:fe:15:70:80	0/2

EX4400-48MP

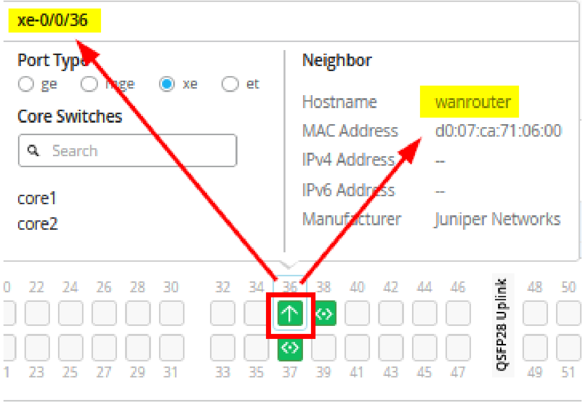
Edit Ports for all EX4400-48MP

Switch	MAC Address	Link to Distribution
access1	f8:c1:16:41:5c:00	0/2

In the current state shown here, all fabric switches are deployed and have OOBM connections to the Juniper Mist cloud to get managed. The fabric links among the switches are also wired already. As all switches send and receive LLDP neighbor messages, we can make use of this fact during the up and downlink configuration of each switch selecting the right interfaces. Here is how this process works using the example of service1 switch.

The links where nothing is attached to each other have no color, hence we just have to click in the remaining ports that are green one by one and review the interface name and reported link member. In the example below the interface reported is `xe-0/0/36` and the LLDP neighbor's name is `wanrouter` .

Figure 37: service1 switch port 36



xe-0/0/36

Port Type

○ ge ○ mge ● xe ○ et

Core Switches

Search

core1

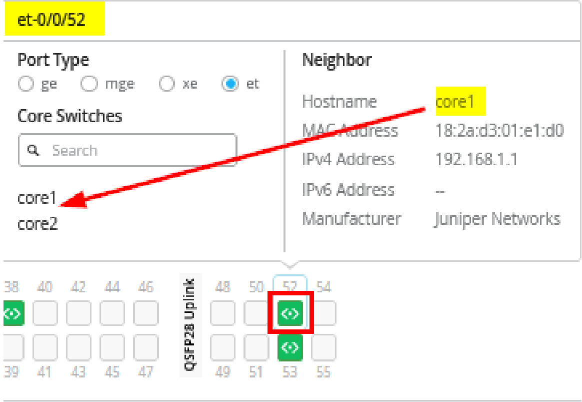
core2

Neighbor

Hostname	wanrouter
MAC Address	d0:07:ca:71:06:00
IPv4 Address	--
IPv6 Address	--
Manufacturer	Juniper Networks

When you go further, you click port 52 where the LLDP neighbor name `core1` also appears in the list of configurable switches. It is obvious that you only need to select `core1` from the list as it's already reported as an LLDP neighbor.

Figure 38: service1 switch port 52



et-0/0/52

Port Type

○ ge ○ mge ○ xe ● et

Core Switches

Search

core1

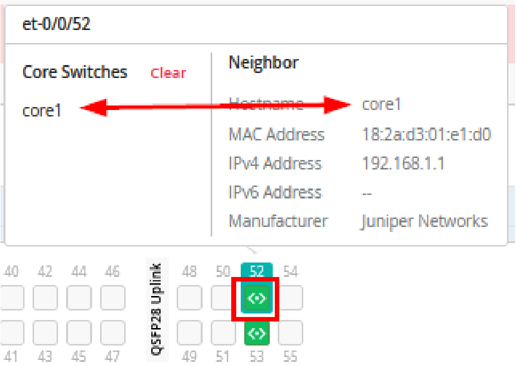
core2

Neighbor

Hostname	core1
MAC Address	18:2a:d3:01:e1:d0
IPv4 Address	192.168.1.1
IPv6 Address	--
Manufacturer	Juniper Networks

When you check the port after configuration, you should have 1:1 mapping between the configured switch name and the reported LLDP neighbor.

Figure 39: Check Port Config



et-0/0/52

Core Switches Clear

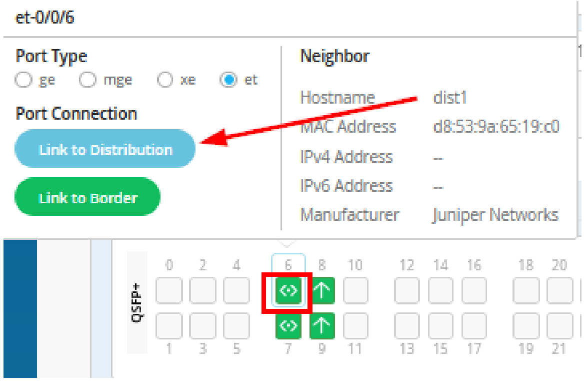
core1

Neighbor

Hostname	core1
MAC Address	18:2a:d3:01:e1:d0
IPv4 Address	192.168.1.1
IPv6 Address	--
Manufacturer	Juniper Networks

With other fabric switches you may need to additionally configure whether the neighbor is an uplink or downlink neighbor. In the example below, we are on port 6 of core1 switch. The LLDP neighbor name `dist1` tells us that this is a downlink neighbor from core to distribution. Hence, we click on **Link to Distribution**.

Figure 40: core1 switch port 6



et-0/0/6

Port Type

○ ge ○ mge ○ xe ● et

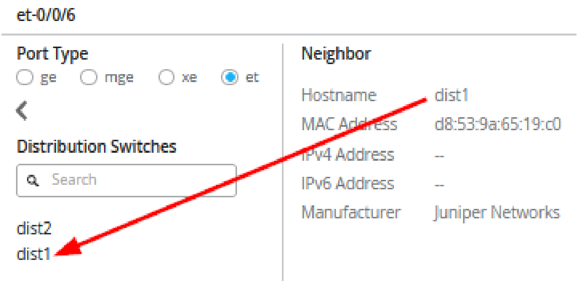
Port Connection

Neighbor

Hostname	dist1
MAC Address	d8:53:9a:65:19:c0
IPv4 Address	--
IPv6 Address	--
Manufacturer	Juniper Networks

When we selected the right Port Configuration, we should see the reported LLDP neighbor appearing in the list of selectable switches to be synced.

Figure 41: Sync the Reported LLDP Neighbor to Configured Switch



et-0/0/6

Port Type
○ ge ○ mge ○ xe ● et

Distribution Switches

Search

dist2
dist1

Neighbor

Hostname dist1
MAC Address d8:53:9a:65:19:c0
IPv4 Address --
IPv6 Address --
Manufacturer Juniper Networks

The above approach helps you make the right configuration choices based on the displayed LLDP neighbors and quickly finish the port configuration in a consistent manner. Also always look that you see the expected port type automatically appearing. If that is not the case, you may have wrong optics or missed some additional Junos CLI to convert a port from the default type to another one.

The other way to configure is based on the desired topology and then confirm the links coming up after fabric creation.

The entire configuration for this fabric is as follows:

- Node=`service1`
  - Port #=`1`
    - Used as=`Downlink`
    - Interface=`et-0/0/52`
    - Switch type=`Core Switches`
    - Switch Name Configured=`core1`
  - Port #=`2`
    - Used as=`Downlink`
    - Interface=`et-0/0/53`
    - Switch type=`Core Switches`
    - Switch Name Configured=`core2`
- Node=`service2`
  - Port #=`1`
    - Used as=`Downlink`
    - Interface=`et-0/0/52`

        - Switch type=`Core Switches`
        - Switch Name Configured=`core2`
    - Port #=`2`
        - Used as=`Downlink`
        - Interface=`et-0/0/53`
        - Switch type=`Core Switches`
        - Switch Name Configured=`core1`
- Node=`core1`
    - Port #=`1`
        - Used as=`Downlink`
        - Interface=`et-0/0/6`
        - Switch type=`Distribution Switches`
        - Switch Name Configured=`dist1`
    - Port #=`2`
        - Used as=`Downlink`
        - Interface=`et-0/0/7`
        - Switch type=`Distribution Switches`
        - Switch Name Configured=`dist2`
    - Port #=`3`
        - Used as=`Uplink`
        - Interface=`et-0/0/8`
        - Switch type=`Border Switches`
        - Switch Name Configured=`service1`
    - Port #=`4`
        - Used as=`Uplink`
        - Interface=`et-0/0/9`
        - Switch type=`Border Switches`

- Switch Name Configured=`service2`

- Node=`core2`
  - Port #=`1`
    - Used as=`Downlink`
    - Interface=`et-0/0/6`
    - Switch type=`Distribution Switches`
    - Switch Name Configured=`dist2`
  - Port #=`2`
    - Used as=`Downlink`
    - Interface=`et-0/0/7`
    - Switch type=`Distribution Switches`
    - Switch Name Configured=`dist1`
  - Port #=`3`
    - Used as=`Uplink`
    - Interface=`et-0/0/8`
    - Switch type=`Border Switches`
    - Switch Name Configured=`service2`
  - Port #=`4`
    - Used as=`Uplink`
    - Interface=`et-0/0/9`
    - Switch type=`Border Switches`
    - Switch Name Configured=`service1`
- Node=`dist1`
  - Port #=`1`
    - Used as=`Downlink`
    - Interface=`xe-0/0/36`
    - Switch type=`Access Switches`

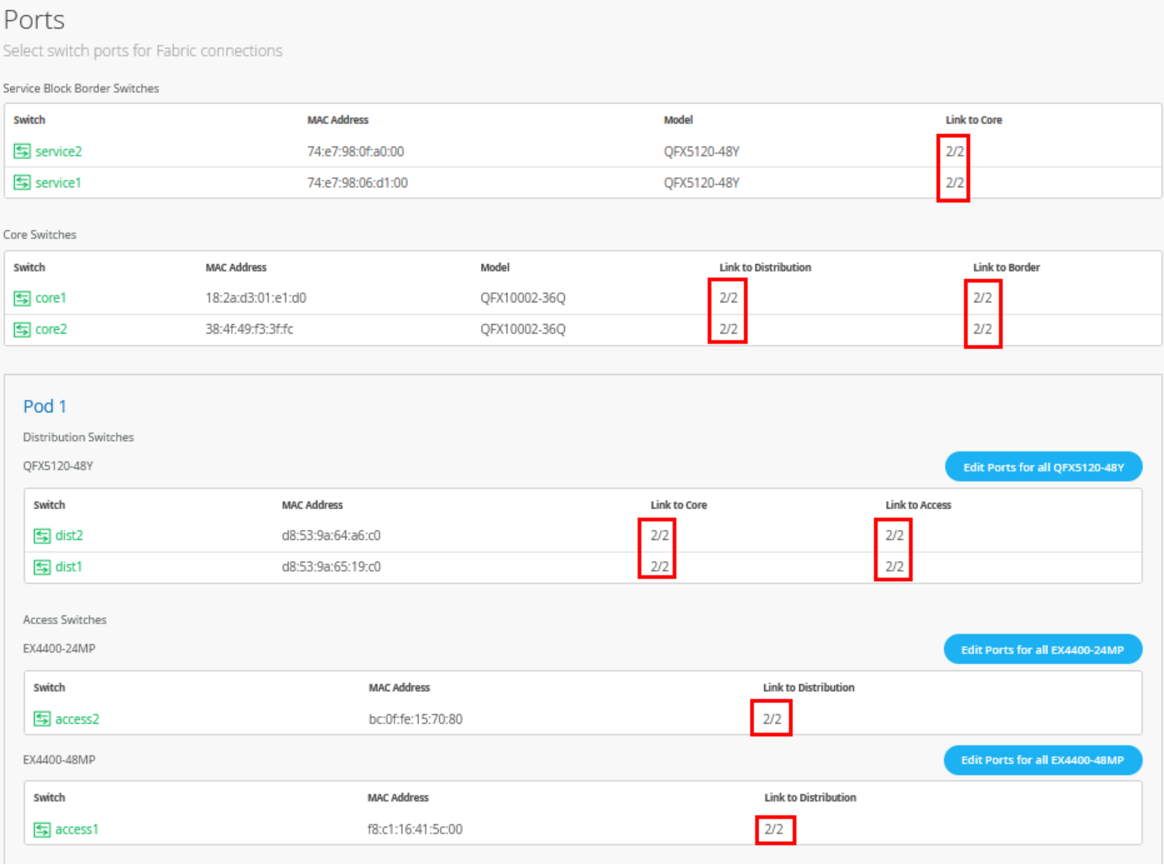
    - Switch Name Configured=`access1`
  - Port #=`2`
    - Used as=`Downlink`
    - Interface=`xe-0/0/37`
    - Switch type=`Access Switches`
    - Switch Name Configured=`access2`
  - Port #=`3`
    - Used as=`Uplink`
    - Interface=`et-0/0/52`
    - Switch type=`Core Switches`
    - Switch Name Configured=`core1`
  - Port #=`4`
    - Used as=`Uplink`
    - Interface=`et-0/0/53`
    - Switch type=`Core Switches`
    - Switch Name Configured=`core2`
- Node=`dist2`
  - Port #=`1`
    - Used as=`Downlink`
    - Interface=`xe-0/0/36`
    - Switch type=`Access Switches`
    - Switch Name Configured=`access2`
  - Port #=`2`
    - Used as=`Downlink`
    - Interface=`xe-0/0/37`
    - Switch type=`Access Switches`
    - Switch Name Configured=`access1`

    - Port #=`3`
        - Used as=`Uplink`
        - Interface=`et-0/0/52`
        - Switch type=`Core Switches`
        - Switch Name Configured=`core2`
    - Port #=`4`
        - Used as=`Uplink`
        - Interface=`et-0/0/53`
        - Switch type=`Core Switches`
        - Switch Name Configured=`core1`
- Node=`access1`
    - Port #=`1`
        - Used as=`Uplink`
        - Interface=`mge-0/0/36`
        - Switch type=`Distribution Switches`
        - Switch Name Configured=`dist1`
    - Port #=`2`
        - Used as=`Uplink`
        - Interface=`mge-0/0/37`
        - Switch type=`Distribution Switches`
        - Switch Name Configured=`dist2`
- Node=`access2`
    - Port #=`1`
        - Used as=`Uplink`
        - Interface=`xe-1/2/0`
        - Switch type=`Distribution Switches`
        - Switch Name Configured=`dist1`

- Port #=`2`
    - Used as=`Uplink`
    - Interface=`xe-2/2/0`
    - Switch type=`Distribution Switches`
    - Switch Name Configured=`dist2`

After you configure all requested links, you should see something similar.

Figure 42: Completed Final Port Configuration



Ports

Select switch ports for Fabric connections

Service Block Border Switches

Switch	MAC Address	Model	Link to Core
service2	74:e7:98:0f:a0:00	QFX5120-48Y	2/2
service1	74:e7:98:06:d1:00	QFX5120-48Y	2/2

Core Switches

Switch	MAC Address	Model	Link to Distribution	Link to Border
core1	18:2a:d3:01:e1:d0	QFX10002-36Q	2/2	2/2
core2	38:4f:49:f3:3f:fc	QFX10002-36Q	2/2	2/2

Pod 1

Distribution Switches

QFX5120-48Y

Switch	MAC Address	Link to Core	Link to Access
dist2	d8:53:9a:64:a6:c0	2/2	2/2
dist1	d8:53:9a:65:19:c0	2/2	2/2

Access Switches

EX4400-24MP

Switch	MAC Address	Link to Distribution
access2	bc:0f:fe:15:70:80	2/2

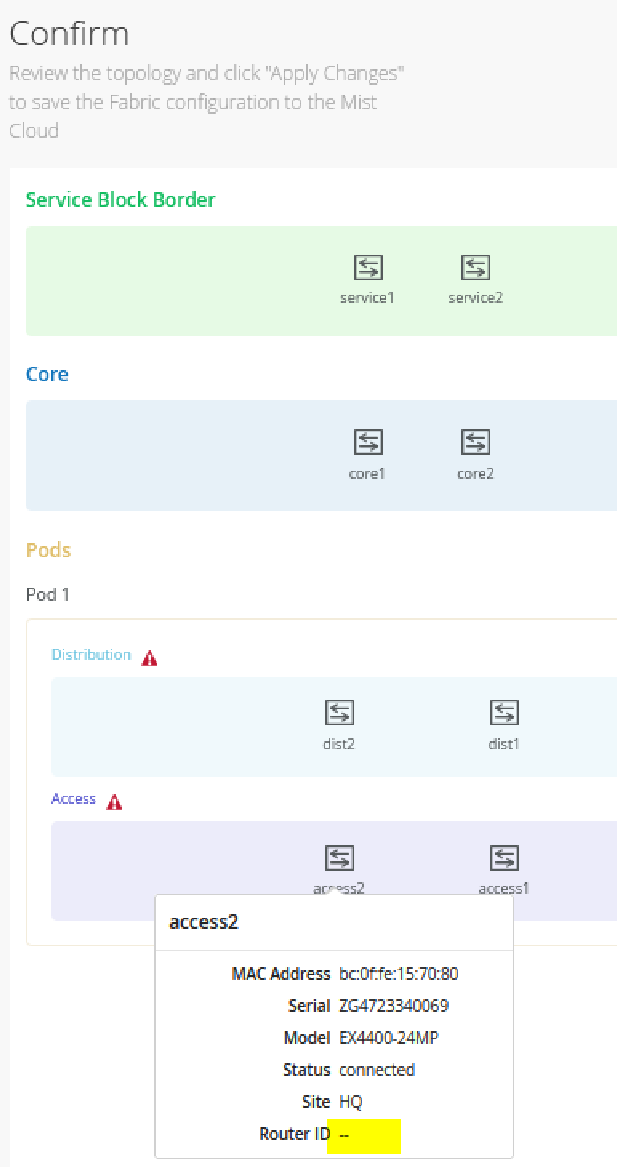
EX4400-48MP

Switch	MAC Address	Link to Distribution
access1	f8:c1:16:41:5c:00	2/2

Confirm Your Topology

12. Check your environment one last time before submitting the final configuration. In the below figure, notice that the router ID is still not assigned as this is a part of the first fabric configuration. Ignore this again.

Figure 43: Topology Check



Confirm

Review the topology and click "Apply Changes" to save the Fabric configuration to the Mist Cloud

Service Block Border

service1 service2

Core

core1 core2

Pods

Pod 1

Distribution ⚠

dist2 dist1

Access ⚠

access2 access1

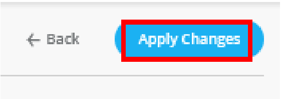
access2

MAC Address	bc:0f:fe:15:70:80
Serial	ZG4723340069
Model	EX4400-24MP
Status	connected
Site	HQ
Router ID	--

Apply Changes

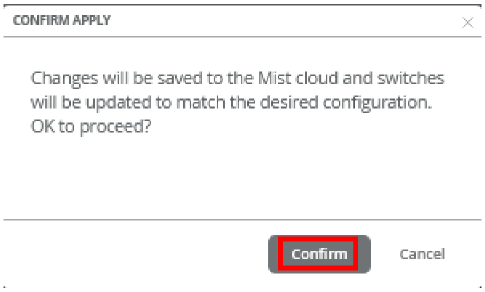
13. Now we can finally **Apply Changes**.

Figure 44: Apply Changes to Fabric



Confirm the dialogue a last time.

Figure 45: Confirm Apply



CONFIRM APPLY

Changes will be saved to the Mist cloud and switches will be updated to match the desired configuration. OK to proceed?

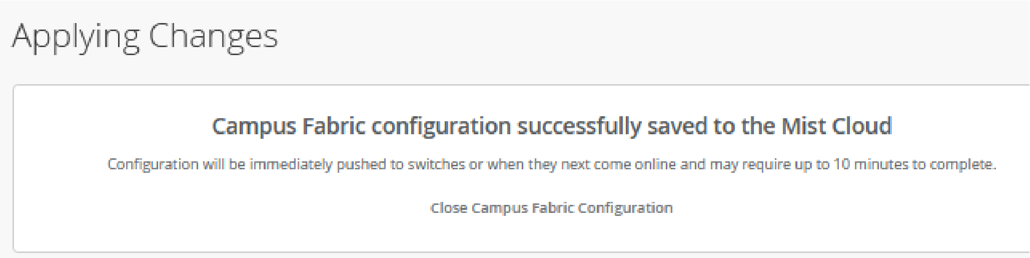
You must complete the second stage confirmation to create the fabric.

Juniper Mist displays the following banner including the estimated time for the campus fabric to be built. The process includes the following:

- Juniper Mist builds the point-to-point interfaces between service, core, distribution and access devices with IP addresses chosen from the range presented at the onset of the build.
- Each device is configured with a loopback address from the range presented at the onset of the build.
- eBGP is provisioned on each device with unique BGP autonomous system numbers. The primary goal of the underlay is to leverage ECMP for load balancing traffic on a per packet basis for device loopback reachability. The primary goal of the eBGP overlay is the support of customer traffic using EVPN-VXLAN.
- IP addressing of each L3 gateway IRB located on Access1 and Access2.
- IP addressing of each lo0.0 loopback, which is done automatically in this case.
- Configuration of routing policies for underlay and overlay connectivity.
- Optimized MTU settings for P2P underlay, L3 IRB, and ESI-LAG bundles.
- VXLAN-to-VLAN mapping using VNI addresses that are automatically assigned.
- VRF creation of corp-it, developers, and guest-wifi and VLAN associated with each VRF.
- VXLAN tunnelling creation between access devices and access-service block devices (in support of the northbound SRX WAN router that is configured in subsequent steps).
- Downloadable connection table (CSV format) that can be used by those involved in the physical buildout of the campus fabric.
- Graphical interface depicting all devices with BGP peering and physical link status.

You will now get the following message that your configuration was saved and the fabric will be built.

Figure 46: Fabric will be Built Now



Applying Changes

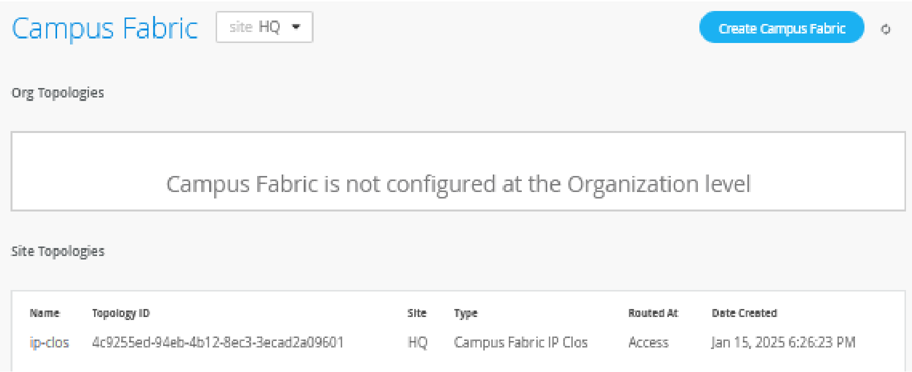
Campus Fabric configuration successfully saved to the Mist Cloud

Configuration will be immediately pushed to switches or when they next come online and may require up to 10 minutes to complete.

NOTE: As the dialogue window indicates, please wait 10 minutes before further actions.

14. Once you click **Close Campus Fabric Configuration**, you can view a summary of the newly created campus fabric IP Clos.

Figure 47: Created IP Clos Fabric View



Org Topologies

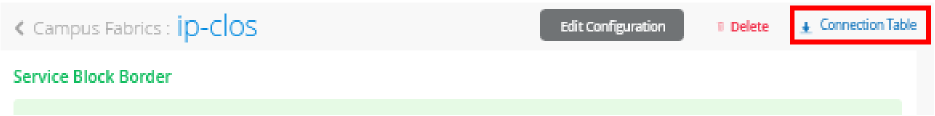
Campus Fabric is not configured at the Organization level

Site Topologies

Name	Topology ID	Site	Type	Routed At	Date Created
ip-clos	4c9255ed-94eb-4b12-8ec3-3ecad2a09601	HQ	Campus Fabric IP Clos	Access	Jan 15, 2025 6:26:23 PM

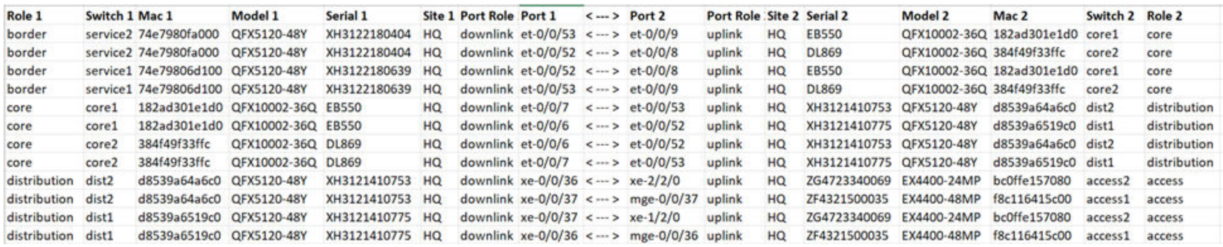
With Juniper Mist Wired Assurance, you can download a connection table (CSV format) representing the physical layout of the campus fabric. This can be used to validate all switch interconnects for those participating in the physical campus fabric build. Once the campus fabric is built or in the process of being built, you can download the connection table.

Figure 48: Download the CSV Connection Table



Connection Table spreadsheet:

Figure 49: Downloaded Connection Table



Role 1	Switch 1	Mac 1	Model 1	Serial 1	Site 1	Port Role	Port 1	< --- >	Port 2	Port Role	Site 2	Serial 2	Model 2	Mac 2	Switch 2	Role 2
border	service2	74e7980fa000	QFX5120-48Y	XH3122180404	HQ	downlink	et-0/0/53	< --- >	et-0/0/9	uplink	HQ	EB550	QFX10002-36Q	182ad301e1d0	core1	core
border	service2	74e7980fa000	QFX5120-48Y	XH3122180404	HQ	downlink	et-0/0/52	< --- >	et-0/0/8	uplink	HQ	DL869	QFX10002-36Q	384f49f33ffc	core2	core
border	service1	74e79806d100	QFX5120-48Y	XH3122180639	HQ	downlink	et-0/0/52	< --- >	et-0/0/8	uplink	HQ	EB550	QFX10002-36Q	182ad301e1d0	core1	core
border	service1	74e79806d100	QFX5120-48Y	XH3122180639	HQ	downlink	et-0/0/53	< --- >	et-0/0/9	uplink	HQ	DL869	QFX10002-36Q	384f49f33ffc	core2	core
core	core1	182ad301e1d0	QFX10002-36Q	EB550	HQ	downlink	et-0/0/7	< --- >	et-0/0/53	uplink	HQ	XH3121410753	QFX5120-48Y	d8539a64a6c0	dist2	distribution
core	core1	182ad301e1d0	QFX10002-36Q	EB550	HQ	downlink	et-0/0/6	< --- >	et-0/0/52	uplink	HQ	XH3121410775	QFX5120-48Y	d8539a6519c0	dist1	distribution
core	core2	384f49f33ffc	QFX10002-36Q	DL869	HQ	downlink	et-0/0/6	< --- >	et-0/0/52	uplink	HQ	XH3121410753	QFX5120-48Y	d8539a64a6c0	dist2	distribution
core	core2	384f49f33ffc	QFX10002-36Q	DL869	HQ	downlink	et-0/0/7	< --- >	et-0/0/53	uplink	HQ	XH3121410775	QFX5120-48Y	d8539a6519c0	dist1	distribution
distribution	dist2	d8539a64a6c0	QFX5120-48Y	XH3121410753	HQ	downlink	xe-0/0/36	< --- >	xe-2/2/0	uplink	HQ	ZG4723340069	EX4400-24MP	bc0ffe157080	access2	access
distribution	dist2	d8539a64a6c0	QFX5120-48Y	XH3121410753	HQ	downlink	xe-0/0/37	< --- >	mge-0/0/37	uplink	HQ	ZF4321500035	EX4400-48MP	f8c116415c00	access1	access
distribution	dist1	d8539a6519c0	QFX5120-48Y	XH3121410775	HQ	downlink	xe-0/0/37	< --- >	xe-1/2/0	uplink	HQ	ZG4723340069	EX4400-24MP	bc0ffe157080	access2	access
distribution	dist1	d8539a6519c0	QFX5120-48Y	XH3121410775	HQ	downlink	xe-0/0/36	< --- >	mge-0/0/36	uplink	HQ	ZF4321500035	EX4400-48MP	f8c116415c00	access1	access

Apply VLANs to Access Ports

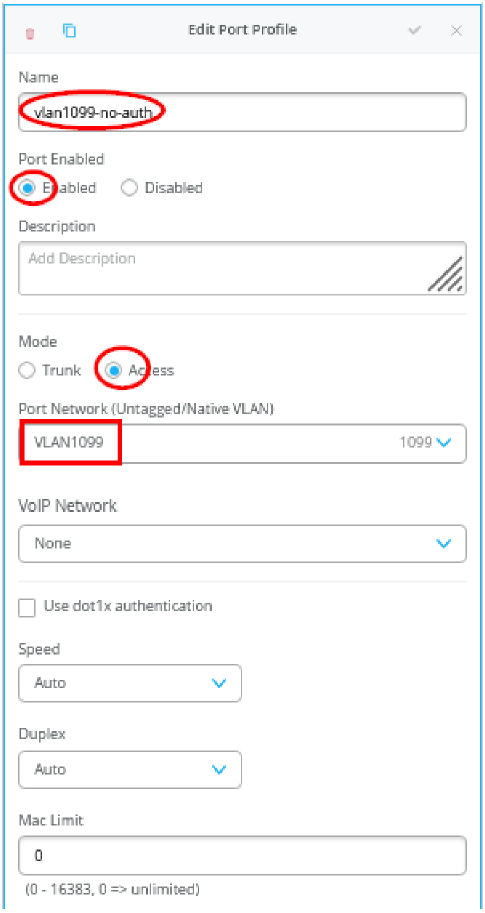
As previously discussed, Juniper Mist provides the ability to templatize well known services such as RADIUS, NTP, DNS, and so on that can be used across all devices within a site. These templates can also include VLANs and port profiles that can be targeted at each device within a site. The last step before verification is to associate VLANs with the requisite ports on each access switch.

In this case, Desktop1 to Desktop5 VMs and the access points are associated with different ports on each access switch which requires the configuration to be applied to Access1 and Access2, respectively. See Figure 14 on page 29.

It is highly recommended that you use the switch template to define the port profiles as this will make them available automatically on all switches for port configuration. In our case, we need the following three Port Profiles defined.

- First Port Profile
  - Name=`vlan1099-no-auth`
  - Port Enabled=`Enabled`
  - Mode=`Access`
  - Port Network (Untagged/Native VLAN)=`VLAN1099`

Figure 50: Port Profile VLAN1099



Edit Port Profile

Name

vlan1099-no-auth

Port Enabled

● Enabled ○ Disabled

Description

Add Description

Mode

○ Trunk ● Access

Port Network (Untagged/Native VLAN)

VLAN1099 1099

VoIP Network

None

□ Use dot1x authentication

Speed

Auto

Duplex

Auto

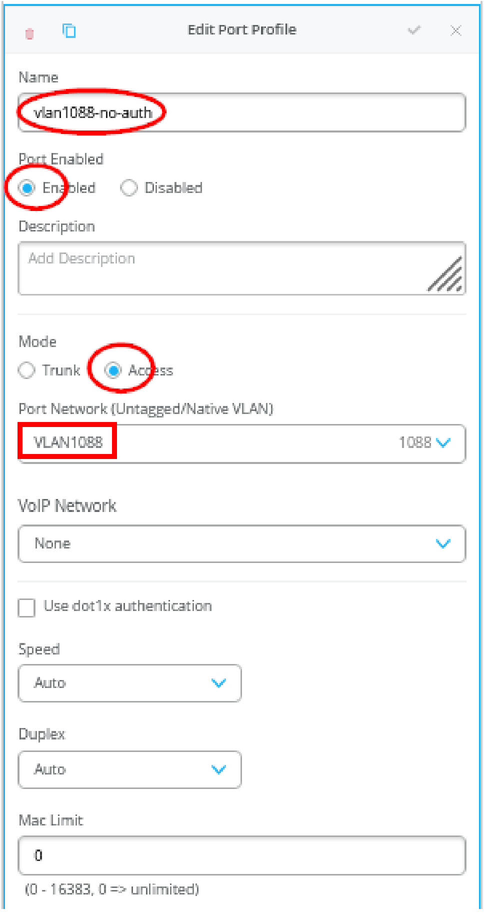
Mac Limit

0

(0 - 16383, 0 => unlimited)

- Second Port Profile
  - Name=`vlan1088-no-auth`
  - Port Enabled=`Enabled`
  - Mode=`Access`
  - Port Network (Untagged/Native VLAN)=`VLAN1088`

Figure 51: Port Profile VLAN1088



Edit Port Profile

Name

vlan1088-no-auth

Port Enabled

● Enabled ○ Disabled

Description

Add Description

Mode

○ Trunk ● Access

Port Network (Untagged/Native VLAN)

VLAN1088 1088

VoIP Network

None

□ Use dot1x authentication

Speed

Auto

Duplex

Auto

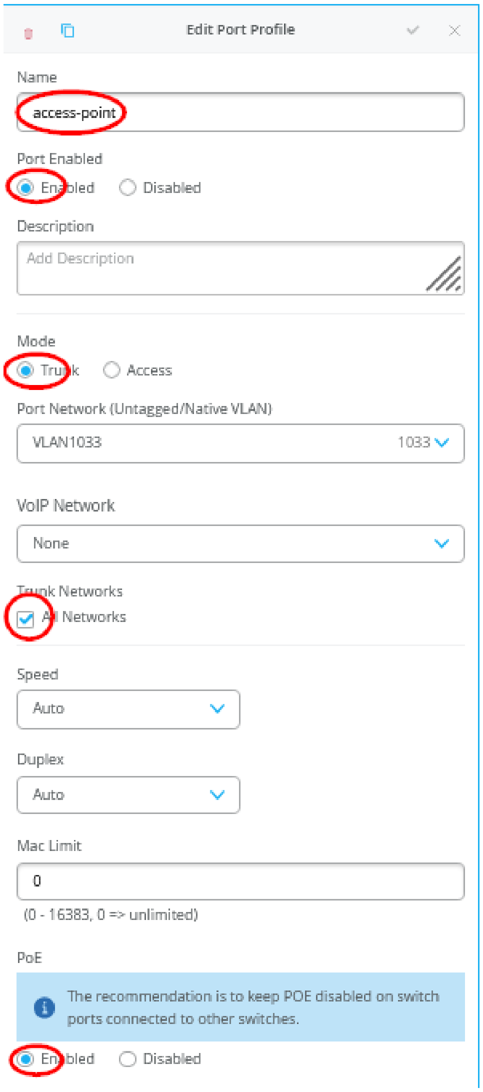
Mac Limit

0

(0 - 16383, 0 => unlimited)

- Third Port Profile
  - Name=`access-point`
  - Port Enabled=`Enabled`
  - Mode=`Trunk`
  - Port Network (Untagged/Native VLAN)=`VLAN1033`
  - Trunk Networks=`All Networks`
  - PoE=`Enabled`

Figure 52: Port Profile Access-Point



Edit Port Profile

Name

access-point

Port Enabled

● Enabled ○ Disabled

Description

Add Description

Mode

● Trunk ○ Access

Port Network (Untagged/Native VLAN)

VLAN1033 1033

VoIP Network

None

Trunk Networks

☑ All Networks

Speed

Auto

Duplex

Auto

Mac Limit

0

(0 - 16383, 0 => unlimited)

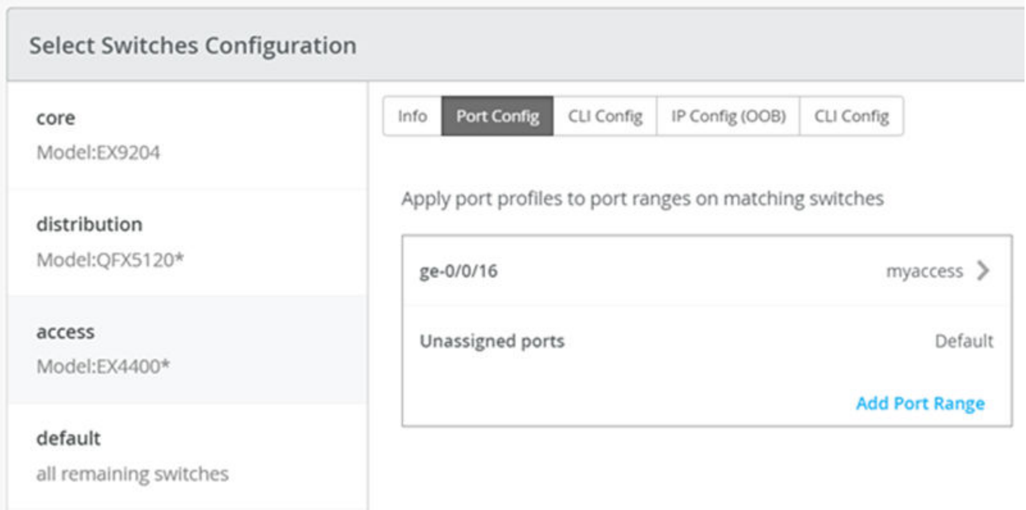
PoE

The recommendation is to keep POE disabled on switch ports connected to other switches.

● Enabled ○ Disabled

If most port configurations are the same across all access switches, a switch template can help to assign those port profiles automatically and you only need to take care of individual device configurations on each access switch. For example, the following found under the switch template option is customized to associate each switch with its role: service, core, distribution, and access. Furthermore, all access switches (defined by the EX4400 Switch, as an example) associated the AP port profile named “access-point” with ge-0/0/16 without needing to configure each independent switch.

Figure 53: Port Configuration Via Switch Template by Switch Model



Apply port profiles to port ranges on matching switches

ge-0/0/16 myaccess

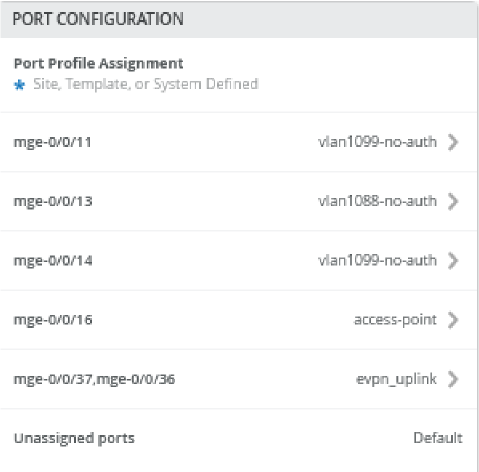
Unassigned ports Default

In our lab however, those port configurations are too unique to use a switch template, hence we assign them individually to each access switch in the following configuration:

- Access Switch=`access1`
  - Port1
    - Interface=`mge-0/0/14`
    - Port Profile=`vlan1099-no-auth`
    - Attached Lab function=`Desktop1 VM`
  - Port2
    - Interface=`mge-0/0/13`
    - Port Profile=`vlan1088-no-auth`
    - Attached Lab function=`Desktop3 VM`
  - Port3
    - Interface=`mge-0/0/11`
    - Port Profile=`vlan1099-no-auth`
    - Attached Lab function=`Desktop5 VM`
  - Port4
    - Interface=`mge-0/0/16`
    - Port Profile=`access-point`

        - Attached Lab function=`AP1`

Figure 54: Port Configuration access1 Switch



PORT CONFIGURATION

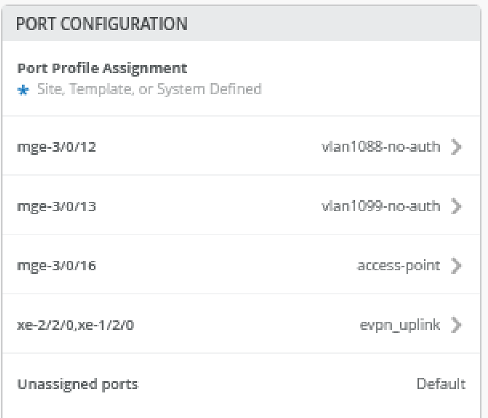
Port Profile Assignment

✱ Site, Template, or System Defined

mge-0/0/11	vlan1099-no-auth
mge-0/0/13	vlan1088-no-auth
mge-0/0/14	vlan1099-no-auth
mge-0/0/16	access-point
mge-0/0/37,mge-0/0/36	evpn_uplink
Unassigned ports	Default

- Access Switch=`access2`
    - Port1
        - Interface=`mge-3/0/12`
        - Port Profile=`vlan1088-no-auth`
        - Attached Lab function=`Desktop2 VM`
    - Port2
        - Interface=`mge-3/0/13`
        - Port Profile=`vlan1099-no-auth`
        - Attached Lab function=`Desktop4 VM`
    - Port3
        - Interface=`mge-3/0/16`
        - Port Profile=`access-point`
        - Attached Lab function=`AP2`

Figure 55: Port Configuration access2 Switch



PORT CONFIGURATION

Port Profile Assignment
* Site, Template, or System Defined

mge-3/0/12	vlan1088-no-auth
mge-3/0/13	vlan1099-no-auth
mge-3/0/16	access-point
xe-2/2/0,xe-1/2/0	evpn_uplink
Unassigned ports	Default

NOTE: At this point, the WAN router integration has not happened yet. However, the DHCP server for the lab is only available through the WAN router using DHCP relay. As a result, no DHCP lease handouts to clients or access points will happen at this time.

APPENDIX: IP Clos Fabric Verification (Optional)

IN THIS SECTION

This chapter is optional and can be skipped. It provides additional insight into the internal workings of the fabric for those interested in a deeper understanding.

The table below outlines what you will see within the fabric regarding wired clients using various commands.

MAC-Address	Location	IP-Address	VLAN-ID	VRF	Interface	VTEP
00:00:5e:e4:31:57	access1+2 sw	10.99.99.1	1099	corp-it	irb.1099	172.16.254.7+8
52:54:00:4a:e5:d0	Desktop5 VM	10.99.99.23	1099	corp-it	mge-0/0/11	172.16.254.8
52:54:00:77:3c:03	Desktop4 VM	10.99.99.42	1099	corp-it	mge-3/0/13	172.16.254.7
52:54:00:7e:f9:b1	Desktop1 VM	10.99.99.99	1099	corp-it	mge-0/0/14	172.16.254.8
00:00:5e:e4:31:57	access1+2 sw	10.88.88.1	1088	developer	irb.1088	172.16.254.7+8
52:54:00:7c:77:40	Desktop3 VM	10.88.88.23	1088	developer	mge-0/0/13	172.16.254.8
52:54:00:32:05:6f	Desktop2 VM	10.88.88.88	1088	developer	mge-3/0/12	172.16.254.7
00:00:5e:e4:31:57	access1+2 sw	10.33.33.1	1033	guest-wifi	irb.1033	172.16.254.7+8

Wired Client Verification

Verification of the campus fabric IP Clos deployment. See Figure 14 on page 29. Currently, there are five desktop VMs to validate the fabric with static IP address configuration. Let’s take a quick look to see if **Desktop1 VM** can connect internally and externally.

```
root@desktop1:~# ifconfig ens5
ens5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.99.99.99  netmask 255.255.255.0  broadcast 10.99.99.255
        inet6 fe80::5054:ff:fe7e:f9b1  prefixlen 64  scopeid 0x20<link>
        ether 52:54:00:7e:f9:b1  txqueuelen 1000  (Ethernet)
        RX packets 163054  bytes 424295446 (424.2 MB)
        RX errors 0  dropped 93549  overruns 0  frame 0
        TX packets 28662  bytes 1915505 (1.9 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
.
root@desktop1:~# ip r
              default via 10.99.99.1 dev ens5 proto static
10.99.99.0/24 dev ens5 proto kernel scope link src 10.99.99.99
192.168.10.0/24 dev ens3 proto kernel scope link src 192.168.10.61
.
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=64 time=3.45 ms
64 bytes from 10.99.99.1: icmp_seq=2 ttl=64 time=8.16 ms
64 bytes from 10.99.99.1: icmp_seq=3 ttl=64 time=9.19 ms
.
root@desktop1:~# ping -c3 10.99.99.23
PING 10.99.99.23 (10.99.99.23) 56(84) bytes of data.
64 bytes from 10.99.99.23: icmp_seq=1 ttl=64 time=0.589 ms
64 bytes from 10.99.99.23: icmp_seq=2 ttl=64 time=0.447 ms
64 bytes from 10.99.99.23: icmp_seq=3 ttl=64 time=0.477 ms
.
root@desktop1:~# ping -c3 10.99.99.42
PING 10.99.99.42 (10.99.99.42) 56(84) bytes of data.
64 bytes from 10.99.99.42: icmp_seq=1 ttl=64 time=0.793 ms
64 bytes from 10.99.99.42: icmp_seq=2 ttl=64 time=0.565 ms
64 bytes from 10.99.99.42: icmp_seq=3 ttl=64 time=0.538 ms
.
root@desktop1:~# ping -c3 10.88.88.88
PING 10.88.88.88 (10.88.88.88) 56(84) bytes of data.
From 10.99.99.1 icmp_seq=1 Destination Net Unreachable
From 10.99.99.1 icmp_seq=2 Destination Net Unreachable
From 10.99.99.1 icmp_seq=3 Destination Net Unreachable
.
root@desktop1:~# ping -c3 10.88.88.23
PING 10.88.88.23 (10.88.88.23) 56(84) bytes of data.
From 10.99.99.1 icmp_seq=1 Destination Net Unreachable
From 10.99.99.1 icmp_seq=2 Destination Net Unreachable
From 10.99.99.1 icmp_seq=3 Destination Net Unreachable
.
root@desktop1:~# ping -c3 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
From 10.99.99.1 icmp_seq=1 Destination Net Unreachable
From 10.99.99.1 icmp_seq=2 Destination Net Unreachable
From 10.99.99.1 icmp_seq=3 Destination Net Unreachable
```

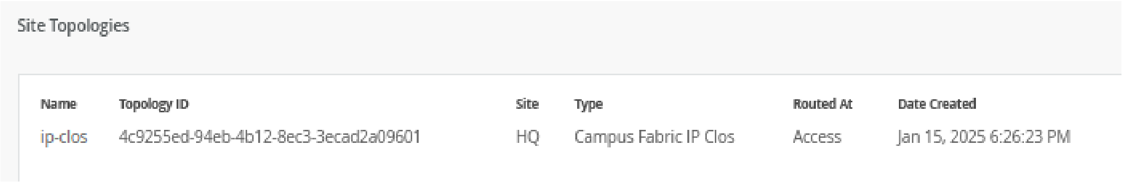
```
.
root@desktop1:~# arp -an
? (10.99.99.1) at 00:00:5e:e4:31:57 [ether] on ens5
? (10.99.99.23) at 52:54:00:4a:e5:d0 [ether] on ens5
? (10.99.99.42) at 52:54:00:77:3c:03 [ether] on ens5
.
root@desktop1:~# dhclient -v ens5
Listening on LPF/ens5/52:54:00:7e:f9:b1
Sending on   LPF/ens5/52:54:00:7e:f9:b1
Sending on   Socket/fallback
DHCPDISCOVER on ens5 to 255.255.255.255 port 67 interval 3 (xid=0x2358693f)
DHCPDISCOVER on ens5 to 255.255.255.255 port 67 interval 6 (xid=0x2358693f)
DHCPDISCOVER on ens5 to 255.255.255.255 port 67 interval 13 (xid=0x2358693f)
```

Validation steps:

- Confirmed local IP address, VLAN, and default gateway were configured on Desktop1 VM.
- Can ping default gateway—indicates that we can reach the access switch.
- Can ping Desktop5 VM in the same VLAN at the same switch.
- Can ping Desktop4 VM in the same VLAN at the other switch (access2 Virtual Chassis).
- Can not ping Desktop2 + 3 VM in different VRF. This is expected as there is isolation amongst VRFs in the fabric and we have not integrated the WAN router yet to hairpin that traffic.
- Can not ping the Internet. This is expected as we have not integrated the WAN router yet.
- ARP shows MAC addresses in Desktop1 VLAN reached to.
- No DHCP lease handouts. This is expected as our DHCP server is beyond the WAN router attached.

To further validate the campus fabric in the portal, select the **Campus Fabric** option under the **Organization** tab on the left side of the portal.

Figure 56: Fabric to Validate

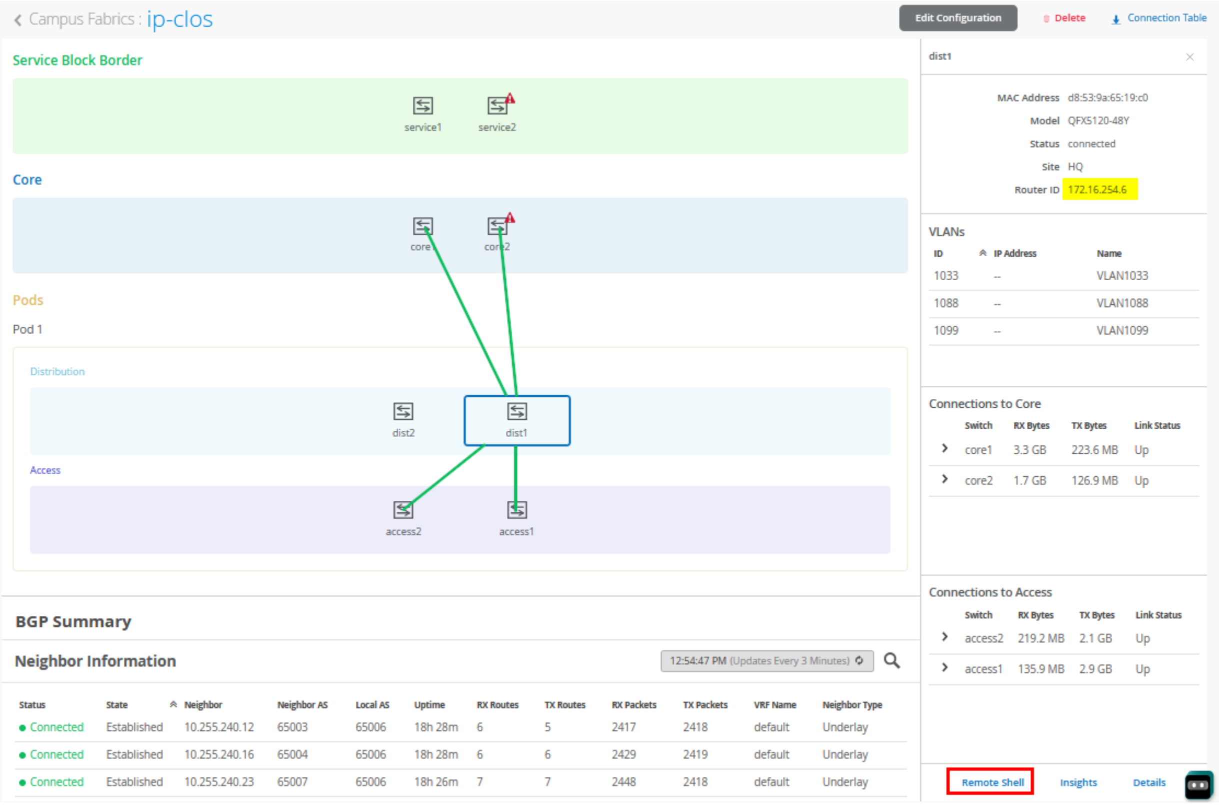


Site Topologies

Name	Topology ID	Site	Type	Routed At	Date Created
ip-clos	4c9255ed-94eb-4b12-8ec3-3ecad2a09601	HQ	Campus Fabric IP Clos	Access	Jan 15, 2025 6:26:23 PM

Remote shell access into each device within the campus fabric is supported here as well as a visual representation of the following capabilities:

- Automatically assigned router ID (lo0.0)
- BGP peering establishment.
- Transmit and receive traffic on a link-by-link basis.
- Telemetry, such as LLDP, from each device that verifies the physical build.



dist1

MAC Address	d8:53:9a:65:19:c0
Model	QFX5120-48Y
Status	connected
Site	HQ
Router ID	172.16.254.6

VLANs

ID	IP Address	Name
1033	--	VLAN1033
1088	--	VLAN1088
1099	--	VLAN1099

Connections to Core

Switch	RX Bytes	TX Bytes	Link Status
core1	3.3 GB	223.6 MB	Up
core2	1.7 GB	126.9 MB	Up

Connections to Access

Switch	RX Bytes	TX Bytes	Link Status
access2	219.2 MB	2.1 GB	Up
access1	135.9 MB	2.9 GB	Up

BGP Summary

Neighbor Information

12:54:47 PM (Updates Every 3 Minutes)

Status	State	Neighbor	Neighbor AS	Local AS	Uptime	RX Routes	TX Routes	RX Packets	TX Packets	VRF Name	Neighbor Type
● Connected	Established	10.255.240.12	65003	65006	18h 28m	6	5	2417	2418	default	Underlay
● Connected	Established	10.255.240.16	65004	65006	18h 28m	6	6	2429	2419	default	Underlay
● Connected	Established	10.255.240.23	65007	65006	18h 26m	7	7	2448	2418	default	Underlay

BGP Underlay Verification

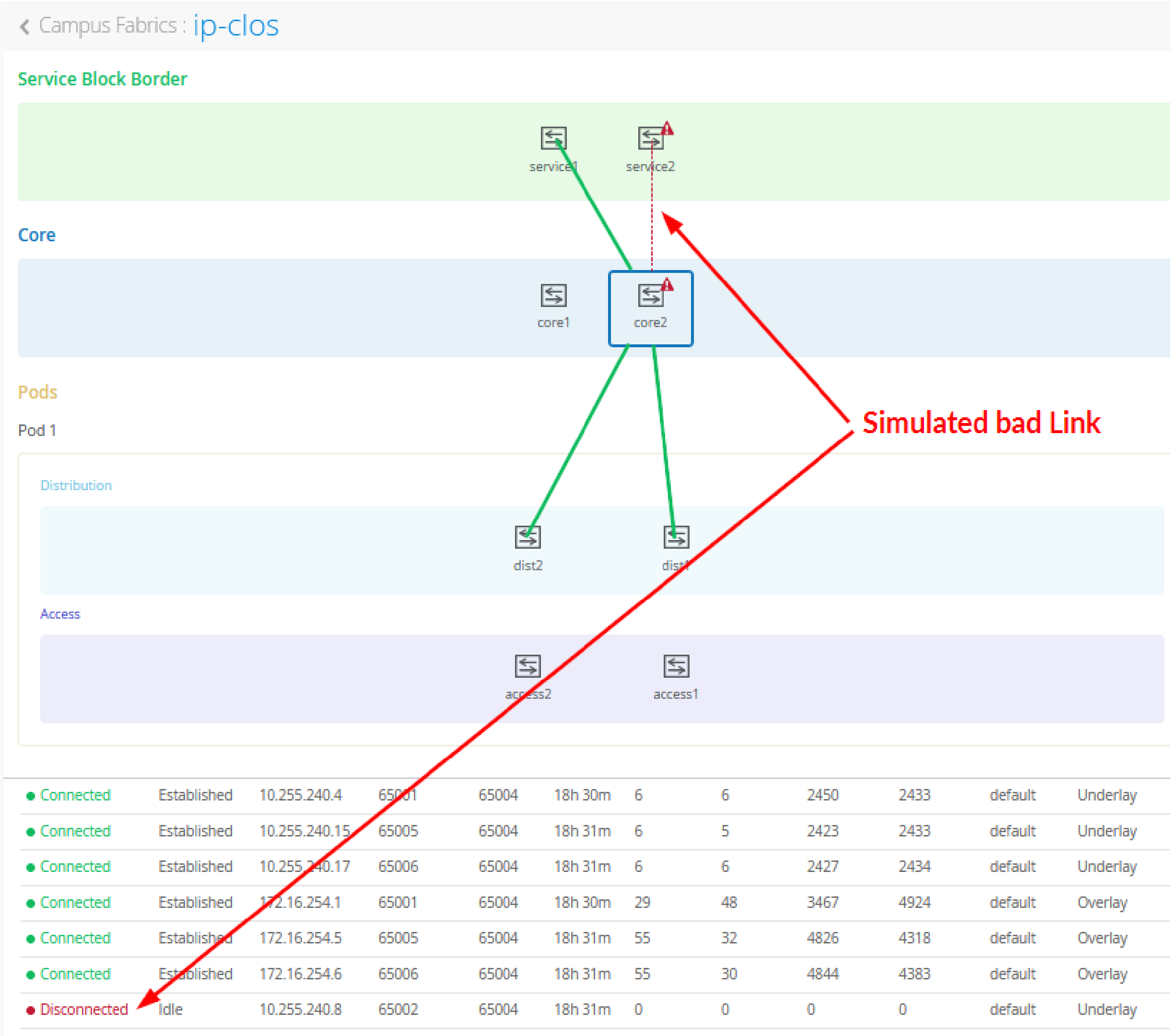
Purpose

Verifying the state of eBGP between the core and distribution layers is essential for EVPN-VXLAN to operate as expected. This network of point-to-point links between each layer supports:

- Load balancing using ECMP for greater resiliency and bandwidth efficiencies.
- BFD to decrease convergence times during failures.

- Loopback reachability to support VXLAN tunnelling.

Figure 57: Bad Fabric Link



● Connected	Established	10.255.240.4	65001	65004	18h 30m	6	6	2450	2433	default	Underlay
● Connected	Established	10.255.240.15	65005	65004	18h 31m	6	5	2423	2433	default	Underlay
● Connected	Established	10.255.240.17	65006	65004	18h 31m	6	6	2427	2434	default	Underlay
● Connected	Established	172.16.254.1	65001	65004	18h 30m	29	48	3467	4924	default	Overlay
● Connected	Established	172.16.254.5	65005	65004	18h 31m	55	32	4826	4318	default	Overlay
● Connected	Established	172.16.254.6	65006	65004	18h 31m	55	30	4844	4383	default	Overlay
● Disconnected	Idle	10.255.240.8	65002	65004	18h 31m	0	0	0	0	default	Underlay

In the above figure, you see a (simulated) bad link between two fabric nodes that you should fix.

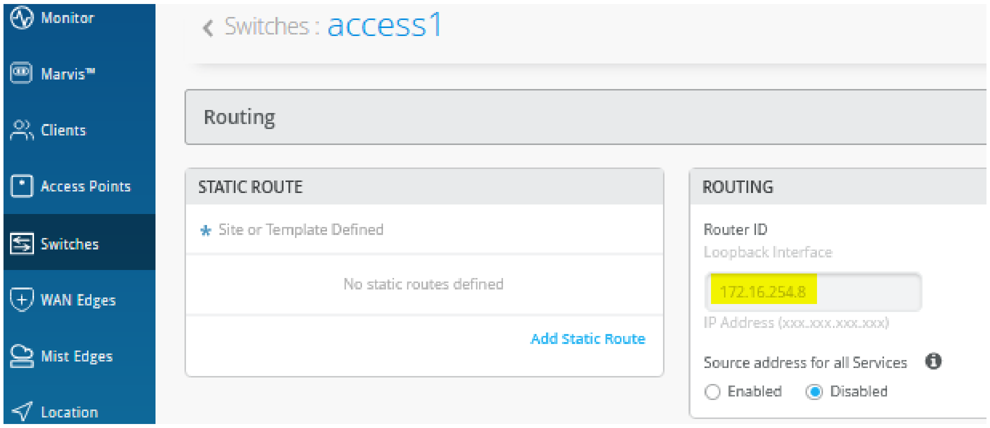
Due to the automated assignment of router IDs and loopback IP addresses in the fabric, we have the following configuration to remember:

Switch Type	Switch Name	Auto assigned Loopback IP
Service Block	service1	172.16.254.1

(Continued)

Switch Type	Switch Name	Auto assigned Loopback IP
Service Block	service2	172.16.254.2
Core	core1	172.16.254.3
Core	core2	172.16.254.4
Distribution	dist1	172.16.254.6
Distribution	dist2	172.16.254.5
Access Standalone	access1	172.16.254.8
Access Virtual Chassis	access2	172.16.254.7

Another way to find the assigned router ID and loopback IP address is the switch configuration panel as in the below figure:



Router ID
Loopback Interface
172.16.254.8
IP Address (xxx.xxx.xxx.xxx)

Source address for all Services
○ Enabled ● Disabled

Action

Verify that BGP sessions are established between core devices and distribution devices to ensure loopback reachability, BFD session status, and load-balancing using ECMP.

> **NOTE**: Operational data can be gathered through the campus fabric section of the portal as Remote Shell or using an external application such as SecureCRT or Putty.

Verification of BGP Peering

Access the Remote Shell via the lower-right corner of the campus fabric, from the switch view, or via Secure Shell (SSH).

Core1 Switch:

```
root@core1> show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 2 Peers: 8 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0
                      14         11          0          0          0          0
bgp.evpn.0
                     123         54          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
10.255.240.2          65001       2496       2471       0       0    18:52:17 Establ
  inet.0: 2/3/3/0
10.255.240.6          65002       2515       2472       0       0    18:52:16 Establ
  inet.0: 1/1/1/0
10.255.240.11         65005       2474       2474       0       0    18:53:35 Establ
  inet.0: 4/5/5/0
10.255.240.13         65006       2476       2472       0       0    18:53:37 Establ
  inet.0: 4/5/5/0
172.16.254.1          65001       4250       4380       0       0    18:52:17 Establ
  bgp.evpn.0: 1/27/27/0
172.16.254.2          65002       2517       5003       0       0    18:52:14 Establ
  bgp.evpn.0: 10/10/10/0
172.16.254.5          65005       4758       4467       0       0    18:53:33 Establ
  bgp.evpn.0: 19/43/43/0
172.16.254.6          65006       4723       4469       0       0    18:53:35 Establ
  bgp.evpn.0: 24/43/43/0
```

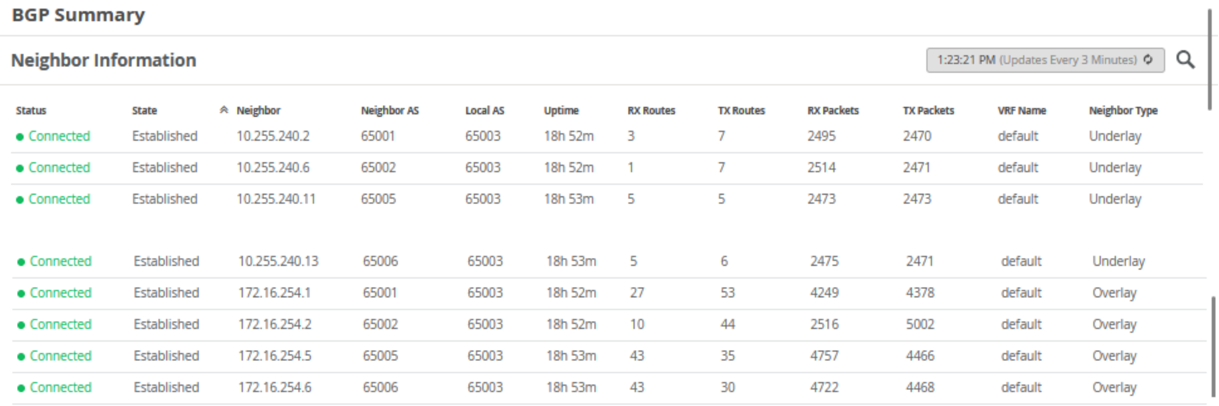
From the BGP summary, we can see that the underlay (10.255.240.X) peer relationships are established, which indicates that the underlay links are attached to the correct devices and the links are up.

It also shows the overlay (172.16.254.x) relationships are established and that it is peering at the correct loopback addresses. This demonstrates underlay loopback reachability.

We can also see routes received; time established are roughly equal which looks good so far.

The campus fabric build illustrates per device real-time BGP peering status shown below from Core1:

Figure 58: BGP Peering Reports



BGP Summary

Neighbor Information

Status	State	Neighbor	Neighbor AS	Local AS	Uptime	RX Routes	TX Routes	RX Packets	TX Packets	VRF Name	Neighbor Type
● Connected	Established	10.255.240.2	65001	65003	18h 52m	3	7	2495	2470	default	Underlay
● Connected	Established	10.255.240.6	65002	65003	18h 52m	1	7	2514	2471	default	Underlay
● Connected	Established	10.255.240.11	65005	65003	18h 53m	5	5	2473	2473	default	Underlay
● Connected	Established	10.255.240.13	65006	65003	18h 53m	5	6	2475	2471	default	Underlay
● Connected	Established	172.16.254.1	65001	65003	18h 52m	27	53	4249	4378	default	Overlay
● Connected	Established	172.16.254.2	65002	65003	18h 52m	10	44	2516	5002	default	Overlay
● Connected	Established	172.16.254.5	65005	65003	18h 53m	43	35	4757	4466	default	Overlay
● Connected	Established	172.16.254.6	65006	65003	18h 53m	43	30	4722	4468	default	Overlay

If BGP is not established then go back and validate the underlay links and addressing, and that the loopback addresses are correct. Loopback addresses should be pingable from other loopback addresses.

Verification of BGP connections can be performed on any of the other switches (not shown).

The primary goal of eBGP in the underlay is to provide loopback reachability between core and distribution devices in the campus fabric. This loopback is used to terminate VXLAN tunnels between devices. The following shows loopback reachability from Core1 to all devices in the campus fabric:

```
root@core1> ping 172.16.254.1 count 2
PING 172.16.254.1 (172.16.254.1): 56 data bytes
64 bytes from 172.16.254.1: icmp_seq=0 ttl=64 time=5.385 ms
64 bytes from 172.16.254.1: icmp_seq=1 ttl=64 time=5.962 ms
--- 172.16.254.1 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 5.385/5.673/5.962/0.289 ms
.
root@core1> ping 172.16.254.2 count 2
PING 172.16.254.2 (172.16.254.2): 56 data bytes
64 bytes from 172.16.254.2: icmp_seq=0 ttl=64 time=1.793 ms
64 bytes from 172.16.254.2: icmp_seq=1 ttl=64 time=7.798 ms
--- 172.16.254.2 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.793/4.795/7.798/3.003 ms
.
root@core1> ping 172.16.254.4 count 2
PING 172.16.254.4 (172.16.254.4): 56 data bytes
64 bytes from 172.16.254.4: icmp_seq=0 ttl=63 time=1.993 ms
64 bytes from 172.16.254.4: icmp_seq=1 ttl=63 time=2.082 ms
```

```
--- 172.16.254.4 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.993/2.038/2.082/0.044 ms
.
root@core1> ping 172.16.254.5 count 2
PING 172.16.254.5 (172.16.254.5): 56 data bytes
64 bytes from 172.16.254.5: icmp_seq=0 ttl=64 time=2.224 ms
64 bytes from 172.16.254.5: icmp_seq=1 ttl=64 time=2.883 ms
--- 172.16.254.5 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.224/2.554/2.883/0.329 ms
.
root@core1> ping 172.16.254.6 count 2
PING 172.16.254.6 (172.16.254.6): 56 data bytes
64 bytes from 172.16.254.6: icmp_seq=0 ttl=64 time=5.341 ms
64 bytes from 172.16.254.6: icmp_seq=1 ttl=64 time=3.041 ms
--- 172.16.254.6 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 3.041/4.191/5.341/1.150 ms
.
root@core1> ping 172.16.254.7 count 2
PING 172.16.254.7 (172.16.254.7): 56 data bytes
64 bytes from 172.16.254.7: icmp_seq=0 ttl=63 time=14.377 ms
64 bytes from 172.16.254.7: icmp_seq=1 ttl=63 time=4.518 ms
--- 172.16.254.7 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 4.518/9.447/14.377/4.929 ms
.
root@core1> ping 172.16.254.8 count 2
PING 172.16.254.8 (172.16.254.8): 56 data bytes
64 bytes from 172.16.254.8: icmp_seq=0 ttl=63 time=11.385 ms
64 bytes from 172.16.254.8: icmp_seq=1 ttl=63 time=6.309 ms
--- 172.16.254.8 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 6.309/8.847/11.385/2.538 ms
```

NOTE: eBGP sessions are established between core-distribution and core-service layers in the campus fabric. Loopback reachability has also been verified between the core and all other devices.

Let's verify that the routes are established to the access layer and other devices across multiple paths. For example, Core1 should leverage both paths through dist1 and dist2 to reach access1 and vice versa.

```
root@core1> show route forwarding-table destination 172.16.254.8
Routing table: default.inet
Internet:
Destination        Type RtRef Next hop           Type Index    NhRef Netif
172.16.254.8/32    user     0                    ulst  2097152     3
                              10.255.240.13      ucst     1725     7 et-0/0/6.0
                              10.255.240.11      ucst     1724     7 et-0/0/7.0
```

This is the reverse check from access1 to be able to reach core1 via dist1 and dist2 switches.

```
root@access1> show route forwarding-table destination 172.16.254.3
Routing table: default.inet
Internet:
Destination        Type RtRef Next hop           Type Index    NhRef Netif
172.16.254.3/32    user     0                    ulst   131070    21
                              10.255.240.24      ucst     1775     6 mge-0/0/36.0
                              10.255.240.20      ucst     1774     6 mge-0/0/37.0
```

Finally, we validate BFD for fast convergence in the case of a link or device failure:

```
root@core1> show bfd session
.                                                  Detect   Transmit
Address                  State     Interface      Time     Interval  Multiplier
10.255.240.2             Up        et-0/0/8.0     3.000     1.000        3
10.255.240.6             Up        et-0/0/9.0     3.000     1.000        3
10.255.240.11            Up        et-0/0/7.0     3.000     1.000        3
10.255.240.13            Up        et-0/0/6.0     3.000     1.000        3
172.16.254.1             Up                       3.000     1.000        3
172.16.254.2             Up                       3.000     1.000        3
172.16.254.5             Up                       3.000     1.000        3
172.16.254.6             Up                       3.000     1.000        3
.
8 sessions, 8 clients
Cumulative transmit rate 8.0 pps, cumulative receive rate 8.0 pps
```

Conclusion: At this point, the BGP underlay and overlay are operational through the verification of eBGP between corresponding layers of the campus fabric and loopback routes are established between service, core, distribution and access layers.

We have one link in the fabric that did not come up so we have not met the full requirement before we can go into production. However, we can further use the fabric as all nodes in the fabric can still reach each other.

EVPN VXLAN Verification Between Access and Service Block Switches

Since the desktop can ping its default gateway, we can assume the Ethernet switching tables are correctly populated, and VLAN and interface modes are correct. If pinging the default gateway failed, then troubleshoot client to access connectivity.

NOTE: In this fabric, the core and distribution switches do not have any VXLAN VTEPs assigned since no VLAN interface is configured there. Hence, we will not find any VXLAN or EVPN information on those devices. Those devices just forward our control and data packets.

Verification of the MAC address table, the EVPN database and remote VXLAN tunnel summary on both access switches is shown below:

Access1 Switch:

```
root@access1> 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,
           B - Blocked MAC)
Ethernet switching table : 7 entries, 7 learned
Routing instance : default-switch
   Vlan                MAC                MAC       GBP    Logical                SVLBNH/
Active
   name                address            flags     tag    interface              VENH Index
source
   VLAN1033            5c:5b:35:be:84:08  D                mge-0/0/16.0
   VLAN1033            5c:5b:35:be:84:12  DR               vtep.32771
172.16.254.7
   VLAN1088            52:54:00:32:05:6f  DR               vtep.32771
172.16.254.7
   VLAN1088            52:54:00:7c:77:40  D                mge-0/0/13.0
   VLAN1099            52:54:00:4a:e5:d0  D                mge-0/0/11.0
   VLAN1099            52:54:00:77:3c:03  DR               vtep.32771
172.16.254.7
```

```
   VLAN1099            52:54:00:7e:f9:b1   D                mge-0/0/14.0
.
root@access1> show evpn database
Instance: default-switch
VLAN  DomainId  MAC address        Active source                  Timestamp        IP address
     10001      f8:c1:16:41:5c:00  irb.0                          Jan 15 17:28:13
     11033      00:00:5e:e4:31:57  irb.1033                       Jan 16 10:51:10  10.33.33.1
     11033      5c:5b:35:be:84:08  mge-0/0/16.0                   Jan 16 10:51:20
     11033      5c:5b:35:be:84:12  172.16.254.7                   Jan 16 10:58:20
     11088      00:00:5e:e4:31:57  irb.1088                       Jan 16 10:51:10  10.88.88.1
     11088      52:54:00:32:05:6f  172.16.254.7                   Jan 16 11:02:28  10.88.88.88
     11088      52:54:00:7c:77:40  mge-0/0/13.0                   Jan 16 13:29:22  10.88.88.23
     11099      00:00:5e:e4:31:57  irb.1099                       Jan 16 10:51:10  10.99.99.1
     11099      3c:fd:fe:d1:65:60  172.16.254.7                   Jan 16 13:35:21
     11099      52:54:00:4a:e5:d0  mge-0/0/11.0                   Jan 16 13:29:22  10.99.99.23
     11099      52:54:00:77:3c:03  172.16.254.7                   Jan 16 11:02:39  10.99.99.42
     11099      52:54:00:7e:f9:b1  mge-0/0/14.0                   Jan 16 13:34:22  10.99.99.99
.
root@access1> show ethernet-switching vxlan-tunnel-end-point remote summary
Logical System Name       Id  SVTEP-IP         IFL   L3-Idx    SVTEP-Mode    ELP-SVTEP-IP
<default>                 0   172.16.254.8     lo0.0    0
 RVTEP-IP         L2-RTT                   IFL-Idx   Interface    NH-Id   RVTEP-Mode  ELP-
IP        Flags
 172.16.254.1     default-switch           609       vtep.32769   1776    RNVE
 172.16.254.2     default-switch           610       vtep.32770   1777    RNVE
 172.16.254.7     default-switch           611       vtep.32771   1778    RNVE
```

Access2 Switch:

```
root@access2> 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,
           B - Blocked MAC)
Ethernet switching table : 9 entries, 9 learned
Routing instance : default-switch
   Vlan                MAC                 MAC       GBP    Logical                SVLBNH/
Active
   name                address             flags     tag    interface              VENH Index
source
   VLAN1033            5c:5b:35:be:84:08   DR               vtep.32771
172.16.254.8
```

```
   VLAN1033            5c:5b:35:be:84:12   D                mge-3/0/16.0
   VLAN1088            3c:fd:fe:d1:65:60   DR               vtep.32771
172.16.254.8
   VLAN1088            52:54:00:32:05:6f   D                mge-3/0/12.0
   VLAN1088            52:54:00:7c:77:40   DR               vtep.32771
172.16.254.8
   VLAN1099            3c:fd:fe:d1:65:60   DR               vtep.32771
172.16.254.8
   VLAN1099            52:54:00:4a:e5:d0   DR               vtep.32771
172.16.254.8
   VLAN1099            52:54:00:77:3c:03   D                mge-3/0/13.0
   VLAN1099            52:54:00:7e:f9:b1   DR               vtep.32771
172.16.254.8
.
root@access2> show evpn database
Instance: default-switch
VLAN  DomainId  MAC address        Active source                  Timestamp        IP address
     10001      bc:0f:fe:15:70:80  irb.0                          Jan 15 17:28:15
     11033      00:00:5e:e4:31:57  irb.1033                       Jan 16 10:58:16  10.33.33.1
     11033      5c:5b:35:be:84:08  172.16.254.8                   Jan 16 10:51:20
     11033      5c:5b:35:be:84:12  mge-3/0/16.0                   Jan 16 10:58:20
     11088      00:00:5e:e4:31:57  irb.1088                       Jan 16 10:58:16  10.88.88.1
     11088      52:54:00:32:05:6f  mge-3/0/12.0                   Jan 16 13:39:25  10.88.88.88
     11088      52:54:00:7c:77:40  172.16.254.8                   Jan 16 11:03:59  10.88.88.23
     11099      00:00:5e:e4:31:57  irb.1099                       Jan 16 10:58:16  10.99.99.1
     11099      3c:fd:fe:d1:65:60  mge-3/0/13.0                   Jan 16 13:35:21
     11099      52:54:00:4a:e5:d0  172.16.254.8                   Jan 16 11:03:53  10.99.99.23
     11099      52:54:00:77:3c:03  mge-3/0/13.0                   Jan 16 13:39:25  10.99.99.42
     11099      52:54:00:7e:f9:b1  172.16.254.8                   Jan 16 11:06:20  10.99.99.99
.
root@access2> show ethernet-switching vxlan-tunnel-end-point remote summary
Logical System Name       Id  SVTEP-IP         IFL   L3-Idx    SVTEP-Mode    ELP-SVTEP-IP
<default>                 0   172.16.254.7     lo0.0    0
 RVTEP-IP         L2-RTT                   IFL-Idx   Interface    NH-Id   RVTEP-Mode  ELP-
IP        Flags
 172.16.254.1     default-switch           693       vtep.32769   1870    RNVE
 172.16.254.2     default-switch           694       vtep.32770   1871    RNVE
 172.16.254.8     default-switch           695       vtep.32771   1877    RNVE
```

Both access switches have all three VLANs assigned locally hence they will show information about local MAC-Addresses as well as once learned remotely via EVPN. The EVPN Database also reflects this learning. Also, the individual remote VXLAN VTEP's are learned using the router ID / lo0.0 assigned to remote switches.

Verification of the MAC-Address table, the EVPN Database and remote VXLAN-Tunnel summary on both Service Block Switches.

Service1 Switch:

```
root@service1> show ethernet-switching table
.
root@service1> show evpn database
Instance: default-switch
VLAN  DomainId  MAC address        Active source                  Timestamp        IP address
     10001      74:e7:98:06:d1:00  irb.0                          Jan 15 17:27:58
     11033      5c:5b:35:be:84:08  172.16.254.8                   Jan 16 10:51:20
     11033      5c:5b:35:be:84:12  172.16.254.7                   Jan 16 10:58:20
     11088      52:54:00:32:05:6f  172.16.254.7                   Jan 16 11:02:28  10.88.88.88
     11088      52:54:00:7c:77:40  172.16.254.8                   Jan 16 11:03:59  10.88.88.23
     11099      3c:fd:fe:d1:65:60  172.16.254.8                   Jan 16 14:06:14
     11099      52:54:00:4a:e5:d0  172.16.254.8                   Jan 16 11:03:53  10.99.99.23
     11099      52:54:00:77:3c:03  172.16.254.7                   Jan 16 11:02:39  10.99.99.42
     11099      52:54:00:7e:f9:b1  172.16.254.8                   Jan 16 11:06:20  10.99.99.99
.
root@service1> show ethernet-switching vxlan-tunnel-end-point remote summary
Logical System Name       Id  SVTEP-IP         IFL   L3-Idx    SVTEP-Mode    ELP-SVTEP-IP
<default>                 0   172.16.254.1     lo0.0    0
 RVTEP-IP         L2-RTT                   IFL-Idx   Interface    NH-Id   RVTEP-Mode  ELP-
IP        Flags
 172.16.254.2     default-switch           822       vtep.32769   1750    RNVE
 172.16.254.7     default-switch           823       vtep.32770   1753    RNVE
 172.16.254.8     default-switch           824       vtep.32771   1754    RNVE
```

Service2 Switch:

```
root@service2> show ethernet-switching table
.
root@service2> show evpn database
Instance: default-switch
VLAN  DomainId  MAC address        Active source                  Timestamp        IP address
     10001      74:e7:98:0f:a0:00  irb.0                          Jan 15 17:27:58
     11033      5c:5b:35:be:84:08  172.16.254.8                   Jan 16 10:51:20
     11033      5c:5b:35:be:84:12  172.16.254.7                   Jan 16 10:58:20
     11088      52:54:00:32:05:6f  172.16.254.7                   Jan 16 11:02:28  10.88.88.88
     11088      52:54:00:7c:77:40  172.16.254.8                   Jan 16 11:03:59  10.88.88.23
     11099      3c:fd:fe:d1:65:60  172.16.254.8                   Jan 16 14:06:14
     11099      52:54:00:4a:e5:d0  172.16.254.8                    Jan 16 11:03:53  10.99.99.23
     11099      52:54:00:77:3c:03  172.16.254.7                    Jan 16 11:02:39  10.99.99.42
     11099      52:54:00:7e:f9:b1  172.16.254.8                    Jan 16 11:06:20  10.99.99.99
     11099      ac:1f:6b:02:af:59  172.16.254.8                    Jan 16 14:10:15
.
root@service2> show ethernet-switching vxlan-tunnel-end-point remote summary
Logical System Name       Id  SVTEP-IP         IFL   L3-Idx    SVTEP-Mode    ELP-SVTEP-IP
<default>                 0   172.16.254.2     lo0.0    0
 RVTEP-IP         L2-RTT                   IFL-Idx   Interface    NH-Id   RVTEP-Mode  ELP-
IP        Flags
 172.16.254.1     default-switch           848       vtep.32769   1834    RNVE
 172.16.254.7     default-switch           849       vtep.32770   1839    RNVE
 172.16.254.8     default-switch           850       vtep.32771   1840    RNVE
```

At this point, you won't see any data in the Ethernet switching table on the service block switches, since no VLAN is defined there yet. The EVPN database is in sync between the service block switches, which is expected. Also, the individual remote VXLAN VTEPs are learned using the router ID / lo0.0 assigned to the remote switches.

In the next step, we verify the configuration of `VLAN1099` and what VNI was assigned to it on the two access and service block switches:

```
root@access1> show configuration vlans | display set | display inheritance | match 1099
set vlans VLAN1099 vlan-id 1099
set vlans VLAN1099 l3-interface irb.1099
set vlans VLAN1099 vxlan vni 11099
.
root@access2> show configuration vlans | display set | display inheritance | match 1099
set vlans VLAN1099 vlan-id 1099
set vlans VLAN1099 l3-interface irb.1099
set vlans VLAN1099 vxlan vni 11099
.
root@service1> show configuration vlans | display set | display inheritance | match 1099
.
root@service1>
.
root@service2> show configuration vlans | display set | display inheritance | match 1099
.
root@service2>
```

NOTE: Juniper Mist cloud management does not enable a user to configure the value for the VNI used for a particular VLAN so as to maintain consistency among different fabrics. Today the VNI is calculated by the VLAN-ID + 10.000 (but may change in the future).

We need to configure the attachment of the WAN router to complete the entire design. Without the WAN router configuration, the fabric only enables the following communications:

- The same VLAN/VNI on the same access switch but different ports.
- The same VLAN/VNI on different access switches.
- Different VLAN/VNI attached to the same VRF on the same access switch, but different ports.
- Different VLAN/VNI attached to the same VRF on different access switches.

All traffic between VRFs is always isolated inside the fabric. For security reasons, there is no possible configuration to perform route leaking between VRFs. This means that traffic between them is handled directly inside the fabric without the need to traverse through the WAN router as a possible enforcement point.

APPENDIX: WAN Router Integration into the Fabric

IN THIS SECTION

In general, there are several possible ways to attach a WAN router to a campus fabric.

- Using a L2 forwarding method:
  - The fabric uplinks are configured as ESI-LAGs and contain one or more tagged VLANs (one for each VRF) to communicate with the WAN router.

  - It is also necessary that you configure the IP address of the WAN router interface manually as the next-hop IP address for default-forwarding on each fabric VRF as already shown above.
  - The WAN router itself needs to understand standard IEEE 802.3ad LAG with active LACP.
  - If you have more than one WAN router attached for redundancy, it is advised to provide failover mechanisms between them for the interface IP addresses towards the fabric. VRRP is recommended.
  - Routes between fabric and WAN router are only statically configured.
- Using an L3 forwarding method:
  - The fabric uplinks are configured as L3 peer-to-peer IP links.
  - Per fabric VRF, a peer-to-peer link needs to be established with the WAN router.
  - Usually, there are multiple peer-to-peer links on a single physical uplink. Those are further segmented using tagged VLANs to provide isolation on the uplinks.
  - There is no need to manually configure next hops for each VRF inside the fabric as it is assumed that the propagation of the default gateways will be obtained from the WAN router through a routing protocol.
  - Between the fabric and the WAN router, a routing protocol must be established to exchange routes.
  - The campus fabric supports exterior BGP and OSPF as routing protocols towards the WAN router.

NOTE: The details of such integration are explained in the following JVD extension for all fabric types. We kept the explanations in this chapter brief because it's better documented along with the backgrounds in the referred extension.

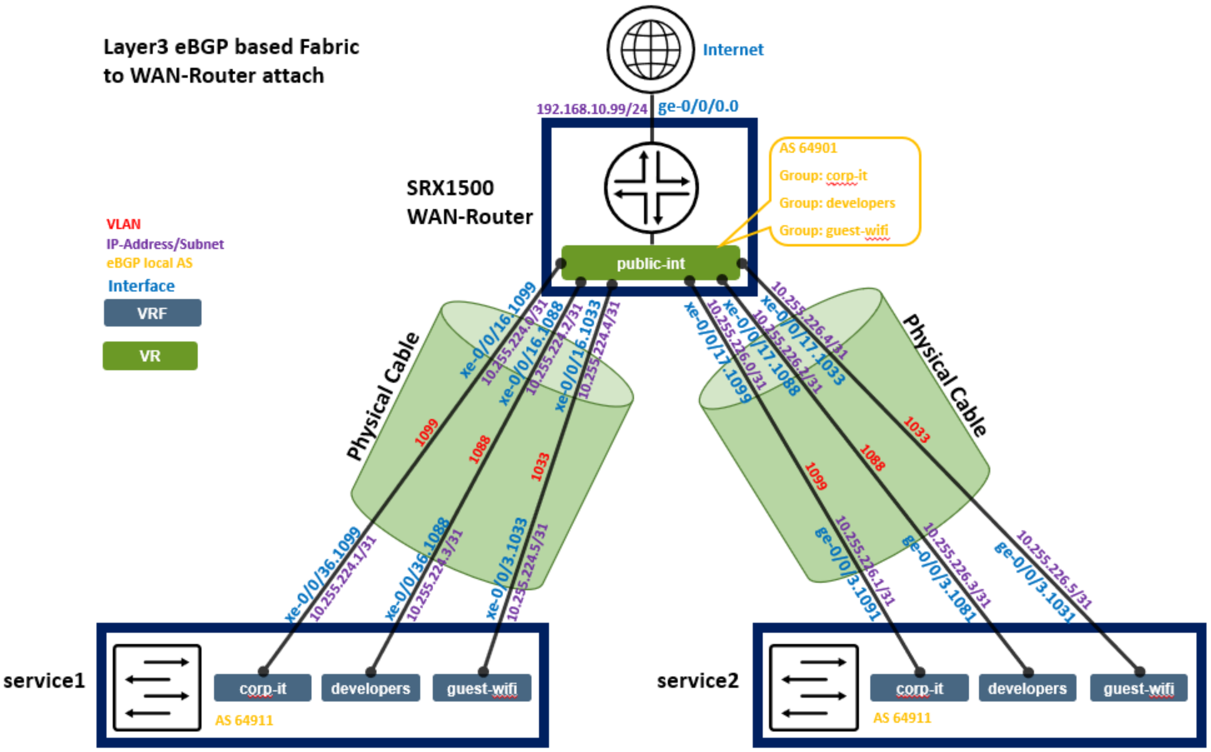
In our lab, we decided to go with L3 eBGP forwarding integration with the SRX WAN router. Before doing this, we reviewed the routing table for the `corp-it` VRF on the access1 switch, as shown below:

```
root@access1> show route table corp-it.inet.0
corp-it.inet.0: 7 destinations, 8 routes (7 active, 0 holddown, 0 hidden)
@ = Routing Use Only, # = Forwarding Use Only
+ = Active Route, - = Last Active, * = Both
10.99.99.0/24      *[Direct/0] 03:54:18
                    >  via irb.1099
                    [EVPN/170] 03:47:12
```

```
                        to 10.255.240.24 via mge-0/0/36.0
                      > to 10.255.240.20 via mge-0/0/37.0
10.99.99.1/32      *[Local/0] 03:54:18
                        Local via irb.1099
10.99.99.23/32     *[EVPN/7] 03:41:35
                      > via irb.1099
10.99.99.42/32     *[EVPN/170] 03:42:49
                      > to 10.255.240.24 via mge-0/0/36.0
                        to 10.255.240.20 via mge-0/0/37.0
10.99.99.99/32     *[EVPN/7] 03:39:08
                      > via irb.1099
172.16.192.4/32    *[Direct/0] 03:54:19
                      > via lo0.1
172.16.192.7/32    *[EVPN/170] 03:47:12
                        to 10.255.240.24 via mge-0/0/36.0
                      > to 10.255.240.20 via mge-0/0/37.0
```

For our lab with one WAN router and three VRFs, the configurations can be reviewed from the figure below:

Figure 59: Fabric WAN Router Integration via eBGP



Below, you see the integration information again as a table for each peering to be configured.

Switch	Switch AS	VRF	Service P2P IP	Service IF	WAN Router	WAN Router P2P IP	WAN Router AS	WAN Router IF	VLAN-ID
service1	64911	corp-it	10.255.224.1/31	xe-0/0/36.1099	wanrouter	10.255.224.0/31	64901	xe-0/0/16.1099	1099
service1	64911	developers	10.255.224.3/31	xe-0/0/36.1088	wanrouter	10.255.224.2/31	64901	xe-0/0/16.1088	1088
service1	64911	guest-wifi	10.255.224.5/31	xe-0/0/36.1033	wanrouter	10.255.224.4/31	64901	xe-0/0/16.1033	1033
service2	64911	corp-it	10.255.226.1/31	xe-0/0/36.1099	wanrouter	10.255.226.0/31	64901	xe-0/0/17.1099	1099
service2	64911	developers	10.255.226.3/31	xe-0/0/36.1088	wanrouter	10.255.226.2/31	64901	xe-0/0/17.1088	1088
service2	64911	guest-wifi	10.255.226.5/31	xe-0/0/36.1033	wanrouter	10.255.226.4/31	64901	xe-0/0/17.1033	1033

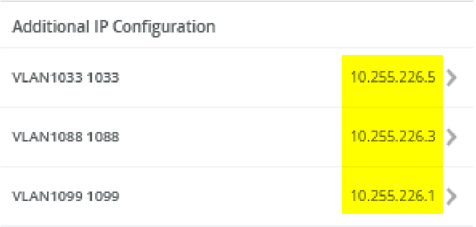
Configuration of service1 Block Switch for WAN Router Integration

We have a verbal description of what needs to be configured on this system here:

```
# configure the Additional IP-Subnet 10.255.224.1 255.255.255.254 to Network/VLAN:VLAN1099
# configure the Additional IP-Subnet 10.255.224.3 255.255.255.254 to Network/VLAN:VLAN1088
# configure the Additional IP-Subnet 10.255.224.5 255.255.255.254 to Network/VLAN:VLAN1033
# Then bind these 3 Network/VLANs to Port Interface xe-0/0/36 as L3-Sub-Interfaces with MTU=9018
.
# Enable BGP and create an Export policy called 'export-vrfs'
# Add to this export Policy the following Networks as:
# - Add Term w. Name=VLAN1099  Prefix=10.99.99.0/24  Protocol=None  Then=Accept
# - Add Term w. Name=VLAN1088  Prefix=10.88.88.0/24  Protocol=None  Then=Accept
# - Add Term w. Name=VLAN1033  Prefix=10.33.33.0/24  Protocol=None  Then=Accept
# - Add Term w. Name=overlaylo0 Prefix=172.16.192.0/24-32  Protocol=None  Then=Accept
.
# Create an Export policy called 'import-default'
# - Name=default  Prefix=0.0.0.0/0  Protocol=BGP  Action=Accept
.
# Create a BGP Group with:
# - Name=corp-it0
# - Type=External
# - Network (VLAN)=VLAN1099
# - BFD interval=1000
# - Local AS=64911
# - Hold Time=90
# - Set Export=export-vrfs and Import=import-default
# Add also the following Neighbor
# - IP_Address=10.255.224.0  Neighbor_AS=64901  Hold-Time=90
.
# Create a BGP Group with:
# - Name=developers0
# - Type=External
# - Network (VLAN)=VLAN1088
# - BFD interval=1000
# - Local AS=64911
# - Hold Time=90
# - Set Export=export-vrfs and Import=import-default
# Add also the following Neighbor
# - IP_Address=10.255.224.2  Neighbor_AS=64901  Hold-Time=90
.
# Create a BGP Group with:
# - Name=guest-wifi0
# - Type=External
# - Network (VLAN)=VLAN1033
# - BFD interval=1000
# - Local AS=64911
# - Hold Time=90
# - Set Export=export-vrfs and Import=import-default
# Add also the following Neighbor
# - IP_Address=10.255.224.4  Neighbor_AS=64901  Hold-Time=90
```

Here are screenshots after the configuration has been done:

Figure 60: service1 Additional IP Configuration

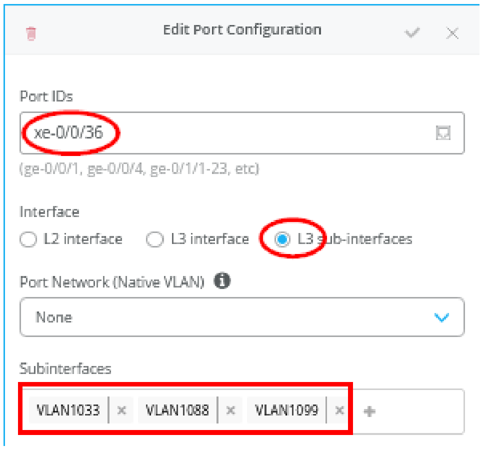


Additional IP Configuration

VLAN1033 1033	10.255.226.5 ›
VLAN1088 1088	10.255.226.3 ›
VLAN1099 1099	10.255.226.1 ›

.

Figure 61: service1 Uplink Interface



Edit Port Configuration

Port IDs

xe-0/0/36

(ge-0/0/1, ge-0/0/4, ge-0/1/1-23, etc)

Interface

○ L2 interface ○ L3 interface ● L3 sub-interfaces

Port Network (Native VLAN)

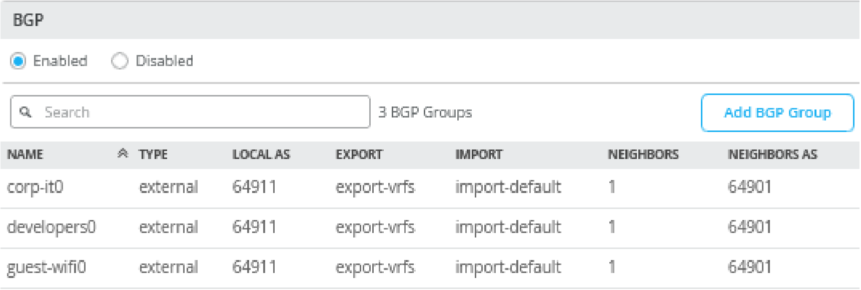
None

Subinterfaces

VLAN1033 × VLAN1088 × VLAN1099 × +

.

Figure 62: service1 BGP Summary



BGP

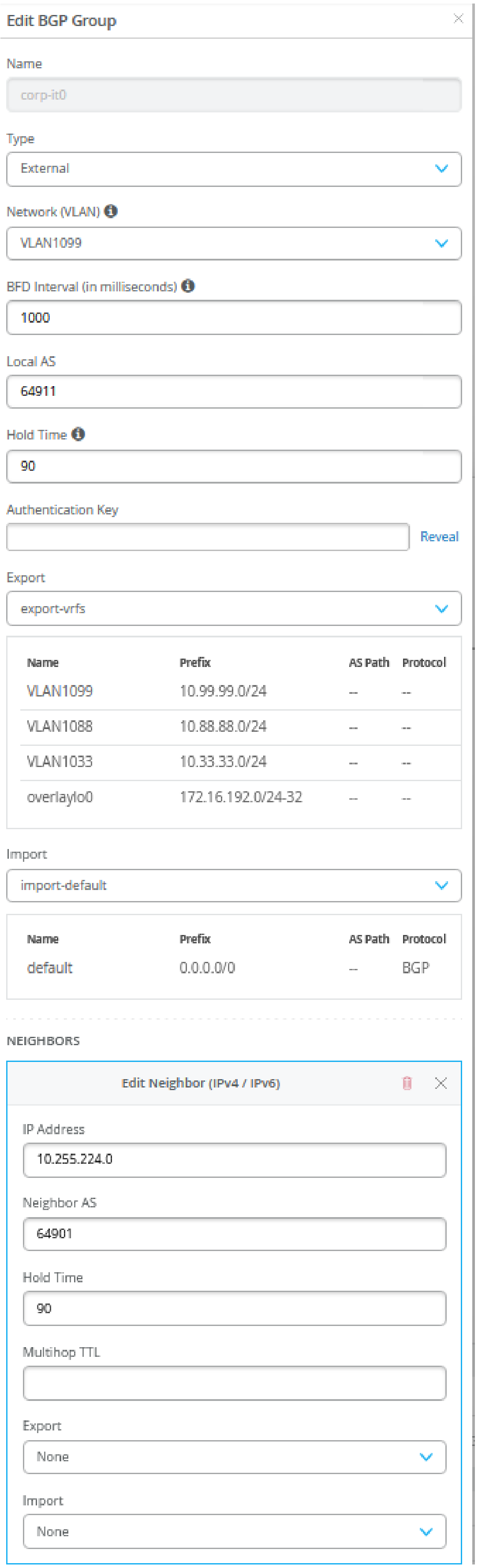
● Enabled ○ Disabled

Search 3 BGP Groups Add BGP Group

NAME	TYPE	LOCAL AS	EXPORT	IMPORT	NEIGHBORS	NEIGHBORS AS
corp-it0	external	64911	export-vrfs	import-default	1	64901
developers0	external	64911	export-vrfs	import-default	1	64901
guest-wifi0	external	64911	export-vrfs	import-default	1	64901

.

Figure 63: service1 First BGP Peering Group



Edit BGP Group

Name
corp-it0

Type
External

Network (VLAN) ℹ
VLAN1099

BFD Interval (in milliseconds) ℹ
1000

Local AS
64911

Hold Time ℹ
90

Authentication Key

Export
export-vrfs

Name	Prefix	AS Path	Protocol
VLAN1099	10.99.99.0/24	--	--
VLAN1088	10.88.88.0/24	--	--
VLAN1033	10.33.33.0/24	--	--
overlaylo0	172.16.192.0/24-32	--	--

Import
import-default

Name	Prefix	AS Path	Protocol
default	0.0.0.0/0	--	BGP

NEIGHBORS

Edit Neighbor (IPv4 / IPv6)

IP Address
10.255.224.0

Neighbor AS
64901

Hold Time
90

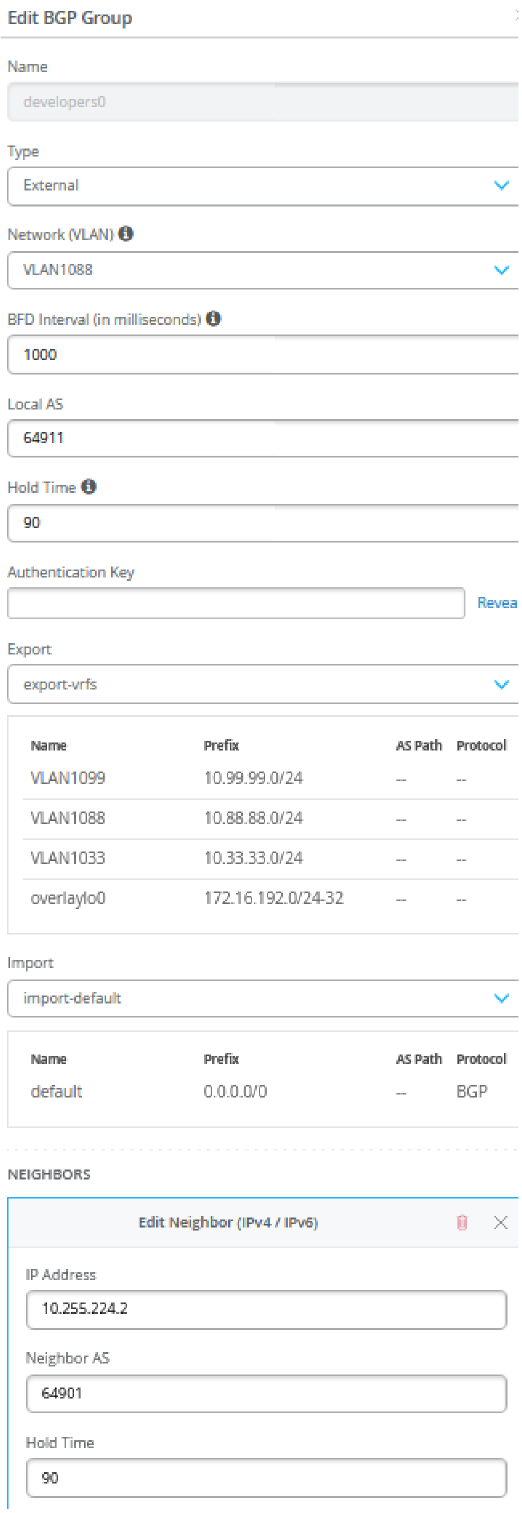
Multihop TTL

Export
None

Import
None

.

Figure 64: service1 Second BGP Peering Group



Edit BGP Group

Name

developers0

Type

External

Network (VLAN)

VLAN1088

BFD Interval (in milliseconds)

1000

Local AS

64911

Hold Time

90

Authentication Key

Revea

Export

export-vrfs

Name	Prefix	AS Path	Protocol
VLAN1099	10.99.99.0/24	--	--
VLAN1088	10.88.88.0/24	--	--
VLAN1033	10.33.33.0/24	--	--
overlaylo0	172.16.192.0/24-32	--	--

Import

import-default

Name	Prefix	AS Path	Protocol
default	0.0.0.0/0	--	BGP

NEIGHBORS

Edit Neighbor (IPv4 / IPv6)

IP Address

10.255.224.2

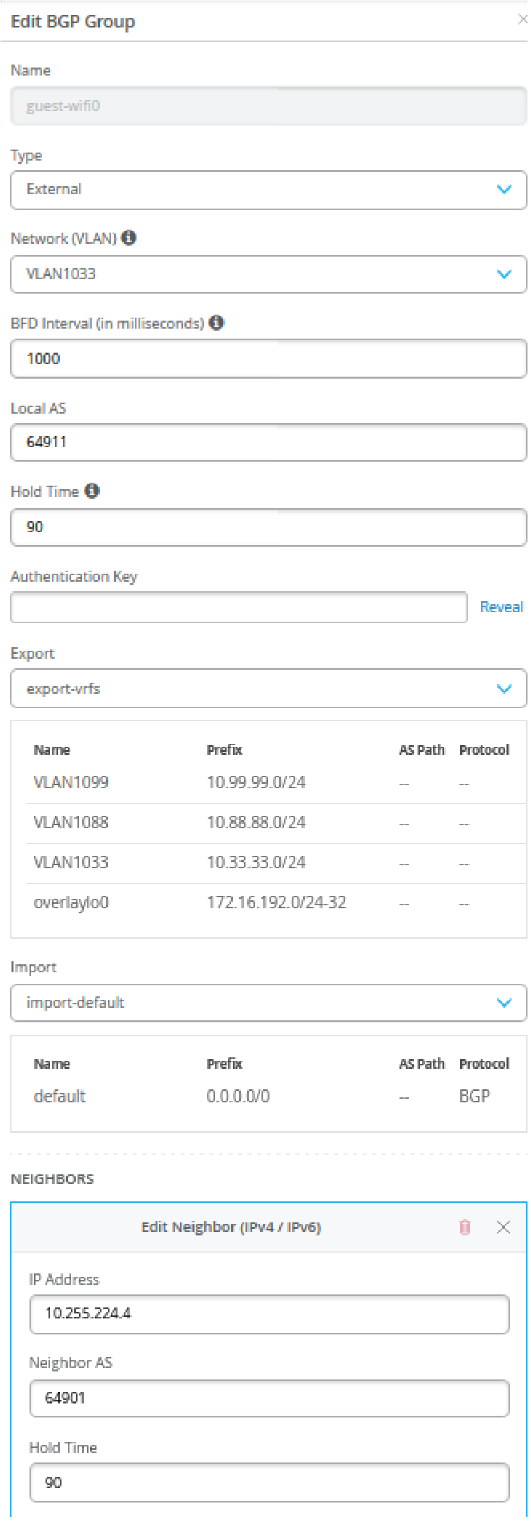
Neighbor AS

64901

Hold Time

90

Figure 65: service1 Third BGP Peering Group



Edit BGP Group

Name

guest-wifi0

Type

External

Network (VLAN)

VLAN1033

BFD Interval (in milliseconds)

1000

Local AS

64911

Hold Time

90

Authentication Key

Reveal

Export

export-vrfs

Name	Prefix	AS Path	Protocol
VLAN1099	10.99.99.0/24	--	--
VLAN1088	10.88.88.0/24	--	--
VLAN1033	10.33.33.0/24	--	--
overlaylo0	172.16.192.0/24-32	--	--

Import

import-default

Name	Prefix	AS Path	Protocol
default	0.0.0.0/0	--	BGP

NEIGHBORS

Edit Neighbor (IPv4 / IPv6)

IP Address

10.255.224.4

Neighbor AS

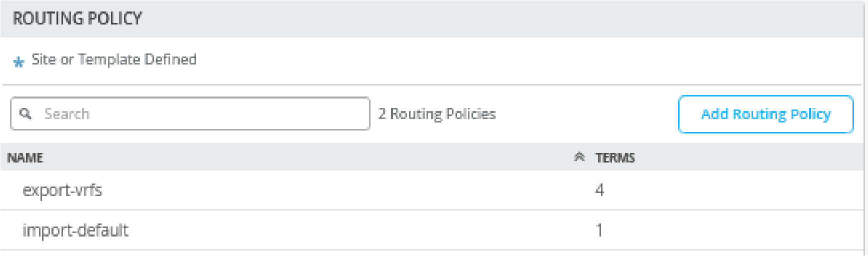
64901

Hold Time

90

.

Figure 66: service1 Routing Policy Summary



ROUTING POLICY

Site or Template Defined

2 Routing Policies

NAME	TERMS
export-vrfs	4
import-default	1

Configuration of service2 Block Switch for WAN Router Integration

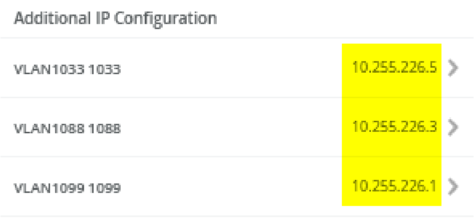
We have a verbal description of what needs to be configured on this system here:

```
# configure the Additional IP-Subnet 10.255.226.1 255.255.255.254 to Network/VLAN:VLAN1099
# configure the Additional IP-Subnet 10.255.226.3 255.255.255.254 to Network/VLAN:VLAN1088
# configure the Additional IP-Subnet 10.255.226.5 255.255.255.254 to Network/VLAN:VLAN1033
# Then bind these 3 Network/VLANs to Port Interface xe-0/0/36 as L3-Sub-Interfaces with MTU=9018
.
# Enable BGP and create an Export policy called 'export-vrfs'
# Add to this export Policy the following Networks as:
# - Add Term w. Name=VLAN1099  Prefix=10.99.99.0/24  Protocol=None  Then=Accept
# - Add Term w. Name=VLAN1088  Prefix=10.88.88.0/24  Protocol=None  Then=Accept
# - Add Term w. Name=VLAN1033  Prefix=10.33.33.0/24  Protocol=None  Then=Accept
# - Add Term w. Name=overlaylo0 Prefix=172.16.192.0/24-32  Protocol=None  Then=Accept
.
# Create an Export policy called 'import-default'
# - Name=default  Prefix=0.0.0.0/0  Protocol=BGP  Action=Accept
.
# Create a BGP Group with:
# - Name=corp-it0
# - Type=External
# - Network (VLAN)=VLAN1099
# - BFD interval=1000
# - Local AS=64911
# - Hold Time=90
# - Set Export=export-vrfs and Import=import-default
# Add also the following Neighbor
# - IP_Address=10.255.226.0  Neighbor_AS=64901  Hold-Time=90
.
```

```
# Create a BGP Group with:
# - Name=developers0
# - Type=External
# - Network (VLAN)=VLAN1088
# - BFD interval=1000
# - Local AS=64911
# - Hold Time=90
# - Set Export=export-vrfs and Import=import-default
# Add also the following Neighbor
# - IP_Address=10.255.226.2  Neighbor_AS=64901  Hold-Time=90
.
# Create a BGP Group with:
# - Name=guest-wifi0
# - Type=External
# - Network (VLAN)=VLAN1033
# - BFD interval=1000
# - Local AS=64911
# - Hold Time=90
# - Set Export=export-vrfs and Import=import-default
# Add also the following Neighbor
# - IP_Address=10.255.226.4  Neighbor_AS=64901  Hold-Time=90
```

Here are screenshots after the configuration is done:

Figure 67: service2 Additional IP Configuration

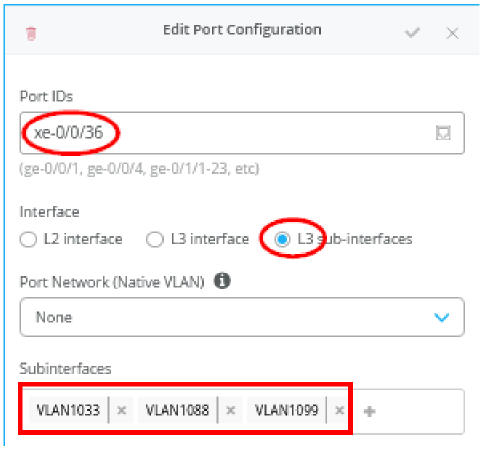


Additional IP Configuration

VLAN1033 1033	10.255.226.5
VLAN1088 1088	10.255.226.3
VLAN1099 1099	10.255.226.1

.

Figure 68: service2 Uplink Interface



Edit Port Configuration

Port IDs

xe-0/0/36

(ge-0/0/1, ge-0/0/4, ge-0/1/1-23, etc)

Interface

○ L2 interface ○ L3 interface ● L3 sub-interfaces

Port Network (Native VLAN)

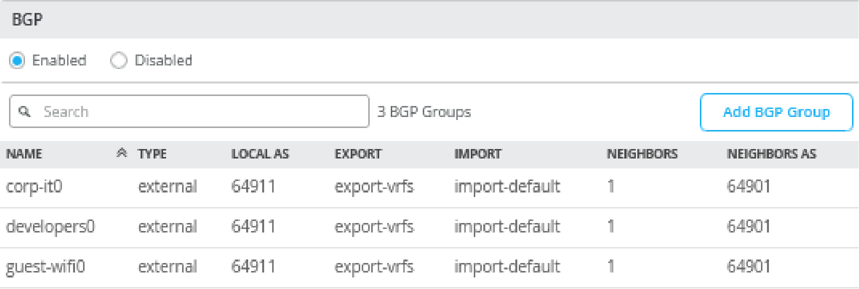
None

Subinterfaces

VLAN1033 × VLAN1088 × VLAN1099 ×

.

Figure 69: service2 BGP Summary



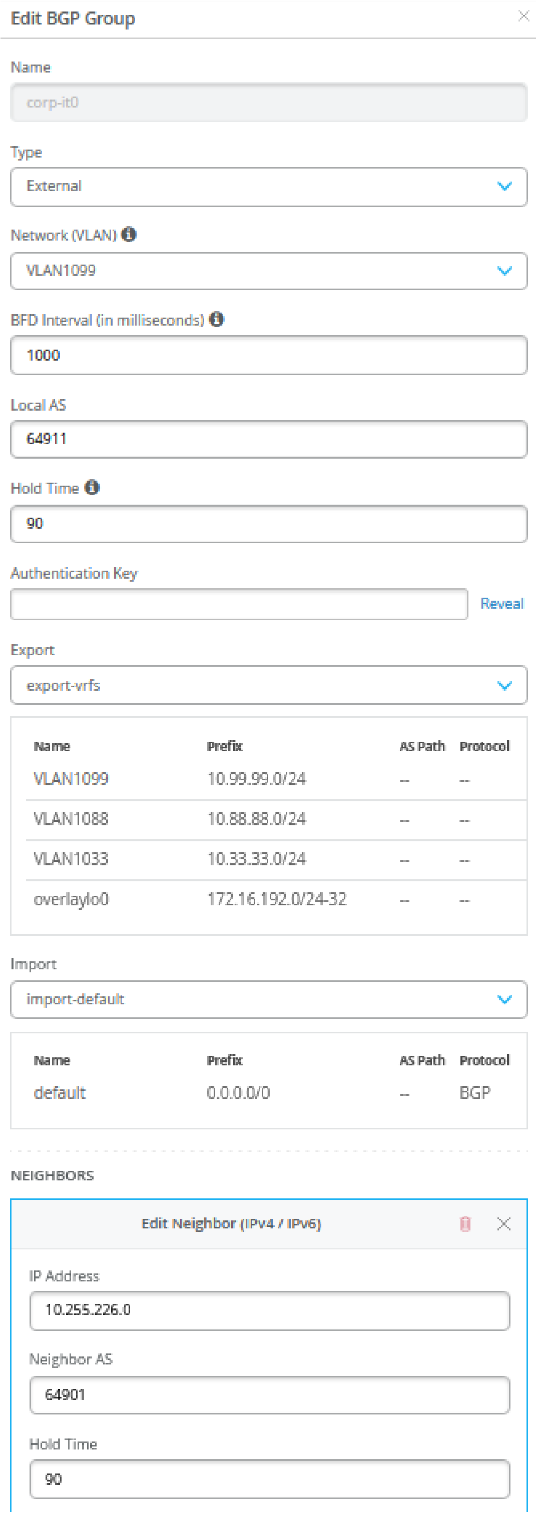
BGP

● Enabled ○ Disabled

3 BGP Groups

NAME	TYPE	LOCAL AS	EXPORT	IMPORT	NEIGHBORS	NEIGHBORS AS
corp-it0	external	64911	export-vrfs	import-default	1	64901
developers0	external	64911	export-vrfs	import-default	1	64901
guest-wifi0	external	64911	export-vrfs	import-default	1	64901

Figure 70: service2 First BGP Peering Group



Edit BGP Group

Name: corp-it0

Type: External

Network (VLAN): VLAN1099

BFD Interval (in milliseconds): 1000

Local AS: 64911

Hold Time: 90

Authentication Key:

Reveal

Export: export-vrfs

Name	Prefix	AS Path	Protocol
VLAN1099	10.99.99.0/24	--	--
VLAN1088	10.88.88.0/24	--	--
VLAN1033	10.33.33.0/24	--	--
overlaylo0	172.16.192.0/24-32	--	--

Import: import-default

Name	Prefix	AS Path	Protocol
default	0.0.0.0/0	--	BGP

NEIGHBORS

Edit Neighbor (IPv4 / IPv6)

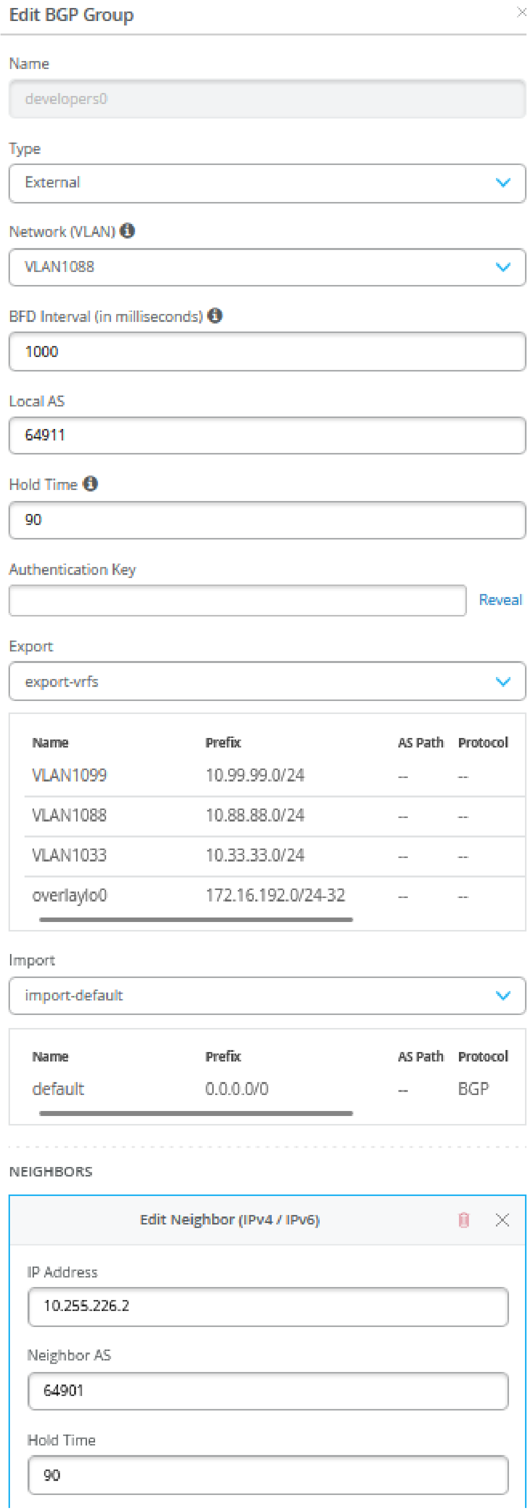
IP Address: 10.255.226.0

Neighbor AS: 64901

Hold Time: 90

.

Figure 71: service2 Second BGP Peering Group



Edit BGP Group

Name

developers0

Type

External

Network (VLAN)

VLAN1088

BFD Interval (in milliseconds)

1000

Local AS

64911

Hold Time

90

Authentication Key

Reveal

Export

export-vrfs

Name	Prefix	AS Path	Protocol
VLAN1099	10.99.99.0/24	--	--
VLAN1088	10.88.88.0/24	--	--
VLAN1033	10.33.33.0/24	--	--
overlaylo0	172.16.192.0/24-32	--	--

Import

import-default

Name	Prefix	AS Path	Protocol
default	0.0.0.0/0	--	BGP

NEIGHBORS

Edit Neighbor (IPv4 / IPv6)

IP Address

10.255.226.2

Neighbor AS

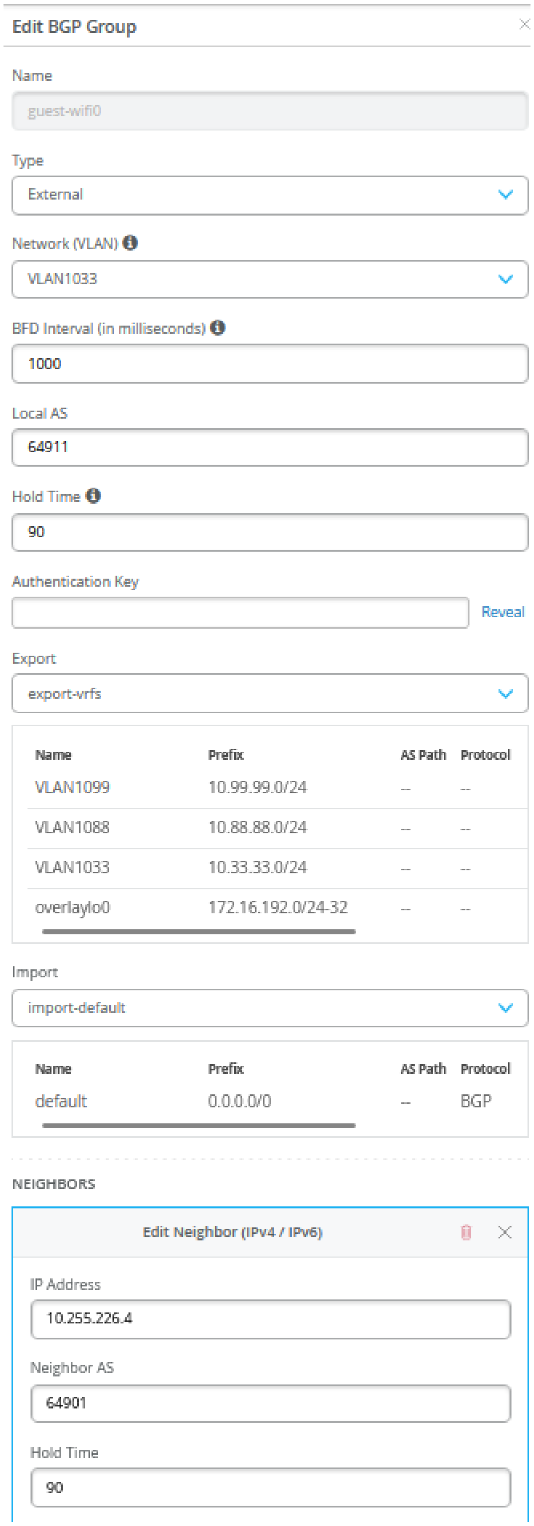
64901

Hold Time

90

.

Figure 72: service2 Third BGP Peering Group



Edit BGP Group

Name

guest-wifi0

Type

External

Network (VLAN)

VLAN1033

BFD Interval (in milliseconds)

1000

Local AS

64911

Hold Time

90

Authentication Key

Export

export-vrfs

Name	Prefix	AS Path	Protocol
VLAN1099	10.99.99.0/24	--	--
VLAN1088	10.88.88.0/24	--	--
VLAN1033	10.33.33.0/24	--	--
overlaylo0	172.16.192.0/24-32	--	--

Import

import-default

Name	Prefix	AS Path	Protocol
default	0.0.0.0/0	--	BGP

NEIGHBORS

Edit Neighbor (IPv4 / IPv6)

IP Address

10.255.226.4

Neighbor AS

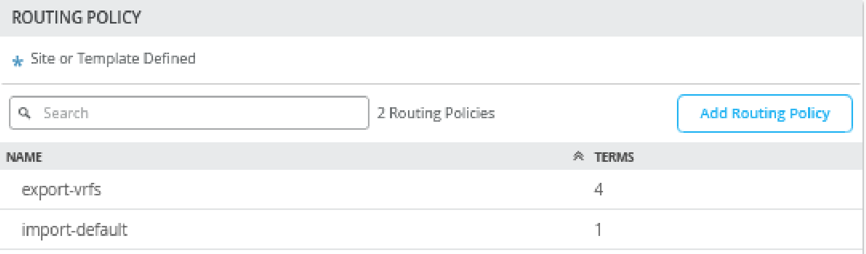
64901

Hold Time

90

.

Figure 73: service2 Third BGP Peering Group



ROUTING POLICY

* Site or Template Defined

NAME	TERMS
export-vrfs	4
import-default	1

Verification Between Service Block Switches and WAN-Router After Integration

NOTE: This step assumes you have also configured the WAN-Router to bring up the BGP route exchanges.

Service1 Switch:

Service1 switch must have established BGP peering with all three peers and obtained a default route from WAN-Router for each of the three VRF’s.

```
root@service1> show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 5 Peers: 7 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0
                      13         11          0          0          0          0
bgp.evpn.0
                     102         56          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
10.255.224.0          64901        141        143       0       1     1:02:40 Establ
  corp-it.inet.0: 1/1/1/0
10.255.224.2          64901        140        141       0       1     1:02:36 Establ
  developers.inet.0: 1/1/1/0
10.255.224.4          64901        140        141       0       1     1:02:32 Establ
  guest-wifi.inet.0: 1/1/1/0
```

```
10.255.240.3          65003       3110       3139       0       0    23:45:45 Establ
  inet.0: 6/7/7/0
10.255.240.5          65004       3127       3148       0       0    23:45:44 Establ
  inet.0: 5/6/6/0
172.16.254.3          65003       5337       5143       0       0    23:45:44 Establ
  bgp.evpn.0: 30/56/56/0
  default-switch.evpn.0: 10/21/21/0
  __default_evpn__.evpn.0: 0/0/0/0
  guest-wifi.evpn.0: 10/10/10/0
  developers.evpn.0: 12/12/12/0
  corp-it.evpn.0: 13/13/13/0
172.16.254.4          65004       6069       4393       0       0    23:45:43 Establ
  bgp.evpn.0: 26/46/46/0
  default-switch.evpn.0: 11/21/21/0
  __default_evpn__.evpn.0: 0/0/0/0
  guest-wifi.evpn.0: 0/7/7/0
  developers.evpn.0: 0/9/9/0
  corp-it.evpn.0: 0/9/9/0
.
root@service1> show bfd session
                                                  Detect   Transmit
Address                  State     Interface      Time     Interval  Multiplier
10.255.224.0             Up        xe-0/0/36.1099 3.000     1.000        3
10.255.224.2             Up        xe-0/0/36.1088 3.000     1.000        3
10.255.224.4             Up        xe-0/0/36.1033 3.000     1.000        3
10.255.240.3             Up        et-0/0/52.0    3.000     1.000        3
10.255.240.5             Up        et-0/0/53.0    3.000     1.000        3
172.16.254.3             Up                       3.000     1.000        3
172.16.254.4             Up                       3.000     1.000        3
.
7 sessions, 7 clients
Cumulative transmit rate 7.0 pps, cumulative receive rate 7.0 pps
.
root@service1> show route table corp-it.inet.0
corp-it.inet.0: 12 destinations, 15 routes (12 active, 0 holddown, 0 hidden)
@ = Routing Use Only, # = Forwarding Use Only
+ = Active Route, - = Last Active, * = Both
.
0.0.0.0/0          *[BGP/170] 01:00:16, localpref 100
                      AS path: 64901 I, validation-state: unverified
                    >  to 10.255.224.0 via xe-0/0/36.1099
                    [EVPN/170] 01:00:15
                    >  to 10.255.240.3 via et-0/0/52.0
```

```
10.99.99.0/24      @[EVPN/170] 02:22:25
                      to 10.255.240.3 via et-0/0/52.0
                    > to 10.255.240.5 via et-0/0/53.0
                    [EVPN/170] 02:22:25
                      to 10.255.240.3 via et-0/0/52.0
                    > to 10.255.240.5 via et-0/0/53.0
                   #[Multipath/255] 02:22:25, metric2 0
                      to 10.255.240.3 via et-0/0/52.0
                    > to 10.255.240.5 via et-0/0/53.0
                      to 10.255.240.3 via et-0/0/52.0
                    > to 10.255.240.5 via et-0/0/53.0
10.99.99.23/32     *[EVPN/170] 02:22:25
                      to 10.255.240.3 via et-0/0/52.0
                    > to 10.255.240.5 via et-0/0/53.0
10.99.99.42/32     *[EVPN/170] 02:22:25
                    > to 10.255.240.3 via et-0/0/52.0
                      to 10.255.240.5 via et-0/0/53.0
10.99.99.99/32     *[EVPN/170] 02:22:25
                      to 10.255.240.3 via et-0/0/52.0
                    > to 10.255.240.5 via et-0/0/53.0
10.255.224.0/31    *[Direct/0] 02:22:25
                    > via xe-0/0/36.1099
10.255.224.1/32    *[Local/0] 02:22:25
                      Local via xe-0/0/36.1099
10.255.226.0/31    *[EVPN/170] 02:22:21
                    > to 10.255.240.3 via et-0/0/52.0
172.16.192.1/32    *[Direct/0] 02:22:25
                    > via lo0.1
172.16.192.4/32    *[EVPN/170] 02:22:25
                      to 10.255.240.3 via et-0/0/52.0
                    > to 10.255.240.5 via et-0/0/53.0
172.16.192.7/32    *[EVPN/170] 02:22:25
                      to 10.255.240.3 via et-0/0/52.0
                    > to 10.255.240.5 via et-0/0/53.0
172.16.192.10/32   *[EVPN/170] 02:22:21
                    > to 10.255.240.3 via et-0/0/52.0
.
root@service1> show route table developers.inet.0
developers.inet.0: 11 destinations, 14 routes (11 active, 0 holddown, 0 hidden)
@ = Routing Use Only, # = Forwarding Use Only
+ = Active Route, - = Last Active, * = Both
.
0.0.0.0/0          *[BGP/170] 01:00:32, localpref 100
```

```
                      AS path: 64901 I, validation-state: unverified
                    >  to 10.255.224.2 via xe-0/0/36.1088
                    [EVPN/170] 01:00:31
                    >  to 10.255.240.3 via et-0/0/52.0
10.88.88.0/24      @[EVPN/170] 02:22:45
                       to 10.255.240.3 via et-0/0/52.0
                    >  to 10.255.240.5 via et-0/0/53.0
                    [EVPN/170] 02:22:45
                       to 10.255.240.3 via et-0/0/52.0
                    >  to 10.255.240.5 via et-0/0/53.0
                   #[Multipath/255] 02:22:45, metric2 0
                       to 10.255.240.3 via et-0/0/52.0
                    >  to 10.255.240.5 via et-0/0/53.0
                       to 10.255.240.3 via et-0/0/52.0
                    >  to 10.255.240.5 via et-0/0/53.0
10.88.88.23/32     *[EVPN/170] 02:22:45
                       to 10.255.240.3 via et-0/0/52.0
                    >  to 10.255.240.5 via et-0/0/53.0
10.88.88.88/32     *[EVPN/170] 02:22:45
                    >  to 10.255.240.3 via et-0/0/52.0
                       to 10.255.240.5 via et-0/0/53.0
10.255.224.2/31    *[Direct/0] 02:22:45
                    >  via xe-0/0/36.1088
10.255.224.3/32    *[Local/0] 02:22:45
                       Local via xe-0/0/36.1088
10.255.226.2/31    *[EVPN/170] 02:22:41
                    >  to 10.255.240.3 via et-0/0/52.0
172.16.192.2/32    *[Direct/0] 02:22:45
                    >  via lo0.2
172.16.192.5/32    *[EVPN/170] 02:22:45
                       to 10.255.240.3 via et-0/0/52.0
                    >  to 10.255.240.5 via et-0/0/53.0
172.16.192.8/32    *[EVPN/170] 02:22:45
                    >  to 10.255.240.3 via et-0/0/52.0
                       to 10.255.240.5 via et-0/0/53.0
172.16.192.11/32   *[EVPN/170] 02:22:41
                    >  to 10.255.240.3 via et-0/0/52.0
.
root@service1> show route table guest-wifi.inet.0
guest-wifi.inet.0: 9 destinations, 12 routes (9 active, 0 holddown, 0 hidden)
@ = Routing Use Only, # = Forwarding Use Only
+ = Active Route, - = Last Active, * = Both
.
```

```
0.0.0.0/0          *[BGP/170] 01:00:45, localpref 100
                      AS path: 64901 I, validation-state: unverified
                    >  to 10.255.224.4 via xe-0/0/36.1033
                    [EVPN/170] 01:00:44
                    >  to 10.255.240.3 via et-0/0/52.0
10.33.33.0/24      @[EVPN/170] 02:23:02
                       to 10.255.240.3 via et-0/0/52.0
                    >  to 10.255.240.5 via et-0/0/53.0
                    [EVPN/170] 02:23:02
                       to 10.255.240.3 via et-0/0/52.0
                    >  to 10.255.240.5 via et-0/0/53.0
                   #[Multipath/255] 02:23:02, metric2 0
                       to 10.255.240.3 via et-0/0/52.0
                    >  to 10.255.240.5 via et-0/0/53.0
                       to 10.255.240.3 via et-0/0/52.0
                    >  to 10.255.240.5 via et-0/0/53.0
10.255.224.4/31    *[Direct/0] 02:23:02
                    >  via xe-0/0/36.1033
10.255.224.5/32    *[Local/0] 02:23:02
                       Local via xe-0/0/36.1033
10.255.226.4/31    *[EVPN/170] 02:22:58
                    >  to 10.255.240.3 via et-0/0/52.0
172.16.192.3/32    *[Direct/0] 02:23:02
                    >  via lo0.3
172.16.192.6/32    *[EVPN/170] 02:23:02
                       to 10.255.240.3 via et-0/0/52.0
                    >  to 10.255.240.5 via et-0/0/53.0
172.16.192.9/32    *[EVPN/170] 02:23:02
                    >  to 10.255.240.3 via et-0/0/52.0
                       to 10.255.240.5 via et-0/0/53.0
172.16.192.12/32   *[EVPN/170] 02:22:58
                    >  to 10.255.240.3 via et-0/0/52.0
```

Service2 Switch:

Service2 switch must have established BGP peering with all three peers and obtained a default route from WAN-Router for each of the three VRFs.

NOTE: Remember that we have a simulated broken Link still between service2 and core2. That is what we lesser routes and bfd sessions.

```
root@service2> show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 5 Peers: 7 Down peers: 2
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0
                       7          7          0          0          0          0
bgp.evpn.0
                      57         57          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
10.255.226.0          64901        160        163       0       1     1:11:18 Establ
  corp-it.inet.0: 1/1/1/0
10.255.226.2          64901        159        161       0       1     1:11:14 Establ
  developers.inet.0: 1/1/1/0
10.255.226.4          64901        159        162       0       1     1:11:10 Establ
  guest-wifi.inet.0: 1/1/1/0
10.255.240.7          65003       3129       3179       0       0    23:54:22 Establ
  inet.0: 7/7/7/0
10.255.240.9          65004          0          0       0       0    23:55:54 Idle
172.16.254.3          65003       6035       3203       0       0    23:54:20 Establ
  bgp.evpn.0: 57/57/57/0
  default-switch.evpn.0: 22/22/22/0
  __default_evpn__.evpn.0: 0/0/0/0
  guest-wifi.evpn.0: 10/10/10/0
  developers.evpn.0: 12/12/12/0
  corp-it.evpn.0: 13/13/13/0
172.16.254.4          65004          0          0       0       0    23:55:54 Active
.
root@service2> show bfd session
                                                  Detect   Transmit
Address                  State     Interface      Time     Interval  Multiplier
10.255.226.0             Up        xe-0/0/36.1099 3.000     1.000        3
10.255.226.2             Up        xe-0/0/36.1088 3.000     1.000        3
10.255.226.4             Up        xe-0/0/36.1033 3.000     1.000        3
10.255.240.7             Up        et-0/0/53.0    3.000     1.000        3
```

```
172.16.254.3             Up                       3.000     1.000        3
.
5 sessions, 5 clients
Cumulative transmit rate 5.0 pps, cumulative receive rate 5.0 pps
.
root@service2> show route table corp-it.inet.0
corp-it.inet.0: 12 destinations, 15 routes (12 active, 0 holddown, 0 hidden)
@ = Routing Use Only, # = Forwarding Use Only
+ = Active Route, - = Last Active, * = Both
.
0.0.0.0/0          *[BGP/170] 01:12:36, localpref 100
                      AS path: 64901 I, validation-state: unverified
                    >  to 10.255.226.0 via xe-0/0/36.1099
                    [EVPN/170] 01:12:36
                    >  to 10.255.240.7 via et-0/0/53.0
10.99.99.0/24      @[EVPN/170] 02:34:43
                    >  to 10.255.240.7 via et-0/0/53.0
                    [EVPN/170] 02:34:43
                    >  to 10.255.240.7 via et-0/0/53.0
                   #[Multipath/255] 02:34:43, metric2 0
                    >  to 10.255.240.7 via et-0/0/53.0
                    >  to 10.255.240.7 via et-0/0/53.0
10.99.99.23/32     *[EVPN/170] 02:34:43
                    >  to 10.255.240.7 via et-0/0/53.0
10.99.99.42/32     *[EVPN/170] 02:34:43
                    >  to 10.255.240.7 via et-0/0/53.0
10.99.99.99/32     *[EVPN/170] 02:34:43
                    >  to 10.255.240.7 via et-0/0/53.0
10.255.224.0/31    *[EVPN/170] 02:34:43
                    >  to 10.255.240.7 via et-0/0/53.0
10.255.226.0/31    *[Direct/0] 02:34:42
                    >  via xe-0/0/36.1099
10.255.226.1/32    *[Local/0] 02:34:42
                       Local via xe-0/0/36.1099
172.16.192.1/32    *[EVPN/170] 02:34:43
                    >  to 10.255.240.7 via et-0/0/53.0
172.16.192.4/32    *[EVPN/170] 02:34:43
                    >  to 10.255.240.7 via et-0/0/53.0
172.16.192.7/32    *[EVPN/170] 02:34:43
                    >  to 10.255.240.7 via et-0/0/53.0
172.16.192.10/32   *[Direct/0] 02:34:42
                    >  via lo0.1
.
```

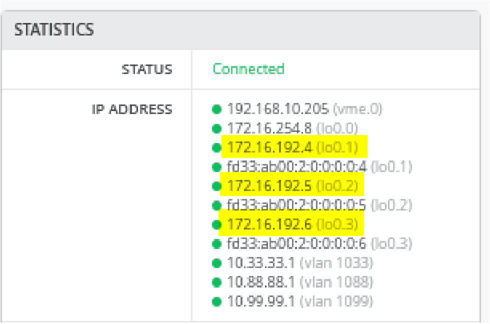
```
root@service2> show route table developers.inet.0
developers.inet.0: 11 destinations, 14 routes (11 active, 0 holddown, 0 hidden)
@ = Routing Use Only, # = Forwarding Use Only
+ = Active Route, - = Last Active, * = Both
.
0.0.0.0/0          *[BGP/170] 01:12:45, localpref 100
                      AS path: 64901 I, validation-state: unverified
                    >  to 10.255.226.2 via xe-0/0/36.1088
                    [EVPN/170] 01:12:45
                    >  to 10.255.240.7 via et-0/0/53.0
10.88.88.0/24      @[EVPN/170] 02:34:56
                    >  to 10.255.240.7 via et-0/0/53.0
                    [EVPN/170] 02:34:56
                    >  to 10.255.240.7 via et-0/0/53.0
                   #[Multipath/255] 02:34:56, metric2 0
                    >  to 10.255.240.7 via et-0/0/53.0
                    >  to 10.255.240.7 via et-0/0/53.0
10.88.88.23/32     *[EVPN/170] 02:34:56
                    >  to 10.255.240.7 via et-0/0/53.0
10.88.88.88/32     *[EVPN/170] 02:34:56
                    >  to 10.255.240.7 via et-0/0/53.0
10.255.224.2/31    *[EVPN/170] 02:34:56
                    >  to 10.255.240.7 via et-0/0/53.0
10.255.226.2/31    *[Direct/0] 02:34:55
                    >  via xe-0/0/36.1088
10.255.226.3/32    *[Local/0] 02:34:55
                       Local via xe-0/0/36.1088
172.16.192.2/32    *[EVPN/170] 02:34:56
                    >  to 10.255.240.7 via et-0/0/53.0
172.16.192.5/32    *[EVPN/170] 02:34:56
                    >  to 10.255.240.7 via et-0/0/53.0
172.16.192.8/32    *[EVPN/170] 02:34:56
                    >  to 10.255.240.7 via et-0/0/53.0
172.16.192.11/32   *[Direct/0] 02:34:55
                    >  via lo0.2
.
root@service2> show route table guest-wifi.inet.0
guest-wifi.inet.0: 9 destinations, 12 routes (9 active, 0 holddown, 0 hidden)
@ = Routing Use Only, # = Forwarding Use Only
+ = Active Route, - = Last Active, * = Both
.
0.0.0.0/0          *[BGP/170] 01:12:57, localpref 100
                      AS path: 64901 I, validation-state: unverified
                    >  to 10.255.226.4 via xe-0/0/36.1033
                    [EVPN/170] 01:12:57
                    >  to 10.255.240.7 via et-0/0/53.0
10.33.33.0/24      @[EVPN/170] 02:35:12
                    >  to 10.255.240.7 via et-0/0/53.0
                    [EVPN/170] 02:35:12
                    >  to 10.255.240.7 via et-0/0/53.0
                   #[Multipath/255] 02:35:12, metric2 0
                    >  to 10.255.240.7 via et-0/0/53.0
                    >  to 10.255.240.7 via et-0/0/53.0
10.255.224.4/31    *[EVPN/170] 02:35:12
                    >  to 10.255.240.7 via et-0/0/53.0
10.255.226.4/31    *[Direct/0] 02:35:11
                    >  via xe-0/0/36.1033
10.255.226.5/32    *[Local/0] 02:35:11
                       Local via xe-0/0/36.1033
172.16.192.3/32    *[EVPN/170] 02:35:12
                    >  to 10.255.240.7 via et-0/0/53.0
172.16.192.6/32    *[EVPN/170] 02:35:12
                    >  to 10.255.240.7 via et-0/0/53.0
172.16.192.9/32    *[EVPN/170] 02:35:12
                    >  to 10.255.240.7 via et-0/0/53.0
172.16.192.12/32   *[Direct/0] 02:35:11
                    >  via lo0.3
```

WAN-Router:

Below we captured the BGP and BFD summary as well as the routes known to the device. Important here are:

- Route `10.99.99.0/24` from overlay=`VLAN1099` assigned to VRF=`corp-it`
- Route `10.88.88.0/24` from overlay=`VLAN1088` assigned to VRF=`developers`
- Route `10.33.33.0/24` from overlay=`VLAN1033` assigned to VRF=`guest-wifi`
- Route `172.16.192.4to9/32` from overlay for DHCP-Relay usage.

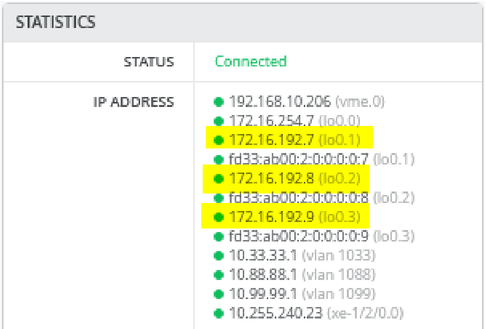
Figure 74: access1 Overlay Loopbacks for DHCP-Relay



STATISTICS	
STATUS	Connected
IP ADDRESS	● 192.168.10.205 (vme.0) ● 172.16.254.8 (lo0.0) ● 172.16.192.4 (lo0.1) ● fd33:ab00:2:0:0:0:0:4 (lo0.1) ● 172.16.192.5 (lo0.2) ● fd33:ab00:2:0:0:0:0:5 (lo0.2) ● 172.16.192.6 (lo0.3) ● fd33:ab00:2:0:0:0:0:6 (lo0.3) ● 10.33.33.1 (vlan 1033) ● 10.88.88.1 (vlan 1088) ● 10.99.99.1 (vlan 1099)

.

Figure 75: access2 Overlay Loopbacks for DHCP-Relay



STATISTICS	
STATUS	Connected
IP ADDRESS	● 192.168.10.206 (vme.0) ● 172.16.254.7 (lo0.0) ● 172.16.192.7 (lo0.1) ● fd33:ab00:2:0:0:0:0:7 (lo0.1) ● 172.16.192.8 (lo0.2) ● fd33:ab00:2:0:0:0:0:8 (lo0.2) ● 172.16.192.9 (lo0.3) ● fd33:ab00:2:0:0:0:0:9 (lo0.3) ● 10.33.33.1 (vlan 1033) ● 10.88.88.1 (vlan 1088) ● 10.99.99.1 (vlan 1099) ● 10.255.240.23 (xe-1/2/0.0)

Review the below information

```
root@wanrouter> show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 3 Peers: 6 Down peers: 0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
10.255.224.1          64911        145        140       0       2     1:02:47 Establ
  public-int.inet.0: 4/5/5/0
10.255.224.3          64911        144        140       0       2     1:02:51 Establ
  public-int.inet.0: 4/5/5/0
10.255.224.5          64911        143        140       0       6     1:02:46 Establ
  public-int.inet.0: 4/5/5/0
10.255.226.1          64911        145        140       0       2     1:02:44 Establ
  public-int.inet.0: 4/5/5/0
10.255.226.3          64911        145        140       0       2     1:02:48 Establ
```

```
  public-int.inet.0: 4/5/5/0
10.255.226.5          64911       144        141       0       6     1:02:51 Establ
  public-int.inet.0: 4/5/5/0
.
root@wanrouter> show bfd session
                                                  Detect   Transmit
Address                  State     Interface      Time     Interval  Multiplier
10.255.224.1             Up        xe-0/0/16.1099 3.000     1.000        3
10.255.224.3             Up        xe-0/0/16.1088 3.000     1.000        3
10.255.224.5             Up        xe-0/0/16.1033 3.000     1.000        3
10.255.226.1             Up        xe-0/0/17.1099 3.000     1.000        3
10.255.226.3             Up        xe-0/0/17.1088 3.000     1.000        3
10.255.226.5             Up        xe-0/0/17.1033 3.000     1.000        3
.
6 sessions, 6 clients
Cumulative transmit rate 6.0 pps, cumulative receive rate 6.0 pps
.
root@wanrouter> show route
              public-int.inet.0: 30 destinations, 45 routes (30 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
.
0.0.0.0/0          *[Static/5] 5d 23:56:46
                    >  to 192.168.10.1 via ge-0/0/0.0
10.33.33.0/24      *[BGP/170] 01:03:06, localpref 100, from 10.255.226.5
                      AS path: 64911 65002 65003 65006 65007 I, validation-state: unverified
                       to 10.255.226.5 via xe-0/0/17.1033
                    >  to 10.255.224.5 via xe-0/0/16.1033
                    [BGP/170] 01:03:16, localpref 100
                      AS path: 64911 65001 65003 65006 65007 I, validation-state: unverified
                    >  to 10.255.224.5 via xe-0/0/16.1033
10.88.88.0/24      *[BGP/170] 01:03:06, localpref 100, from 10.255.224.3
                      AS path: 64911 65001 65003 65006 65007 I, validation-state: unverified
                       to 10.255.224.3 via xe-0/0/16.1088
                    >  to 10.255.226.3 via xe-0/0/17.1088
                    [BGP/170] 01:03:18, localpref 100
                      AS path: 64911 65002 65003 65006 65007 I, validation-state: unverified
                    >  to 10.255.226.3 via xe-0/0/17.1088
10.99.99.0/24      *[BGP/170] 01:03:06, localpref 100, from 10.255.224.1
                      AS path: 64911 65001 65003 65006 65007 I, validation-state: unverified
                       to 10.255.224.1 via xe-0/0/16.1099
                    >  to 10.255.226.1 via xe-0/0/17.1099
                    [BGP/170] 01:03:14, localpref 100
                      AS path: 64911 65002 65003 65006 65007 I, validation-state: unverified
                    >  to 10.255.226.1 via xe-0/0/17.1099
10.255.224.0/31    *[Direct/0] 22:24:32
                    >  via xe-0/0/16.1099
10.255.224.0/32    *[Local/0] 22:24:32
                       Local via xe-0/0/16.1099
10.255.224.2/31    *[Direct/0] 22:24:32
                    >  via xe-0/0/16.1088
10.255.224.2/32    *[Local/0] 22:24:32
                       Local via xe-0/0/16.1088
10.255.224.4/31    *[Direct/0] 22:24:32
                    >  via xe-0/0/16.1033
10.255.224.4/32    *[Local/0] 22:24:32
                       Local via xe-0/0/16.1033
10.255.226.0/31    *[Direct/0] 22:24:32
                    >  via xe-0/0/17.1099
10.255.226.0/32    *[Local/0] 22:24:32
                       Local via xe-0/0/17.1099
10.255.226.2/31    *[Direct/0] 22:24:32
                    >  via xe-0/0/17.1088
10.255.226.2/32    *[Local/0] 22:24:32
                       Local via xe-0/0/17.1088
10.255.226.4/31    *[Direct/0] 22:24:32
                    >  via xe-0/0/17.1033
10.255.226.4/32    *[Local/0] 22:24:32
                       Local via xe-0/0/17.1033
172.16.192.1/32    *[BGP/170] 01:03:17, localpref 100
                      AS path: 64911 I, validation-state: unverified
                    >  to 10.255.224.1 via xe-0/0/16.1099
                    [BGP/170] 01:03:14, localpref 100
                      AS path: 64911 65002 65003 65001 I, validation-state: unverified
                    >  to 10.255.226.1 via xe-0/0/17.1099
172.16.192.2/32    *[BGP/170] 01:03:21, localpref 100
                      AS path: 64911 I, validation-state: unverified
                    >  to 10.255.224.3 via xe-0/0/16.1088
                    [BGP/170] 01:03:18, localpref 100
                      AS path: 64911 65002 65003 65001 I, validation-state: unverified
                    >  to 10.255.226.3 via xe-0/0/17.1088
172.16.192.3/32    *[BGP/170] 01:03:16, localpref 100
                      AS path: 64911 I, validation-state: unverified
                    >  to 10.255.224.5 via xe-0/0/16.1033
                    [BGP/170] 01:03:22, localpref 100
                      AS path: 64911 65002 65003 65001 I, validation-state: unverified
                    >  to 10.255.226.5 via xe-0/0/17.1033
```

```
172.16.192.4/32    *[BGP/170] 01:03:06, localpref 100, from 10.255.224.1
                      AS path: 64911 65001 65003 65005 65008 I, validation-state: unverified
                       to 10.255.224.1 via xe-0/0/16.1099
                    >  to 10.255.226.1 via xe-0/0/17.1099
                    [BGP/170] 01:03:14, localpref 100
                      AS path: 64911 65002 65003 65005 65008 I, validation-state: unverified
                    >  to 10.255.226.1 via xe-0/0/17.1099
172.16.192.5/32    *[BGP/170] 01:03:06, localpref 100, from 10.255.224.3
                      AS path: 64911 65001 65003 65005 65008 I, validation-state: unverified
                       to 10.255.224.3 via xe-0/0/16.1088
                    >  to 10.255.226.3 via xe-0/0/17.1088
                    [BGP/170] 01:03:18, localpref 100
                      AS path: 64911 65002 65003 65005 65008 I, validation-state: unverified
                    >  to 10.255.226.3 via xe-0/0/17.1088
172.16.192.6/32    *[BGP/170] 01:03:06, localpref 100, from 10.255.226.5
                      AS path: 64911 65002 65003 65006 65008 I, validation-state: unverified
                       to 10.255.226.5 via xe-0/0/17.1033
                    >  to 10.255.224.5 via xe-0/0/16.1033
                    [BGP/170] 01:03:16, localpref 100
                      AS path: 64911 65001 65003 65006 65008 I, validation-state: unverified
                    >  to 10.255.224.5 via xe-0/0/16.1033
172.16.192.7/32    *[BGP/170] 01:03:06, localpref 100, from 10.255.224.1
                      AS path: 64911 65001 65003 65005 65007 I, validation-state: unverified
                       to 10.255.224.1 via xe-0/0/16.1099
                    >  to 10.255.226.1 via xe-0/0/17.1099
                    [BGP/170] 01:03:14, localpref 100
                      AS path: 64911 65002 65003 65005 65007 I, validation-state: unverified
                    >  to 10.255.226.1 via xe-0/0/17.1099
172.16.192.8/32    *[BGP/170] 01:03:06, localpref 100
                      AS path: 64911 65001 65003 65006 65007 I, validation-state: unverified
                    >  to 10.255.224.3 via xe-0/0/16.1088
                       to 10.255.226.3 via xe-0/0/17.1088
                    [BGP/170] 01:03:18, localpref 100
                      AS path: 64911 65002 65003 65006 65007 I, validation-state: unverified
                    >  to 10.255.226.3 via xe-0/0/17.1088
172.16.192.9/32    *[BGP/170] 01:03:06, localpref 100
                      AS path: 64911 65002 65003 65006 65007 I, validation-state: unverified
                    >  to 10.255.226.5 via xe-0/0/17.1033
                       to 10.255.224.5 via xe-0/0/16.1033
                    [BGP/170] 01:03:16, localpref 100
                      AS path: 64911 65001 65003 65006 65007 I, validation-state: unverified
                    >  to 10.255.224.5 via xe-0/0/16.1033
172.16.192.10/32   *[BGP/170] 01:03:14, localpref 100
```

```
                      AS path: 64911 I, validation-state: unverified
                    >  to 10.255.226.1 via xe-0/0/17.1099
                    [BGP/170] 01:03:17, localpref 100
                      AS path: 64911 65001 65003 65002 I, validation-state: unverified
                    >  to 10.255.224.1 via xe-0/0/16.1099
172.16.192.11/32   *[BGP/170] 01:03:18, localpref 100
                      AS path: 64911 I, validation-state: unverified
                    >  to 10.255.226.3 via xe-0/0/17.1088
                    [BGP/170] 01:03:21, localpref 100
                      AS path: 64911 65001 65003 65002 I, validation-state: unverified
                    >  to 10.255.224.3 via xe-0/0/16.1088
172.16.192.12/32   *[BGP/170] 01:03:22, localpref 100
                      AS path: 64911 I, validation-state: unverified
                    >  to 10.255.226.5 via xe-0/0/17.1033
                    [BGP/170] 01:03:16, localpref 100
                      AS path: 64911 65001 65003 65002 I, validation-state: unverified
                    >  to 10.255.224.5 via xe-0/0/16.1033
192.168.10.0/24    *[Direct/0] 6d 00:00:22
                    >  via ge-0/0/0.0
192.168.10.99/32   *[Local/0] 6d 00:00:22
                       Local via ge-0/0/0.0
```

Fabric VRF Route Updates

Now that we exchange routes with the WAN-Router, all Access Switches should have default routes obtained via the service block switches. In our example we review VRF `corp-it` on access1 switch to see the difference between before and after WAN-Router integration.

```
root@access1> show route table corp-it.inet.0
corp-it.inet.0: 12 destinations, 15 routes (12 active, 0 holddown, 0 hidden)
@ = Routing Use Only, # = Forwarding Use Only
+ = Active Route, - = Last Active, * = Both
0.0.0.0/0          @[EVPN/170] 00:02:34
                    >  to 10.255.240.24 via mge-0/0/36.0
                       to 10.255.240.20 via mge-0/0/37.0
                    [EVPN/170] 00:02:30
                    >  to 10.255.240.24 via mge-0/0/36.0
                       to 10.255.240.20 via mge-0/0/37.0
                   #[Multipath/255] 00:02:30, metric2 0
```

```
                    >  to 10.255.240.24 via mge-0/0/36.0
                       to 10.255.240.20 via mge-0/0/37.0
                    >  to 10.255.240.24 via mge-0/0/36.0
                       to 10.255.240.20 via mge-0/0/37.0
10.99.99.0/24      *[Direct/0] 04:00:31
                    >  via irb.1099
                    [EVPN/170] 03:53:25
                       to 10.255.240.24 via mge-0/0/36.0
                    >  to 10.255.240.20 via mge-0/0/37.0
10.99.99.1/32      *[Local/0] 04:00:31
                       Local via irb.1099
10.99.99.23/32     *[EVPN/7] 03:47:48
                    >  via irb.1099
10.99.99.42/32     *[EVPN/170] 03:49:02
                    >  to 10.255.240.24 via mge-0/0/36.0
                       to 10.255.240.20 via mge-0/0/37.0
10.99.99.99/32     *[EVPN/7] 03:45:21
                    >  via irb.1099
10.255.224.0/31    *[EVPN/170] 00:02:44
                    >  to 10.255.240.24 via mge-0/0/36.0
                       to 10.255.240.20 via mge-0/0/37.0
10.255.226.0/31    *[EVPN/170] 00:02:41
                    >  to 10.255.240.24 via mge-0/0/36.0
                       to 10.255.240.20 via mge-0/0/37.0
172.16.192.1/32    *[EVPN/170] 00:02:44
                       to 10.255.240.24 via mge-0/0/36.0
                    >  to 10.255.240.20 via mge-0/0/37.0
172.16.192.4/32    *[Direct/0] 04:00:32
                    >  via lo0.1
172.16.192.7/32    *[EVPN/170] 03:53:25
                       to 10.255.240.24 via mge-0/0/36.0
                    >  to 10.255.240.20 via mge-0/0/37.0
172.16.192.10/32   *[EVPN/170] 00:02:41
                    >  to 10.255.240.24 via mge-0/0/36.0
                       to 10.255.240.20 via mge-0/0/37.0
```

Client Communication Verification Repeated

The final test now is to repeat the client communication verification that we have performed here: "Wired Client Verification" on page 72. In contrast we can now ping clients in other VRFs as the WAN router hair-pins this traffic, communication to Internet is now possible and we also can obtain DHCP-Leases.

```
root@desktop1:~# ifconfig ens5
ens5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.99.99.99  netmask 255.255.255.0  broadcast 10.99.99.255
        inet6 fe80::5054:ff:fe7e:f9b1  prefixlen 64  scopeid 0x20<link>
        ether 52:54:00:7e:f9:b1  txqueuelen 1000  (Ethernet)
        RX packets 170291  bytes 424870428 (424.8 MB)
        RX errors 0  dropped 99974  overruns 0  frame 0
        TX packets 29012  bytes 1942778 (1.9 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
.
root@desktop1:~# ip r
              default via 10.99.99.1 dev ens5 proto static
10.99.99.0/24 dev ens5 proto kernel scope link src 10.99.99.99
192.168.10.0/24 dev ens3 proto kernel scope link src 192.168.10.61
.
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=64 time=8.70 ms
64 bytes from 10.99.99.1: icmp_seq=2 ttl=64 time=7.06 ms
64 bytes from 10.99.99.1: icmp_seq=3 ttl=64 time=3.43 ms
.
root@desktop1:~# ping -c3 10.99.99.23
PING 10.99.99.23 (10.99.99.23) 56(84) bytes of data.
64 bytes from 10.99.99.23: icmp_seq=1 ttl=64 time=0.566 ms
64 bytes from 10.99.99.23: icmp_seq=2 ttl=64 time=0.531 ms
64 bytes from 10.99.99.23: icmp_seq=3 ttl=64 time=0.737 ms
.
root@desktop1:~# ping -c3 10.99.99.42
PING 10.99.99.42 (10.99.99.42) 56(84) bytes of data.
64 bytes from 10.99.99.42: icmp_seq=1 ttl=64 time=0.799 ms
64 bytes from 10.99.99.42: icmp_seq=2 ttl=64 time=0.532 ms
64 bytes from 10.99.99.42: icmp_seq=3 ttl=64 time=0.538 ms
.
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=59 time=0.671 ms
64 bytes from 10.88.88.88: icmp_seq=2 ttl=59 time=0.576 ms
64 bytes from 10.88.88.88: icmp_seq=3 ttl=59 time=0.611 ms
.
```

```
root@desktop1:~# ping -c3 10.88.88.23
PING 10.88.88.23 (10.88.88.23) 56(84) bytes of data.
64 bytes from 10.88.88.23: icmp_seq=1 ttl=59 time=1.09 ms
64 bytes from 10.88.88.23: icmp_seq=2 ttl=59 time=0.645 ms
64 bytes from 10.88.88.23: icmp_seq=3 ttl=59 time=0.650 ms
.
root@desktop1:~# 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=3.12 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=51 time=3.03 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=51 time=3.07 ms
.
root@desktop1:~# arp -an
? (10.99.99.1) at 00:00:5e:e4:31:57 [ether] on ens5
? (10.99.99.23) at 52:54:00:4a:e5:d0 [ether] on ens5
? (10.99.99.42) at 52:54:00:77:3c:03 [ether] on ens5
.
root@desktop1:~# dhclient -v ens5
Listening on LPF/ens5/52:54:00:7e:f9:b1
Sending on   LPF/ens5/52:54:00:7e:f9:b1
Sending on   Socket/fallback
DHCPDISCOVER on ens5 to 255.255.255.255 port 67 interval 3 (xid=0xefa2357a)
DHCPOFFER of 10.99.99.10 from 172.16.192.4
DHCPREQUEST for 10.99.99.10 on ens5 to 255.255.255.255 port 67 (xid=0x7a35a2ef)
DHCPACK of 10.99.99.10 from 172.16.192.4 (xid=0xefa2357a)
bound to 10.99.99.10 -- renewal in 765 seconds.
```

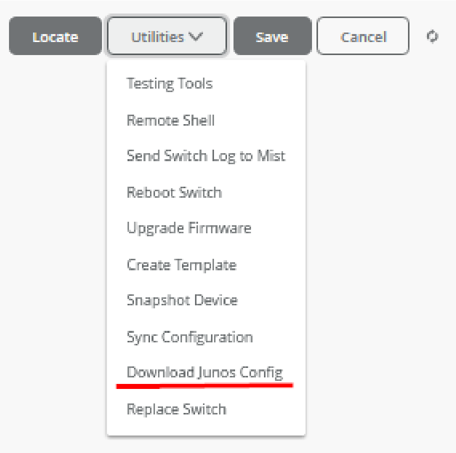
APPENDIX: Junos Configuration for this IP-Clos Fabric (Optional)

IN THIS SECTION

In this chapter we have documented the entire configuration Mist Cloud has pushed to all Fabric nodes. **It is not required that you read this chapter**. You may optionally refer to it if you are interested in the details of the configuration a new Fabric would receive by January 2025 built. If your Fabric has been built before this date some of the configurations may not be pushed the same way.

The best way to retrieve the actual configuration that is pushed to each switch is to use the **Utilities-Tab** > Download Junos Config .

Figure 76: Download Junos Config



In all below, examples we've reorganized the Junos configuration into different buckets and commented those. This will better help to understand the individual configuration option Junos uses.

Service1 Switch Junos Configuration

```
              # global system housekeeping
set system host-name service1
set system time-zone UTC
set system commit synchronize
set protocols lldp interface all
set protocols lldp port-id-subtype interface-name
set protocols lldp port-description-type interface-alias
set protocols lldp-med interface all
set protocols rstp interface all
set protocols rstp bpdu-block-on-edge
set groups top system commit no-delta-synchronize
set groups top system name-server 8.8.8.8
set groups top system name-server 9.9.9.9
set groups top system ntp server 192.168.10.1
set groups top system syslog file messages authorization any
set groups top system syslog file messages archive files 5
set groups top system syslog file messages archive size 2m
set groups top system syslog file interactive-commands match "!(.*mist.*)"
set groups top system syslog file interactive-commands archive files 5
set groups top system syslog file interactive-commands archive size 2m
set groups top system syslog file escript.log archive files 5
set groups top system syslog file escript.log archive size 2m
set groups top system syslog file op-script.log archive files 5
set groups top system syslog file op-script.log archive size 2m
set groups top system syslog file snapshot archive files 5
set groups top system syslog file snapshot archive size 2m
set apply-groups top
#
# up or downlink interfaces to other fabric nodes
set interfaces et-0/0/52 unit 0 family inet address 10.255.240.2/31
set interfaces et-0/0/52 description evpn_downlink-to-182ad301e1d0
set interfaces et-0/0/53 unit 0 family inet address 10.255.240.4/31
set interfaces et-0/0/53 description evpn_downlink-to-384f49f33ffc
#
# Underlay Loopback interface, router ID, and AS number
set groups top interfaces lo0 unit 0 family inet address 172.16.254.1/32
set groups top routing-options router-id 172.16.254.1
set groups top routing-options autonomous-system 65001
```

```
#
# Per-packet load balancing
set groups top policy-options policy-statement ecmp_policy then load-balance per-packet
set groups top policy-options policy-statement ecmp_policy then accept
set groups top routing-options forwarding-table export ecmp_policy
set groups top forwarding-options vxlan-routing overlay-ecmp
set routing-options forwarding-table ecmp-fast-reroute
set routing-options forwarding-table chained-composite-next-hop ingress evpn
#
# BGP underlay network to other fabric nodes
set groups top policy-options policy-statement evpn_underlay_export term 01-loopback from route-filter 172.16.254.0/23 orlonger
set groups top policy-options policy-statement evpn_underlay_export term 01-loopback then accept
set groups top policy-options policy-statement evpn_underlay_export term 02-default then reject
set groups top policy-options policy-statement evpn_underlay_import term 01-loopback from route-filter 172.16.254.0/23 orlonger
set groups top policy-options policy-statement evpn_underlay_import term 01-loopback then accept
set groups top policy-options policy-statement evpn_underlay_import term 02-default then reject
set protocols bgp group evpn_underlay type external
set protocols bgp group evpn_underlay local-as 65001
set protocols bgp group evpn_underlay multipath multiple-as
set protocols bgp group evpn_underlay authentication-key <not-disclosed-here>
set protocols bgp group evpn_underlay family inet unicast
set protocols bgp group evpn_underlay bfd-liveness-detection minimum-interval 1000
set protocols bgp group evpn_underlay bfd-liveness-detection multiplier 3
set protocols bgp group evpn_underlay log-updown
set protocols bgp group evpn_underlay export evpn_underlay_export
set protocols bgp group evpn_underlay import evpn_underlay_import
set protocols bgp group evpn_underlay neighbor 10.255.240.3 peer-as 65003
set protocols bgp group evpn_underlay neighbor 10.255.240.5 peer-as 65004
set protocols bgp graceful-restart
#
# EVPN signalling to other fabric nodes
set groups top switch-options vrf-target target:65000:1
set protocols bgp group evpn_overlay type external
set protocols bgp group evpn_overlay local-address 172.16.254.1
set protocols bgp group evpn_overlay local-as 65001
set protocols bgp group evpn_overlay multipath multiple-as
set protocols bgp group evpn_overlay authentication-key <not-disclosed-here>
set protocols bgp group evpn_overlay family evpn signaling loops 2
set protocols bgp group evpn_overlay log-updown
set protocols bgp group evpn_overlay bfd-liveness-detection minimum-interval 1000
set protocols bgp group evpn_overlay bfd-liveness-detection multiplier 3
set protocols bgp group evpn_overlay bfd-liveness-detection session-mode automatic
set protocols bgp group evpn_overlay multihop ttl 1
set protocols bgp group evpn_overlay multihop no-nexthop-change
set protocols bgp group evpn_overlay neighbor 172.16.254.3 peer-as 65003
set protocols bgp group evpn_overlay neighbor 172.16.254.4 peer-as 65004
#
# EVPN type2/5 coexistence
set groups top policy-options policy-statement evpn_export_type5 term 01_ipv4 from protocol evpn
set groups top policy-options policy-statement evpn_export_type5 term 01_ipv4 from route-filter
0.0.0.0/0 prefix-length-range /32-/32
set groups top policy-options policy-statement evpn_export_type5 term 01_ipv4 then accept
set groups top policy-options policy-statement evpn_export_type5 term 02_ipv6 from protocol evpn
set groups top policy-options policy-statement evpn_export_type5 term 02_ipv6 from family inet6
set groups top policy-options policy-statement evpn_export_type5 term 02_ipv6 from route-filter
0::0/0 prefix-length-range /128-/128
set groups top policy-options policy-statement evpn_export_type5 term 02_ipv6 then accept
set groups top policy-options policy-statement evpn_export_type5 term 03_direct from protocol
direct
set groups top policy-options policy-statement evpn_export_type5 term 03_direct then accept
set groups top policy-options policy-statement evpn_export_type5 term 04_bgp from protocol bgp
set groups top policy-options policy-statement evpn_export_type5 term 04_bgp then accept
#
# interface housekeeping
set interfaces interface-range default apply-groups default
set interfaces interface-range default member et-0/0/[0-51]
set interfaces interface-range default member et-0/0/[54-55]
set interfaces interface-range default member ge-0/0/[0-47]
set interfaces interface-range default member xe-0/0/[0-35]
set interfaces interface-range default member xe-0/0/[37-47]
set interfaces interface-range evpn_downlink apply-groups evpn_downlink
set interfaces interface-range evpn_downlink member et-0/0/52
set interfaces interface-range evpn_downlink member et-0/0/53
set interfaces vme unit 0 family inet dhcp vendor-id Juniper
set interfaces vme unit 0 family inet dhcp force-discover
set interfaces vme unit 0 family inet dhcp retransmission-attempt 60
set interfaces vme unit 0 family inet dhcp client-identifier user-id ascii 74e79806d100-M4aLquH9
set interfaces irb unit 0 family inet dhcp vendor-id Juniper
set interfaces irb unit 0 family inet dhcp force-discover
set interfaces irb unit 0 family inet dhcp retransmission-attempt 60
set interfaces irb unit 0 family inet dhcp client-identifier user-id ascii 74e79806d100-0
set interfaces irb unit 0 family inet mtu 9000
set interfaces irb unit 0 description default
set groups default interfaces <*> unit 0 family ethernet-switching vlan members [ default ]
set groups evpn_downlink interfaces <*> mtu 9192
set groups top forwarding-options storm-control-profiles default all
#
# first VRF (includes BGP-peering with WAN-Router and DHCP-Relay)
set groups top routing-instances corp-it instance-type vrf
set groups top routing-instances corp-it interface xe-0/0/36.1099
set groups top routing-instances corp-it protocols bgp group corp-it0 type external
set groups top routing-instances corp-it protocols bgp group corp-it0 log-updown
set groups top routing-instances corp-it protocols bgp group corp-it0 multipath multiple-as
set groups top routing-instances corp-it protocols bgp group corp-it0 neighbor 10.255.224.0 peer-as 64901
set groups top routing-instances corp-it protocols bgp group corp-it0 neighbor 10.255.224.0 hold-time 90
set groups top routing-instances corp-it protocols bgp group corp-it0 local-as 64911
set groups top routing-instances corp-it protocols bgp group corp-it0 hold-time 90
set groups top routing-instances corp-it protocols bgp group corp-it0 import import-default
set groups top routing-instances corp-it protocols bgp group corp-it0 export export-vrfs
set groups top routing-instances corp-it protocols bgp group corp-it0 bfd-liveness-detection minimum-interval 1000
set groups top routing-instances corp-it protocols bgp group corp-it0 bfd-liveness-detection multiplier 3
set groups top routing-instances corp-it protocols bgp group corp-it0 bfd-liveness-detection session-mode automatic
set groups top routing-instances corp-it protocols evpn ip-prefix-routes advertise direct-nexthop
set groups top routing-instances corp-it protocols evpn ip-prefix-routes encapsulation vxlan
set groups top routing-instances corp-it protocols evpn ip-prefix-routes vni 11284517
set groups top routing-instances corp-it protocols evpn ip-prefix-routes export evpn_export_type5
set groups top routing-instances corp-it route-distinguisher 172.16.254.1:101
set groups top routing-instances corp-it vrf-target target:65000:101
set groups top routing-instances corp-it vrf-table-label
set groups top routing-instances corp-it routing-options auto-export
set groups top routing-instances corp-it routing-options multipath
set groups top routing-instances corp-it interface lo0.1
#
# second VRF (includes BGP-peering with WAN-Router and DHCP-Relay)
set groups top routing-instances developers instance-type vrf
set groups top routing-instances developers interface xe-0/0/36.1088
set groups top routing-instances developers protocols bgp group developers0 type external
set groups top routing-instances developers protocols bgp group developers0 log-updown
set groups top routing-instances developers protocols bgp group developers0 multipath multiple-as
set groups top routing-instances developers protocols bgp group developers0 neighbor 10.255.224.2 peer-as 64901
set groups top routing-instances developers protocols bgp group developers0 neighbor
```

```
10.255.224.2 hold-time 90
set groups top routing-instances developers protocols bgp group developers0 local-as 64911
set groups top routing-instances developers protocols bgp group developers0 hold-time 90
set groups top routing-instances developers protocols bgp group developers0 import import-default
set groups top routing-instances developers protocols bgp group developers0 export export-vrfs
set groups top routing-instances developers protocols bgp group developers0 bfd-liveness-
detection minimum-interval 1000
set groups top routing-instances developers protocols bgp group developers0 bfd-liveness-
detection multiplier 3
set groups top routing-instances developers protocols bgp group developers0 bfd-liveness-
detection session-mode automatic
set groups top routing-instances developers protocols evpn ip-prefix-routes advertise direct-
nexthop
set groups top routing-instances developers protocols evpn ip-prefix-routes encapsulation vxlan
set groups top routing-instances developers protocols evpn ip-prefix-routes vni 15600414
set groups top routing-instances developers protocols evpn ip-prefix-routes export
evpn_export_type5
set groups top routing-instances developers route-distinguisher 172.16.254.1:102
set groups top routing-instances developers vrf-target target:65000:102
set groups top routing-instances developers vrf-table-label
set groups top routing-instances developers routing-options auto-export
set groups top routing-instances developers routing-options multipath
set groups top routing-instances developers interface lo0.2
#
# third VRF (includes BGP-peering with WAN-Router and DHCP-Relay)
set groups top routing-instances guest-wifi instance-type vrf
set groups top routing-instances guest-wifi interface xe-0/0/36.1033
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 type external
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 log-updown
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 multipath multiple-as
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 neighbor
10.255.224.4 peer-as 64901
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 neighbor
10.255.224.4 hold-time 90
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 local-as 64911
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 hold-time 90
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 import import-default
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 export export-vrfs
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 bfd-liveness-
detection minimum-interval 1000
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 bfd-liveness-
detection multiplier 3
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 bfd-liveness-
```

```
detection session-mode automatic
set groups top routing-instances guest-wifi protocols evpn ip-prefix-routes advertise direct-
nexthop
set groups top routing-instances guest-wifi protocols evpn ip-prefix-routes encapsulation vxlan
set groups top routing-instances guest-wifi protocols evpn ip-prefix-routes vni 15560868
set groups top routing-instances guest-wifi protocols evpn ip-prefix-routes export
evpn_export_type5
set groups top routing-instances guest-wifi route-distinguisher 172.16.254.1:103
set groups top routing-instances guest-wifi vrf-target target:65000:103
set groups top routing-instances guest-wifi vrf-table-label
set groups top routing-instances guest-wifi routing-options auto-export
set groups top routing-instances guest-wifi routing-options multipath
set groups top routing-instances guest-wifi interface lo0.3
#
# Overlay loopbacks for DHCP-Relay
set groups top interfaces lo0 unit 3 family inet address 172.16.192.3/32
set groups top interfaces lo0 unit 3 family inet6 address fd33:ab00:2::3/128
set groups top interfaces lo0 unit 2 family inet address 172.16.192.2/32
set groups top interfaces lo0 unit 2 family inet6 address fd33:ab00:2::2/128
set groups top interfaces lo0 unit 1 family inet address 172.16.192.1/32
set groups top interfaces lo0 unit 1 family inet6 address fd33:ab00:2::1/128
#
# uplink interfaces to WAN-Router
set interfaces xe-0/0/36 flexible-vlan-tagging
set interfaces xe-0/0/36 unit 1033 family inet address 10.255.224.5/31
set interfaces xe-0/0/36 unit 1033 description VLAN1033
set interfaces xe-0/0/36 unit 1033 vlan-id 1033
set interfaces xe-0/0/36 mtu 9018
set interfaces xe-0/0/36 unit 1088 family inet address 10.255.224.3/31
set interfaces xe-0/0/36 unit 1088 description VLAN1088
set interfaces xe-0/0/36 unit 1088 vlan-id 1088
set interfaces xe-0/0/36 unit 1099 family inet address 10.255.224.1/31
set interfaces xe-0/0/36 unit 1099 description VLAN1099
set interfaces xe-0/0/36 unit 1099 vlan-id 1099
set groups inet interfaces <*> mtu 9018
set interfaces interface-range inet apply-groups inet
set interfaces interface-range inet member xe-0/0/36
#
# BGP route policies for WAN-Router integration
set groups top policy-options route-filter-list 10-99-99-0_24 10.99.99.0/24 exact
set groups top policy-options policy-statement export-vrfs term 01_VLAN1099 from route-filter-
list 10-99-99-0_24
set groups top policy-options policy-statement export-vrfs term 01_VLAN1099 then accept
set groups top policy-options policy-statement export-vrfs term 02_VLAN1088 from route-filter-list 10-88-88-0_24
set groups top policy-options policy-statement export-vrfs term 02_VLAN1088 then accept
set groups top policy-options policy-statement export-vrfs term 03_VLAN1033 from route-filter-list 10-33-33-0_24
set groups top policy-options policy-statement export-vrfs term 03_VLAN1033 then accept
set groups top policy-options policy-statement export-vrfs term 04_overlaylo0 from route-filter-list 172-16-192-0_24-32
set groups top policy-options policy-statement export-vrfs term 04_overlaylo0 then accept
set groups top policy-options route-filter-list 10-88-88-0_24 10.88.88.0/24 exact
set groups top policy-options route-filter-list 10-33-33-0_24 10.33.33.0/24 exact
set groups top policy-options route-filter-list 172-16-192-0_24-32 172.16.192.0/24 upto /32
set groups top policy-options route-filter-list 0-0-0-0_0 0.0.0.0/0 exact
set groups top policy-options policy-statement import-default term 01_default from protocol [ bgp ]
set groups top policy-options policy-statement import-default term 01_default from route-filter-list 0-0-0-0_0
set groups top policy-options policy-statement import-default term 01_default then accept
#
# VXLAN global settings
set groups top protocols evpn encapsulation vxlan
set groups top protocols evpn default-gateway do-not-advertise
set groups top protocols evpn extended-vni-list all
set groups top switch-options vtep-source-interface lo0.0
set groups top switch-options route-distinguisher 172.16.254.1:1
#
# VXLAN tuneing parameters based on device model
set groups top forwarding-options vxlan-routing next-hop 45056
set groups top forwarding-options vxlan-routing interface-num 8192
#
# VLAN to VNI mapping
set vlans VLAN1033 vlan-id 1033
set vlans VLAN1033 vxlan vni 11033
set vlans VLAN1088 vlan-id 1088
set vlans VLAN1088 vxlan vni 11088
set vlans VLAN1099 vlan-id 1099
set vlans VLAN1099 vxlan vni 11099
set vlans default vlan-id 1
set vlans default l3-interface irb.0
set vlans default vxlan vni 10001
```

Service2 Switch Junos Configuration

```
              # global system housekeeping
set system host-name service2
set system time-zone UTC
set system commit synchronize
set protocols lldp interface all
set protocols lldp port-id-subtype interface-name
set protocols lldp port-description-type interface-alias
set protocols lldp-med interface all
set protocols rstp interface all
set protocols rstp bpdu-block-on-edge
set groups top system commit no-delta-synchronize
set groups top system name-server 8.8.8.8
set groups top system name-server 9.9.9.9
set groups top system ntp server 192.168.10.1
set groups top system syslog file messages authorization any
set groups top system syslog file messages archive files 5
set groups top system syslog file messages archive size 2m
set groups top system syslog file interactive-commands match "!(.*mist.*)"
set groups top system syslog file interactive-commands archive files 5
set groups top system syslog file interactive-commands archive size 2m
set groups top system syslog file escript.log archive files 5
set groups top system syslog file escript.log archive size 2m
set groups top system syslog file op-script.log archive files 5
set groups top system syslog file op-script.log archive size 2m
set groups top system syslog file snapshot archive files 5
set groups top system syslog file snapshot archive size 2m
set apply-groups top
#
# up or downlink interfaces to other fabric nodes
set interfaces et-0/0/53 unit 0 family inet address 10.255.240.6/31
set interfaces et-0/0/53 description evpn_downlink-to-182ad301e1d0
set interfaces et-0/0/52 unit 0 family inet address 10.255.240.8/31
set interfaces et-0/0/52 description evpn_downlink-to-384f49f33ffc
#
# Underlay Loopback interface, router ID, and AS number
set groups top interfaces lo0 unit 0 family inet address 172.16.254.2/32
set groups top routing-options router-id 172.16.254.2
set groups top routing-options autonomous-system 65002
```

```
#
# Per-packet load balancing
set groups top policy-options policy-statement ecmp_policy then load-balance per-packet
set groups top policy-options policy-statement ecmp_policy then accept
set groups top routing-options forwarding-table export ecmp_policy
set groups top forwarding-options vxlan-routing overlay-ecmp
set routing-options forwarding-table ecmp-fast-reroute
set routing-options forwarding-table chained-composite-next-hop ingress evpn
#
# BGP underlay network to other fabric nodes
set groups top policy-options policy-statement evpn_underlay_export term 01-loopback from route-filter 172.16.254.0/23 orlonger
set groups top policy-options policy-statement evpn_underlay_export term 01-loopback then accept
set groups top policy-options policy-statement evpn_underlay_export term 02-default then reject
set groups top policy-options policy-statement evpn_underlay_import term 01-loopback from route-filter 172.16.254.0/23 orlonger
set groups top policy-options policy-statement evpn_underlay_import term 01-loopback then accept
set groups top policy-options policy-statement evpn_underlay_import term 02-default then reject
set protocols bgp group evpn_underlay type external
set protocols bgp group evpn_underlay local-as 65002
set protocols bgp group evpn_underlay multipath multiple-as
set protocols bgp group evpn_underlay authentication-key <not-disclosed-here>
set protocols bgp group evpn_underlay family inet unicast
set protocols bgp group evpn_underlay bfd-liveness-detection minimum-interval 1000
set protocols bgp group evpn_underlay bfd-liveness-detection multiplier 3
set protocols bgp group evpn_underlay log-updown
set protocols bgp group evpn_underlay export evpn_underlay_export
set protocols bgp group evpn_underlay import evpn_underlay_import
set protocols bgp group evpn_underlay neighbor 10.255.240.7 peer-as 65003
set protocols bgp group evpn_underlay neighbor 10.255.240.9 peer-as 65004
set protocols bgp graceful-restart
#
# EVPN signalling to other fabric nodes
set groups top switch-options vrf-target target:65000:1
set protocols bgp group evpn_overlay type external
set protocols bgp group evpn_overlay local-address 172.16.254.2
set protocols bgp group evpn_overlay local-as 65002
set protocols bgp group evpn_overlay multipath multiple-as
set protocols bgp group evpn_overlay authentication-key <not-disclosed-here>
set protocols bgp group evpn_overlay family evpn signaling loops 2
set protocols bgp group evpn_overlay log-updown
set protocols bgp group evpn_overlay bfd-liveness-detection minimum-interval 1000
set protocols bgp group evpn_overlay bfd-liveness-detection multiplier 3
set protocols bgp group evpn_overlay bfd-liveness-detection session-mode automatic
set protocols bgp group evpn_overlay multihop ttl 1
set protocols bgp group evpn_overlay multihop no-nexthop-change
set protocols bgp group evpn_overlay neighbor 172.16.254.3 peer-as 65003
set protocols bgp group evpn_overlay neighbor 172.16.254.4 peer-as 65004
#
# EVPN type2/5 coexistence
set groups top policy-options policy-statement evpn_export_type5 term 01_ipv4 from protocol evpn
set groups top policy-options policy-statement evpn_export_type5 term 01_ipv4 from route-filter
0.0.0.0/0 prefix-length-range /32-/32
set groups top policy-options policy-statement evpn_export_type5 term 01_ipv4 then accept
set groups top policy-options policy-statement evpn_export_type5 term 02_ipv6 from protocol evpn
set groups top policy-options policy-statement evpn_export_type5 term 02_ipv6 from family inet6
set groups top policy-options policy-statement evpn_export_type5 term 02_ipv6 from route-filter
0::0/0 prefix-length-range /128-/128
set groups top policy-options policy-statement evpn_export_type5 term 02_ipv6 then accept
set groups top policy-options policy-statement evpn_export_type5 term 03_direct from protocol
direct
set groups top policy-options policy-statement evpn_export_type5 term 03_direct then accept
set groups top policy-options policy-statement evpn_export_type5 term 04_bgp from protocol bgp
set groups top policy-options policy-statement evpn_export_type5 term 04_bgp then accept
#
# interface housekeeping
set interfaces interface-range default apply-groups default
set interfaces interface-range default member et-0/0/[0-51]
set interfaces interface-range default member et-0/0/[54-55]
set interfaces interface-range default member ge-0/0/[0-47]
set interfaces interface-range default member xe-0/0/[0-35]
set interfaces interface-range default member xe-0/0/[37-47]
set interfaces interface-range evpn_downlink apply-groups evpn_downlink
set interfaces interface-range evpn_downlink member et-0/0/53
set interfaces interface-range evpn_downlink member et-0/0/52
set interfaces vme unit 0 family inet dhcp vendor-id Juniper
set interfaces vme unit 0 family inet dhcp force-discover
set interfaces vme unit 0 family inet dhcp retransmission-attempt 60
set interfaces vme unit 0 family inet dhcp client-identifier user-id ascii 74e7980fa000-M4aLquH9
set interfaces irb unit 0 family inet dhcp vendor-id Juniper
set interfaces irb unit 0 family inet dhcp force-discover
set interfaces irb unit 0 family inet dhcp retransmission-attempt 60
set interfaces irb unit 0 family inet dhcp client-identifier user-id ascii 74e7980fa000-0
set interfaces irb unit 0 family inet mtu 9000
set interfaces irb unit 0 description default
set groups default interfaces <*> unit 0 family ethernet-switching vlan members [ default ]
set groups evpn_downlink interfaces <*> mtu 9192
set groups top forwarding-options storm-control-profiles default all
#
# first VRF (includes BGP-peering with WAN-Router and DHCP-Relay)
set groups top routing-instances corp-it instance-type vrf
set groups top routing-instances corp-it interface xe-0/0/36.1099
set groups top routing-instances corp-it protocols bgp group corp-it0 type external
set groups top routing-instances corp-it protocols bgp group corp-it0 log-updown
set groups top routing-instances corp-it protocols bgp group corp-it0 multipath multiple-as
set groups top routing-instances corp-it protocols bgp group corp-it0 neighbor 10.255.226.0 peer-as 64901
set groups top routing-instances corp-it protocols bgp group corp-it0 neighbor 10.255.226.0 hold-time 90
set groups top routing-instances corp-it protocols bgp group corp-it0 local-as 64911
set groups top routing-instances corp-it protocols bgp group corp-it0 hold-time 90
set groups top routing-instances corp-it protocols bgp group corp-it0 import import-default
set groups top routing-instances corp-it protocols bgp group corp-it0 export export-vrfs
set groups top routing-instances corp-it protocols bgp group corp-it0 bfd-liveness-detection minimum-interval 1000
set groups top routing-instances corp-it protocols bgp group corp-it0 bfd-liveness-detection multiplier 3
set groups top routing-instances corp-it protocols bgp group corp-it0 bfd-liveness-detection session-mode automatic
set groups top routing-instances corp-it protocols evpn ip-prefix-routes advertise direct-nexthop
set groups top routing-instances corp-it protocols evpn ip-prefix-routes encapsulation vxlan
set groups top routing-instances corp-it protocols evpn ip-prefix-routes vni 11284517
set groups top routing-instances corp-it protocols evpn ip-prefix-routes export evpn_export_type5
set groups top routing-instances corp-it route-distinguisher 172.16.254.2:101
set groups top routing-instances corp-it vrf-target target:65000:101
set groups top routing-instances corp-it vrf-table-label
set groups top routing-instances corp-it routing-options auto-export
set groups top routing-instances corp-it routing-options multipath
set groups top routing-instances corp-it interface lo0.1
#
# second VRF (includes BGP-peering with WAN-Router and DHCP-Relay)
set groups top routing-instances developers instance-type vrf
set groups top routing-instances developers interface xe-0/0/36.1088
set groups top routing-instances developers protocols bgp group developers0 type external
set groups top routing-instances developers protocols bgp group developers0 log-updown
set groups top routing-instances developers protocols bgp group developers0 multipath multiple-as
set groups top routing-instances developers protocols bgp group developers0 neighbor 10.255.226.2 peer-as 64901
set groups top routing-instances developers protocols bgp group developers0 neighbor
```

```
10.255.226.2 hold-time 90
set groups top routing-instances developers protocols bgp group developers0 local-as 64911
set groups top routing-instances developers protocols bgp group developers0 hold-time 90
set groups top routing-instances developers protocols bgp group developers0 import import-default
set groups top routing-instances developers protocols bgp group developers0 export export-vrfs
set groups top routing-instances developers protocols bgp group developers0 bfd-liveness-
detection minimum-interval 1000
set groups top routing-instances developers protocols bgp group developers0 bfd-liveness-
detection multiplier 3
set groups top routing-instances developers protocols bgp group developers0 bfd-liveness-
detection session-mode automatic
set groups top routing-instances developers protocols evpn ip-prefix-routes advertise direct-
nexthop
set groups top routing-instances developers protocols evpn ip-prefix-routes encapsulation vxlan
set groups top routing-instances developers protocols evpn ip-prefix-routes vni 15600414
set groups top routing-instances developers protocols evpn ip-prefix-routes export
evpn_export_type5
set groups top routing-instances developers route-distinguisher 172.16.254.2:102
set groups top routing-instances developers vrf-target target:65000:102
set groups top routing-instances developers vrf-table-label
set groups top routing-instances developers routing-options auto-export
set groups top routing-instances developers routing-options multipath
set groups top routing-instances developers interface lo0.2
#
# third VRF (includes BGP-peering with WAN-Router and DHCP-Relay)
set groups top routing-instances guest-wifi instance-type vrf
set groups top routing-instances guest-wifi interface xe-0/0/36.1033
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 type external
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 log-updown
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 multipath multiple-as
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 neighbor
10.255.226.4 peer-as 64901
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 neighbor
10.255.226.4 hold-time 90
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 local-as 64911
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 hold-time 90
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 import import-default
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 export export-vrfs
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 bfd-liveness-
detection minimum-interval 1000
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 bfd-liveness-
detection multiplier 3
set groups top routing-instances guest-wifi protocols bgp group guest-wifi0 bfd-liveness-
```

```
detection session-mode automatic
set groups top routing-instances guest-wifi protocols evpn ip-prefix-routes advertise direct-
nexthop
set groups top routing-instances guest-wifi protocols evpn ip-prefix-routes encapsulation vxlan
set groups top routing-instances guest-wifi protocols evpn ip-prefix-routes vni 15560868
set groups top routing-instances guest-wifi protocols evpn ip-prefix-routes export
evpn_export_type5
set groups top routing-instances guest-wifi route-distinguisher 172.16.254.2:103
set groups top routing-instances guest-wifi vrf-target target:65000:103
set groups top routing-instances guest-wifi vrf-table-label
set groups top routing-instances guest-wifi routing-options auto-export
set groups top routing-instances guest-wifi routing-options multipath
set groups top routing-instances guest-wifi interface lo0.3
#
# Overlay loopbacks for DHCP-Relay
set groups top interfaces lo0 unit 3 family inet address 172.16.192.12/32
set groups top interfaces lo0 unit 3 family inet6 address fd33:ab00:2::c/128
set groups top interfaces lo0 unit 2 family inet address 172.16.192.11/32
set groups top interfaces lo0 unit 2 family inet6 address fd33:ab00:2::b/128
set groups top interfaces lo0 unit 1 family inet address 172.16.192.10/32
set groups top interfaces lo0 unit 1 family inet6 address fd33:ab00:2::a/128
#
# uplink interfaces to WAN-Router
set interfaces xe-0/0/36 flexible-vlan-tagging
set interfaces xe-0/0/36 unit 1033 family inet address 10.255.226.5/31
set interfaces xe-0/0/36 unit 1033 description VLAN1033
set interfaces xe-0/0/36 unit 1033 vlan-id 1033
set interfaces xe-0/0/36 mtu 9018
set interfaces xe-0/0/36 unit 1088 family inet address 10.255.226.3/31
set interfaces xe-0/0/36 unit 1088 description VLAN1088
set interfaces xe-0/0/36 unit 1088 vlan-id 1088
set interfaces xe-0/0/36 unit 1099 family inet address 10.255.226.1/31
set interfaces xe-0/0/36 unit 1099 description VLAN1099
set interfaces xe-0/0/36 unit 1099 vlan-id 1099
set groups inet interfaces <*> mtu 9018
set interfaces interface-range inet apply-groups inet
set interfaces interface-range inet member xe-0/0/36
#
# BGP route policies for WAN-Router integration
set groups top policy-options route-filter-list 10-99-99-0_24 10.99.99.0/24 exact
set groups top policy-options policy-statement export-vrfs term 01_VLAN1099 from route-filter-
list 10-99-99-0_24
set groups top policy-options policy-statement export-vrfs term 01_VLAN1099 then accept
set groups top policy-options policy-statement export-vrfs term 02_VLAN1088 from route-filter-list 10-88-88-0_24
set groups top policy-options policy-statement export-vrfs term 02_VLAN1088 then accept
set groups top policy-options policy-statement export-vrfs term 03_VLAN1033 from route-filter-list 10-33-33-0_24
set groups top policy-options policy-statement export-vrfs term 03_VLAN1033 then accept
set groups top policy-options policy-statement export-vrfs term 04_overlaylo0 from route-filter-list 172-16-192-0_24-32
set groups top policy-options policy-statement export-vrfs term 04_overlaylo0 then accept
set groups top policy-options route-filter-list 10-88-88-0_24 10.88.88.0/24 exact
set groups top policy-options route-filter-list 10-33-33-0_24 10.33.33.0/24 exact
set groups top policy-options route-filter-list 172-16-192-0_24-32 172.16.192.0/24 upto /32
set groups top policy-options route-filter-list 0-0-0-0_0 0.0.0.0/0 exact
set groups top policy-options policy-statement import-default term 01_default from protocol [ bgp ]
set groups top policy-options policy-statement import-default term 01_default from route-filter-list 0-0-0-0_0
set groups top policy-options policy-statement import-default term 01_default then accept
#
# VXLAN global settings
set groups top protocols evpn encapsulation vxlan
set groups top protocols evpn default-gateway do-not-advertise
set groups top protocols evpn extended-vni-list all
set groups top switch-options vtep-source-interface lo0.0
set groups top switch-options route-distinguisher 172.16.254.2:1
#
# VXLAN tuneing parameters based on device model
set groups top forwarding-options vxlan-routing next-hop 45056
set groups top forwarding-options vxlan-routing interface-num 8192
#
# VLAN to VNI mapping
set vlans VLAN1033 vlan-id 1033
set vlans VLAN1033 vxlan vni 11033
set vlans VLAN1088 vlan-id 1088
set vlans VLAN1088 vxlan vni 11088
set vlans VLAN1099 vlan-id 1099
set vlans VLAN1099 vxlan vni 11099
set vlans default vlan-id 1
set vlans default l3-interface irb.0
set vlans default vxlan vni 10001
```

Core1 Switch Junos Configuration

```
              # global system housekeeping
set system host-name core1
set system time-zone UTC
set system commit synchronize
set protocols lldp interface all
set protocols lldp port-id-subtype interface-name
set protocols lldp port-description-type interface-alias
set protocols lldp-med interface all
set protocols rstp interface all
set protocols rstp bpdu-block-on-edge
set groups top system commit no-delta-synchronize
set groups top system name-server 8.8.8.8
set groups top system name-server 9.9.9.9
set groups top system ntp server 192.168.10.1
set groups top system syslog file messages authorization any
set groups top system syslog file messages archive files 5
set groups top system syslog file messages archive size 2m
set groups top system syslog file interactive-commands match "!(.*mist.*)"
set groups top system syslog file interactive-commands archive files 5
set groups top system syslog file interactive-commands archive size 2m
set groups top system syslog file escript.log archive files 5
set groups top system syslog file escript.log archive size 2m
set groups top system syslog file op-script.log archive files 5
set groups top system syslog file op-script.log archive size 2m
set groups top system syslog file snapshot archive files 5
set groups top system syslog file snapshot archive size 2m
set apply-groups top
#
# up or downlink interfaces to other fabric nodes
set interfaces et-0/0/7 unit 0 family inet address 10.255.240.10/31
set interfaces et-0/0/7 description evpn_downlink-to-d8539a64a6c0
set interfaces et-0/0/6 unit 0 family inet address 10.255.240.12/31
set interfaces et-0/0/6 description evpn_downlink-to-d8539a6519c0
set interfaces et-0/0/8 unit 0 family inet address 10.255.240.3/31
set interfaces et-0/0/8 description evpn_uplink-to-74e79806d100
set interfaces et-0/0/9 unit 0 family inet address 10.255.240.7/31
set interfaces et-0/0/9 description evpn_uplink-to-74e7980fa000
#
```

```
# Underlay Loopback interface, router ID, and AS number
set groups top interfaces lo0 unit 0 family inet address 172.16.254.3/32
set groups top routing-options router-id 172.16.254.3
set groups top routing-options autonomous-system 65003
#
# Per-packet load balancing
set groups top policy-options policy-statement ecmp_policy then load-balance per-packet
set groups top policy-options policy-statement ecmp_policy then accept
set groups top routing-options forwarding-table export ecmp_policy
set routing-options forwarding-table ecmp-fast-reroute
set routing-options forwarding-table chained-composite-next-hop ingress evpn
#
# BGP underlay network to other fabric nodes
set groups top policy-options policy-statement evpn_underlay_export term 01-loopback from route-filter 172.16.254.0/23 orlonger
set groups top policy-options policy-statement evpn_underlay_export term 01-loopback then accept
set groups top policy-options policy-statement evpn_underlay_export term 02-default then reject
set groups top policy-options policy-statement evpn_underlay_import term 01-loopback from route-filter 172.16.254.0/23 orlonger
set groups top policy-options policy-statement evpn_underlay_import term 01-loopback then accept
set groups top policy-options policy-statement evpn_underlay_import term 02-default then reject
set protocols bgp group evpn_underlay type external
set protocols bgp group evpn_underlay local-as 65003
set protocols bgp group evpn_underlay multipath multiple-as
set protocols bgp group evpn_underlay authentication-key <not-disclosed-here>
set protocols bgp group evpn_underlay family inet unicast
set protocols bgp group evpn_underlay bfd-liveness-detection minimum-interval 1000
set protocols bgp group evpn_underlay bfd-liveness-detection multiplier 3
set protocols bgp group evpn_underlay log-updown
set protocols bgp group evpn_underlay export evpn_underlay_export
set protocols bgp group evpn_underlay import evpn_underlay_import
set protocols bgp group evpn_underlay neighbor 10.255.240.2 peer-as 65001
set protocols bgp group evpn_underlay neighbor 10.255.240.6 peer-as 65002
set protocols bgp group evpn_underlay neighbor 10.255.240.11 peer-as 65005
set protocols bgp group evpn_underlay neighbor 10.255.240.13 peer-as 65006
set protocols bgp graceful-restart
#
# EVPN signalling to other fabric nodes
set protocols bgp group evpn_overlay type external
set protocols bgp group evpn_overlay local-address 172.16.254.3
set protocols bgp group evpn_overlay local-as 65003
set protocols bgp group evpn_overlay multipath multiple-as
set protocols bgp group evpn_overlay authentication-key <not-disclosed-here>
set protocols bgp group evpn_overlay family evpn signaling loops 2
set protocols bgp group evpn_overlay log-updown
set protocols bgp group evpn_overlay bfd-liveness-detection minimum-interval 1000
set protocols bgp group evpn_overlay bfd-liveness-detection multiplier 3
set protocols bgp group evpn_overlay bfd-liveness-detection session-mode automatic
set protocols bgp group evpn_overlay multihop ttl 1
set protocols bgp group evpn_overlay multihop no-nexthop-change
set protocols bgp group evpn_overlay neighbor 172.16.254.1 peer-as 65001
set protocols bgp group evpn_overlay neighbor 172.16.254.2 peer-as 65002
set protocols bgp group evpn_overlay neighbor 172.16.254.5 peer-as 65005
set protocols bgp group evpn_overlay neighbor 172.16.254.6 peer-as 65006
#
# interface housekeeping
set interfaces interface-range default apply-groups default
set interfaces interface-range default member et-0/0/[0-5]
set interfaces interface-range default member et-0/0/[10-35]
set interfaces interface-range evpn_downlink apply-groups evpn_downlink
set interfaces interface-range evpn_downlink member et-0/0/7
set interfaces interface-range evpn_downlink member et-0/0/6
set interfaces interface-range evpn_uplink apply-groups evpn_uplink
set interfaces interface-range evpn_uplink member et-0/0/8
set interfaces interface-range evpn_uplink member et-0/0/9
set interfaces em0 unit 0 family inet dhcp vendor-id Juniper
set interfaces em0 unit 0 family inet dhcp force-discover
set interfaces em0 unit 0 family inet dhcp retransmission-attempt 60
set interfaces em0 unit 0 family inet dhcp client-identifier user-id ascii 182ad301e1d0-YGXs1ox4
set interfaces irb unit 0 family inet dhcp vendor-id Juniper
set interfaces irb unit 0 family inet dhcp force-discover
set interfaces irb unit 0 family inet dhcp retransmission-attempt 60
set interfaces irb unit 0 family inet dhcp client-identifier user-id ascii 182ad301e1d0-0
set interfaces irb unit 0 description default
set groups default interfaces <*> unit 0 family ethernet-switching vlan members [ default ]
set groups evpn_downlink interfaces <*> mtu 9192
set groups evpn_uplink interfaces <*> mtu 9192
set groups top forwarding-options storm-control-profiles default all
set vlans default vlan-id 1
set vlans default l3-interface irb.0
#
# additional CLI
set interfaces em0.0 family inet dhcp
```

Core2 Switch Junos Configuration

```
              # global system housekeeping
set system host-name core2
set system time-zone UTC
set system commit synchronize
set protocols lldp interface all
set protocols lldp port-id-subtype interface-name
set protocols lldp port-description-type interface-alias
set protocols lldp-med interface all
set protocols rstp interface all
set protocols rstp bpdu-block-on-edge
set groups top system commit no-delta-synchronize
set groups top system name-server 8.8.8.8
set groups top system name-server 9.9.9.9
set groups top system ntp server 192.168.10.1
set groups top system syslog file messages authorization any
set groups top system syslog file messages archive files 5
set groups top system syslog file messages archive size 2m
set groups top system syslog file interactive-commands match "!(.*mist.*)"
set groups top system syslog file interactive-commands archive files 5
set groups top system syslog file interactive-commands archive size 2m
set groups top system syslog file escript.log archive files 5
set groups top system syslog file escript.log archive size 2m
set groups top system syslog file op-script.log archive files 5
set groups top system syslog file op-script.log archive size 2m
set groups top system syslog file snapshot archive files 5
set groups top system syslog file snapshot archive size 2m
set apply-groups top
#
# up or downlink interfaces to other fabric nodes
set interfaces et-0/0/6 unit 0 family inet address 10.255.240.14/31
set interfaces et-0/0/6 description evpn_downlink-to-d8539a64a6c0
set interfaces et-0/0/7 unit 0 family inet address 10.255.240.16/31
set interfaces et-0/0/7 description evpn_downlink-to-d8539a6519c0
set interfaces et-0/0/9 unit 0 family inet address 10.255.240.5/31
set interfaces et-0/0/9 description evpn_uplink-to-74e79806d100
set interfaces et-0/0/8 unit 0 family inet address 10.255.240.9/31
set interfaces et-0/0/8 description evpn_uplink-to-74e7980fa000
#
```

```
# Underlay Loopback interface, router ID, and AS number
set groups top interfaces lo0 unit 0 family inet address 172.16.254.4/32
set groups top routing-options router-id 172.16.254.4
set groups top routing-options autonomous-system 65004
#
# Per-packet load balancing
set groups top policy-options policy-statement ecmp_policy then load-balance per-packet
set groups top policy-options policy-statement ecmp_policy then accept
set groups top routing-options forwarding-table export ecmp_policy
set routing-options forwarding-table ecmp-fast-reroute
set routing-options forwarding-table chained-composite-next-hop ingress evpn
#
# BGP underlay network to other fabric nodes
set groups top policy-options policy-statement evpn_underlay_export term 01-loopback from route-filter 172.16.254.0/23 orlonger
set groups top policy-options policy-statement evpn_underlay_export term 01-loopback then accept
set groups top policy-options policy-statement evpn_underlay_export term 02-default then reject
set groups top policy-options policy-statement evpn_underlay_import term 01-loopback from route-filter 172.16.254.0/23 orlonger
set groups top policy-options policy-statement evpn_underlay_import term 01-loopback then accept
set groups top policy-options policy-statement evpn_underlay_import term 02-default then reject
set protocols bgp group evpn_underlay type external
set protocols bgp group evpn_underlay local-as 65004
set protocols bgp group evpn_underlay multipath multiple-as
set protocols bgp group evpn_underlay authentication-key <not-disclosed-here>
set protocols bgp group evpn_underlay family inet unicast
set protocols bgp group evpn_underlay bfd-liveness-detection minimum-interval 1000
set protocols bgp group evpn_underlay bfd-liveness-detection multiplier 3
set protocols bgp group evpn_underlay log-updown
set protocols bgp group evpn_underlay export evpn_underlay_export
set protocols bgp group evpn_underlay import evpn_underlay_import
set protocols bgp group evpn_underlay neighbor 10.255.240.4 peer-as 65001
set protocols bgp group evpn_underlay neighbor 10.255.240.8 peer-as 65002
set protocols bgp group evpn_underlay neighbor 10.255.240.15 peer-as 65005
set protocols bgp group evpn_underlay neighbor 10.255.240.17 peer-as 65006
set protocols bgp graceful-restart
#
# EVPN signalling to other fabric nodes
set protocols bgp group evpn_overlay type external
set protocols bgp group evpn_overlay local-address 172.16.254.4
set protocols bgp group evpn_overlay local-as 65004
set protocols bgp group evpn_overlay multipath multiple-as
set protocols bgp group evpn_overlay authentication-key <not-disclosed-here>
set protocols bgp group evpn_overlay family evpn signaling loops 2
set protocols bgp group evpn_overlay log-updown
set protocols bgp group evpn_overlay bfd-liveness-detection minimum-interval 1000
set protocols bgp group evpn_overlay bfd-liveness-detection multiplier 3
set protocols bgp group evpn_overlay bfd-liveness-detection session-mode automatic
set protocols bgp group evpn_overlay multihop ttl 1
set protocols bgp group evpn_overlay multihop no-nexthop-change
set protocols bgp group evpn_overlay neighbor 172.16.254.1 peer-as 65001
set protocols bgp group evpn_overlay neighbor 172.16.254.2 peer-as 65002
set protocols bgp group evpn_overlay neighbor 172.16.254.5 peer-as 65005
set protocols bgp group evpn_overlay neighbor 172.16.254.6 peer-as 65006
#
# interface housekeeping
set interfaces interface-range default apply-groups default
set interfaces interface-range default member et-0/0/[0-5]
set interfaces interface-range default member et-0/0/[10-35]
set interfaces interface-range evpn_downlink apply-groups evpn_downlink
set interfaces interface-range evpn_downlink member et-0/0/6
set interfaces interface-range evpn_downlink member et-0/0/7
set interfaces interface-range evpn_uplink apply-groups evpn_uplink
set interfaces interface-range evpn_uplink member et-0/0/9
set interfaces interface-range evpn_uplink member et-0/0/8
set interfaces em0 unit 0 family inet dhcp vendor-id Juniper
set interfaces em0 unit 0 family inet dhcp force-discover
set interfaces em0 unit 0 family inet dhcp retransmission-attempt 60
set interfaces em0 unit 0 family inet dhcp client-identifier user-id ascii 384f49f33ffc-YGXs1ox4
set interfaces irb unit 0 family inet dhcp vendor-id Juniper
set interfaces irb unit 0 family inet dhcp force-discover
set interfaces irb unit 0 family inet dhcp retransmission-attempt 60
set interfaces irb unit 0 family inet dhcp client-identifier user-id ascii 384f49f33ffc-0
set interfaces irb unit 0 description default
set groups default interfaces <*> unit 0 family ethernet-switching vlan members [ default ]
set groups evpn_downlink interfaces <*> mtu 9192
set groups evpn_uplink interfaces <*> mtu 9192
set groups top forwarding-options storm-control-profiles default all
set vlans default vlan-id 1
set vlans default l3-interface irb.0
#
# additional CLI
set interfaces em0.0 family inet dhcp
```

Dist1 Switch Junos Configuration

```
              # global system housekeeping
set system host-name dist1
set system time-zone UTC
set system commit synchronize
set protocols lldp interface all
set protocols lldp port-id-subtype interface-name
set protocols lldp port-description-type interface-alias
set protocols lldp-med interface all
set protocols rstp interface all
set protocols rstp bpdu-block-on-edge
set groups top system commit no-delta-synchronize
set groups top system name-server 8.8.8.8
set groups top system name-server 9.9.9.9
set groups top system ntp server 192.168.10.1
set groups top system syslog file messages authorization any
set groups top system syslog file messages archive files 5
set groups top system syslog file messages archive size 2m
set groups top system syslog file interactive-commands match "!(.*mist.*)"
set groups top system syslog file interactive-commands archive files 5
set groups top system syslog file interactive-commands archive size 2m
set groups top system syslog file escript.log archive files 5
set groups top system syslog file escript.log archive size 2m
set groups top system syslog file op-script.log archive files 5
set groups top system syslog file op-script.log archive size 2m
set groups top system syslog file snapshot archive files 5
set groups top system syslog file snapshot archive size 2m
set apply-groups top
#
# up or downlink interfaces to other fabric nodes
set interfaces et-0/0/52 unit 0 family inet address 10.255.240.13/31
set interfaces et-0/0/52 description evpn_uplink-to-182ad301e1d0
set interfaces et-0/0/53 unit 0 family inet address 10.255.240.17/31
set interfaces et-0/0/53 description evpn_uplink-to-384f49f33ffc
set interfaces xe-0/0/37 unit 0 family inet address 10.255.240.22/31
set interfaces xe-0/0/37 description evpn_downlink-to-bc0ffe157080
set interfaces xe-0/0/36 unit 0 family inet address 10.255.240.24/31
set interfaces xe-0/0/36 description evpn_downlink-to-f8c116415c00
#
```

```
# Underlay Loopback interface, router ID, and AS number
set groups top interfaces lo0 unit 0 family inet address 172.16.254.6/32
set groups top routing-options router-id 172.16.254.6
set groups top routing-options autonomous-system 65006
#
# Per-packet load balancing
set groups top policy-options policy-statement ecmp_policy then load-balance per-packet
set groups top policy-options policy-statement ecmp_policy then accept
set groups top routing-options forwarding-table export ecmp_policy
set routing-options forwarding-table ecmp-fast-reroute
set routing-options forwarding-table chained-composite-next-hop ingress evpn
#
# BGP underlay network to other fabric nodes
set groups top policy-options policy-statement evpn_underlay_export term 01-loopback from route-filter 172.16.254.0/23 orlonger
set groups top policy-options policy-statement evpn_underlay_export term 01-loopback then accept
set groups top policy-options policy-statement evpn_underlay_export term 02-default then reject
set groups top policy-options policy-statement evpn_underlay_import term 01-loopback from route-filter 172.16.254.0/23 orlonger
set groups top policy-options policy-statement evpn_underlay_import term 01-loopback then accept
set groups top policy-options policy-statement evpn_underlay_import term 02-default then reject
set protocols bgp group evpn_underlay type external
set protocols bgp group evpn_underlay local-as 65006
set protocols bgp group evpn_underlay multipath multiple-as
set protocols bgp group evpn_underlay authentication-key <not-disclosed-here>
set protocols bgp group evpn_underlay family inet unicast
set protocols bgp group evpn_underlay bfd-liveness-detection minimum-interval 1000
set protocols bgp group evpn_underlay bfd-liveness-detection multiplier 3
set protocols bgp group evpn_underlay log-updown
set protocols bgp group evpn_underlay export evpn_underlay_export
set protocols bgp group evpn_underlay import evpn_underlay_import
set protocols bgp group evpn_underlay neighbor 10.255.240.12 peer-as 65003
set protocols bgp group evpn_underlay neighbor 10.255.240.16 peer-as 65004
set protocols bgp group evpn_underlay neighbor 10.255.240.23 peer-as 65007
set protocols bgp group evpn_underlay neighbor 10.255.240.25 peer-as 65008
set protocols bgp graceful-restart
#
# EVPN signalling to other fabric nodes
set protocols bgp group evpn_overlay type external
set protocols bgp group evpn_overlay local-address 172.16.254.6
set protocols bgp group evpn_overlay local-as 65006
set protocols bgp group evpn_overlay multipath multiple-as
set protocols bgp group evpn_overlay authentication-key <not-disclosed-here>
set protocols bgp group evpn_overlay family evpn signaling loops 2
set protocols bgp group evpn_overlay log-updown
set protocols bgp group evpn_overlay bfd-liveness-detection minimum-interval 1000
set protocols bgp group evpn_overlay bfd-liveness-detection multiplier 3
set protocols bgp group evpn_overlay bfd-liveness-detection session-mode automatic
set protocols bgp group evpn_overlay multihop ttl 1
set protocols bgp group evpn_overlay multihop no-nexthop-change
set protocols bgp group evpn_overlay neighbor 172.16.254.3 peer-as 65003
set protocols bgp group evpn_overlay neighbor 172.16.254.4 peer-as 65004
set protocols bgp group evpn_overlay neighbor 172.16.254.7 peer-as 65007
set protocols bgp group evpn_overlay neighbor 172.16.254.8 peer-as 65008
#
# interface housekeeping
set interfaces interface-range default apply-groups default
set interfaces interface-range default member et-0/0/[0-51]
set interfaces interface-range default member et-0/0/[54-55]
set interfaces interface-range default member ge-0/0/[0-47]
set interfaces interface-range default member xe-0/0/[0-35]
set interfaces interface-range default member xe-0/0/[38-47]
set interfaces interface-range evpn_uplink apply-groups evpn_uplink
set interfaces interface-range evpn_uplink member et-0/0/52
set interfaces interface-range evpn_uplink member et-0/0/53
set interfaces interface-range evpn_downlink apply-groups evpn_downlink
set interfaces interface-range evpn_downlink member xe-0/0/37
set interfaces interface-range evpn_downlink member xe-0/0/36
set interfaces vme unit 0 family inet dhcp vendor-id Juniper
set interfaces vme unit 0 family inet dhcp force-discover
set interfaces vme unit 0 family inet dhcp retransmission-attempt 60
set interfaces vme unit 0 family inet dhcp client-identifier user-id ascii d8539a6519c0-M4aLquH9
set interfaces irb unit 0 family inet dhcp vendor-id Juniper
set interfaces irb unit 0 family inet dhcp force-discover
set interfaces irb unit 0 family inet dhcp retransmission-attempt 60
set interfaces irb unit 0 family inet dhcp client-identifier user-id ascii d8539a6519c0-0
set interfaces irb unit 0 description default
set groups default interfaces <*> unit 0 family ethernet-switching vlan members [ default ]
set groups evpn_uplink interfaces <*> mtu 9192
set groups evpn_downlink interfaces <*> mtu 9192
set groups top forwarding-options storm-control-profiles default all
set vlans default vlan-id 1
set vlans default l3-interface irb.0
#
# additional CLI
set chassis fpc 0 pic 0 port 0 speed 1G
set chassis fpc 0 pic 0 port 8 speed 1G
set interfaces et-0/0/48 disable
set interfaces et-0/0/49 disable
```

Dist2 Switch Junos Configuration

```


              # global system housekeeping
set system host-name dist2
set system time-zone UTC
set system commit synchronize
set protocols lldp interface all
set protocols lldp port-id-subtype interface-name
set protocols lldp port-description-type interface-alias
set protocols lldp-med interface all
set protocols rstp interface all
set protocols rstp bpdu-block-on-edge
set groups top system commit no-delta-synchronize
set groups top system name-server 8.8.8.8
set groups top system name-server 9.9.9.9
set groups top system ntp server 192.168.10.1
set groups top system syslog file messages authorization any
set groups top system syslog file messages archive files 5
set groups top system syslog file messages archive size 2m
set groups top system syslog file interactive-commands match "!(.*mist.*)"
set groups top system syslog file interactive-commands archive files 5
set groups top system syslog file interactive-commands archive size 2m
set groups top system syslog file escript.log archive files 5
set groups top system syslog file escript.log archive size 2m
set groups top system syslog file op-script.log archive files 5
set groups top system syslog file op-script.log archive size 2m
set groups top system syslog file snapshot archive files 5
set groups top system syslog file snapshot archive size 2m
set apply-groups top
#
# up or downlink interfaces to other fabric nodes
set interfaces et-0/0/53 unit 0 family inet address 10.255.240.11/31
set interfaces et-0/0/53 description evpn_uplink-to-182ad301e1d0
set interfaces et-0/0/52 unit 0 family inet address 10.255.240.15/31
set interfaces et-0/0/52 description evpn_uplink-to-384f49f33ffc
set interfaces xe-0/0/36 unit 0 family inet address 10.255.240.18/31
set interfaces xe-0/0/36 description evpn_downlink-to-bc0ffe157080
set interfaces xe-0/0/37 unit 0 family inet address 10.255.240.20/31
set interfaces xe-0/0/37 description evpn_downlink-to-f8c116415c00
#
# Underlay Loopback interface, router ID, and AS number
set groups top interfaces lo0 unit 0 family inet address 172.16.254.5/32
set groups top routing-options router-id 172.16.254.5
set groups top routing-options autonomous-system 65005
#
# Per-packet load balancing
set groups top policy-options policy-statement ecmp_policy then load-balance per-packet
set groups top policy-options policy-statement ecmp_policy then accept
set groups top routing-options forwarding-table export ecmp_policy
set routing-options forwarding-table ecmp-fast-reroute
set routing-options forwarding-table chained-composite-next-hop ingress evpn
#
# BGP underlay network to other fabric nodes
set groups top policy-options policy-statement evpn_underlay_export term 01-loopback from route-filter 172.16.254.0/23 orlonger
set groups top policy-options policy-statement evpn_underlay_export term 01-loopback then accept
set groups top policy-options policy-statement evpn_underlay_export term 02-default then reject
set groups top policy-options policy-statement evpn_underlay_import term 01-loopback from route-filter 172.16.254.0/23 orlonger
set groups top policy-options policy-statement evpn_underlay_import term 01-loopback then accept
set groups top policy-options policy-statement evpn_underlay_import term 02-default then reject
set protocols bgp group evpn_underlay type external
set protocols bgp group evpn_underlay local-as 65005
set protocols bgp group evpn_underlay multipath multiple-as
set protocols bgp group evpn_underlay authentication-key <not-disclosed-here>
set protocols bgp group evpn_underlay family inet unicast
set protocols bgp group evpn_underlay bfd-liveness-detection minimum-interval 1000
set protocols bgp group evpn_underlay bfd-liveness-detection multiplier 3
set protocols bgp group evpn_underlay log-updown
set protocols bgp group evpn_underlay export evpn_underlay_export
set protocols bgp group evpn_underlay import evpn_underlay_import
set protocols bgp group evpn_underlay neighbor 10.255.240.10 peer-as 65003
set protocols bgp group evpn_underlay neighbor 10.255.240.14 peer-as 65004
set protocols bgp group evpn_underlay neighbor 10.255.240.19 peer-as 65007
set protocols bgp group evpn_underlay neighbor 10.255.240.21 peer-as 65008
set protocols bgp graceful-restart
#
```

```
# EVPN signalling to other fabric nodes
set protocols bgp group evpn_overlay type external
set protocols bgp group evpn_overlay local-address 172.16.254.5
set protocols bgp group evpn_overlay local-as 65005
set protocols bgp group evpn_overlay multipath multiple-as
set protocols bgp group evpn_overlay authentication-key <not-disclosed-here>
set protocols bgp group evpn_overlay family evpn signaling loops 2
set protocols bgp group evpn_overlay log-updown
set protocols bgp group evpn_overlay bfd-liveness-detection minimum-interval 1000
set protocols bgp group evpn_overlay bfd-liveness-detection multiplier 3
set protocols bgp group evpn_overlay bfd-liveness-detection session-mode automatic
set protocols bgp group evpn_overlay multihop ttl 1
set protocols bgp group evpn_overlay multihop no-nexthop-change
set protocols bgp group evpn_overlay neighbor 172.16.254.3 peer-as 65003
set protocols bgp group evpn_overlay neighbor 172.16.254.4 peer-as 65004
set protocols bgp group evpn_overlay neighbor 172.16.254.7 peer-as 65007
set protocols bgp group evpn_overlay neighbor 172.16.254.8 peer-as 65008
#
# interface housekeeping
set interfaces interface-range default apply-groups default
set interfaces interface-range default member et-0/0/[0-51]
set interfaces interface-range default member et-0/0/[54-55]
set interfaces interface-range default member ge-0/0/[0-47]
set interfaces interface-range default member xe-0/0/[0-35]
set interfaces interface-range default member xe-0/0/[38-47]
set interfaces interface-range evpn_uplink apply-groups evpn_uplink
set interfaces interface-range evpn_uplink member et-0/0/53
set interfaces interface-range evpn_uplink member et-0/0/52
set interfaces interface-range evpn_downlink apply-groups evpn_downlink
set interfaces interface-range evpn_downlink member xe-0/0/36
set interfaces interface-range evpn_downlink member xe-0/0/37
set interfaces vme unit 0 family inet dhcp vendor-id Juniper
set interfaces vme unit 0 family inet dhcp force-discover
set interfaces vme unit 0 family inet dhcp retransmission-attempt 60
set interfaces vme unit 0 family inet dhcp client-identifier user-id ascii d8539a64a6c0-M4aLquH9
set interfaces irb unit 0 family inet dhcp vendor-id Juniper
set interfaces irb unit 0 family inet dhcp force-discover
set interfaces irb unit 0 family inet dhcp retransmission-attempt 60
set interfaces irb unit 0 family inet dhcp client-identifier user-id ascii d8539a64a6c0-0
set interfaces irb unit 0 description default
set groups default interfaces <*> unit 0 family ethernet-switching vlan members [ default ]
set groups evpn_uplink interfaces <*> mtu 9192
set groups evpn_downlink interfaces <*> mtu 9192
set groups top forwarding-options storm-control-profiles default all
set vlans default vlan-id 1
set vlans default l3-interface irb.0
#
# additional CLI
set chassis fpc 0 pic 0 port 0 speed 1G
set chassis fpc 0 pic 0 port 8 speed 1G
set interfaces et-0/0/48 disable
set interfaces et-0/0/49 disable
```

Access1 Switch Junos Configuration

```


              # global system housekeeping
set system host-name access1
set system time-zone UTC
set system commit synchronize
set protocols lldp interface all
set protocols lldp port-id-subtype interface-name
set protocols lldp port-description-type interface-alias
set protocols lldp-med interface all
set protocols rstp interface all
set protocols rstp bpdu-block-on-edge
set groups top system commit no-delta-synchronize
set groups top system name-server 8.8.8.8
set groups top system name-server 9.9.9.9
set groups top system ntp server 192.168.10.1
set groups top system syslog file messages authorization any
set groups top system syslog file messages archive files 5
set groups top system syslog file messages archive size 2m
set groups top system syslog file interactive-commands match "!(.*mist.*)"
set groups top system syslog file interactive-commands archive files 5
set groups top system syslog file interactive-commands archive size 2m
set groups top system syslog file escript.log archive files 5
set groups top system syslog file escript.log archive size 2m
set groups top system syslog file op-script.log archive files 5
set groups top system syslog file op-script.log archive size 2m
set groups top system syslog file snapshot archive files 5
set groups top system syslog file snapshot archive size 2m
set apply-groups top
#
# up or downlink interfaces to other fabric nodes
set interfaces mge-0/0/37 unit 0 family inet address 10.255.240.21/31
set interfaces mge-0/0/37 description evpn_uplink-to-d8539a64a6c0
set interfaces mge-0/0/36 unit 0 family inet address 10.255.240.25/31
set interfaces mge-0/0/36 description evpn_uplink-to-d8539a6519c0
#
# Underlay Loopback interface, router ID, and AS number
set groups top interfaces lo0 unit 0 family inet address 172.16.254.8/32
set groups top routing-options router-id 172.16.254.8
set groups top routing-options autonomous-system 65008
#
# Per-packet load balancing
set groups top policy-options policy-statement ecmp_policy then load-balance per-packet
set groups top policy-options policy-statement ecmp_policy then accept
set groups top routing-options forwarding-table export ecmp_policy
set routing-options forwarding-table ecmp-fast-reroute
set routing-options forwarding-table chained-composite-next-hop ingress evpn
set groups top forwarding-options vxlan-routing overlay-ecmp
#
# BGP underlay network to other fabric nodes
set groups top policy-options policy-statement evpn_underlay_export term 01-loopback from route-filter 172.16.254.0/23 orlonger
set groups top policy-options policy-statement evpn_underlay_export term 01-loopback then accept
set groups top policy-options policy-statement evpn_underlay_export term 02-default then reject
set groups top policy-options policy-statement evpn_underlay_import term 01-loopback from route-filter 172.16.254.0/23 orlonger
set groups top policy-options policy-statement evpn_underlay_import term 01-loopback then accept
set groups top policy-options policy-statement evpn_underlay_import term 02-default then reject
set protocols bgp group evpn_underlay type external
set protocols bgp group evpn_underlay local-as 65008
set protocols bgp group evpn_underlay multipath multiple-as
set protocols bgp group evpn_underlay authentication-key <not-disclosed-here>
set protocols bgp group evpn_underlay family inet unicast
set protocols bgp group evpn_underlay bfd-liveness-detection minimum-interval 1000
set protocols bgp group evpn_underlay bfd-liveness-detection multiplier 3
set protocols bgp group evpn_underlay log-updown
set protocols bgp group evpn_underlay export evpn_underlay_export
set protocols bgp group evpn_underlay import evpn_underlay_import
set protocols bgp group evpn_underlay neighbor 10.255.240.20 peer-as 65005
set protocols bgp group evpn_underlay neighbor 10.255.240.24 peer-as 65006
set protocols bgp graceful-restart
```

```
#
# EVPN signalling to other fabric nodes
set groups top switch-options vrf-target target:65000:1
set protocols bgp group evpn_overlay type external
set protocols bgp group evpn_overlay local-address 172.16.254.8
set protocols bgp group evpn_overlay local-as 65008
set protocols bgp group evpn_overlay multipath multiple-as
set protocols bgp group evpn_overlay authentication-key <not-disclosed-here>
set protocols bgp group evpn_overlay family evpn signaling loops 2
set protocols bgp group evpn_overlay log-updown
set protocols bgp group evpn_overlay bfd-liveness-detection minimum-interval 1000
set protocols bgp group evpn_overlay bfd-liveness-detection multiplier 3
set protocols bgp group evpn_overlay bfd-liveness-detection session-mode automatic
set protocols bgp group evpn_overlay multihop ttl 1
set protocols bgp group evpn_overlay multihop no-nexthop-change
set protocols bgp group evpn_overlay neighbor 172.16.254.5 peer-as 65005
set protocols bgp group evpn_overlay neighbor 172.16.254.6 peer-as 65006
#
# EVPN type2/5 coexistence
set groups top policy-options policy-statement evpn_export_type5 term 01_ipv4 from protocol evpn
set groups top policy-options policy-statement evpn_export_type5 term 01_ipv4 from route-filter
0.0.0.0/0 prefix-length-range /32-/32
set groups top policy-options policy-statement evpn_export_type5 term 01_ipv4 then accept
set groups top policy-options policy-statement evpn_export_type5 term 02_ipv6 from protocol evpn
set groups top policy-options policy-statement evpn_export_type5 term 02_ipv6 from family inet6
set groups top policy-options policy-statement evpn_export_type5 term 02_ipv6 from route-filter
0::0/0 prefix-length-range /128-/128
set groups top policy-options policy-statement evpn_export_type5 term 02_ipv6 then accept
set groups top policy-options policy-statement evpn_export_type5 term 03_direct from protocol
direct
set groups top policy-options policy-statement evpn_export_type5 term 03_direct then accept
#
# interface housekeeping
set interfaces interface-range default apply-groups default
set interfaces interface-range default member et-0/1/[0-3]
set interfaces interface-range default member mge-0/0/[0-10]
set interfaces interface-range default member mge-0/0/12
set interfaces interface-range default member mge-0/0/15
set interfaces interface-range default member mge-0/0/[17-35]
set interfaces interface-range default member mge-0/0/[38-47]
set interfaces interface-range vlan1099-no-auth apply-groups vlan1099-no-auth
set interfaces interface-range vlan1099-no-auth member mge-0/0/11
set interfaces interface-range vlan1099-no-auth member mge-0/0/14
set interfaces interface-range vlan1088-no-auth apply-groups vlan1088-no-auth
set interfaces interface-range vlan1088-no-auth member mge-0/0/13
set interfaces interface-range access-point apply-groups access-point
set interfaces interface-range access-point member mge-0/0/16
set interfaces interface-range evpn_uplink apply-groups evpn_uplink
set interfaces interface-range evpn_uplink member mge-0/0/37
set interfaces interface-range evpn_uplink member mge-0/0/36
set interfaces vme unit 0 family inet dhcp vendor-id Juniper
set interfaces vme unit 0 family inet dhcp force-discover
set interfaces vme unit 0 family inet dhcp retransmission-attempt 60
set interfaces vme unit 0 family inet dhcp client-identifier user-id ascii f8c116415c00-M4aLquH9
set interfaces irb unit 0 family inet dhcp vendor-id Juniper
set interfaces irb unit 0 family inet dhcp force-discover
set interfaces irb unit 0 family inet dhcp retransmission-attempt 60
set interfaces irb unit 0 family inet dhcp client-identifier user-id ascii f8c116415c00-0
set interfaces irb unit 0 family inet mtu 9000
set interfaces irb unit 0 description default
set groups default interfaces <*> unit 0 family ethernet-switching vlan members [ default ]
set groups vlan1099-no-auth interfaces <*> unit 0 family ethernet-switching vlan members
[ VLAN1099 ]
set groups vlan1088-no-auth interfaces <*> unit 0 family ethernet-switching vlan members
[ VLAN1088 ]
set groups access-point interfaces <*> unit 0 family ethernet-switching interface-mode trunk
set groups access-point interfaces <*> unit 0 family ethernet-switching vlan members [ all ]
set groups access-point interfaces <*> native-vlan-id 1033
set groups evpn_uplink interfaces <*> mtu 9192
set groups top poe interface all
set groups top forwarding-options storm-control-profiles default all
#
# IRB's and default GW's for VLANs
set interfaces irb unit 1033 family inet address 10.33.33.1/24
set interfaces irb unit 1033 family inet mtu 9000
set interfaces irb unit 1033 description VLAN1033
set interfaces irb unit 1033 no-dhcp-flood
set interfaces irb unit 1033 mac 00:00:5e:e4:31:57
set interfaces irb unit 1088 family inet address 10.88.88.1/24
set interfaces irb unit 1088 family inet mtu 9000
set interfaces irb unit 1088 description VLAN1088
set interfaces irb unit 1088 no-dhcp-flood
set interfaces irb unit 1088 mac 00:00:5e:e4:31:57
set interfaces irb unit 1099 family inet address 10.99.99.1/24
set interfaces irb unit 1099 family inet mtu 9000
set interfaces irb unit 1099 description VLAN1099
set interfaces irb unit 1099 no-dhcp-flood
set interfaces irb unit 1099 mac 00:00:5e:e4:31:57
#
# first VRF (includes IRB and DHCP-Relay)
set groups top routing-instances developers instance-type vrf
set groups top routing-instances developers interface irb.1088
set groups top routing-instances developers forwarding-options dhcp-relay server-group VLAN1088 192.168.10.11
set groups top routing-instances developers forwarding-options dhcp-relay group VLAN1088 interface irb.1088
set groups top routing-instances developers forwarding-options dhcp-relay group VLAN1088 active-server-group VLAN1088
set groups top routing-instances developers forwarding-options dhcp-relay group VLAN1088 relay-option-82 circuit-id vlan-id-only
set groups top routing-instances developers forwarding-options dhcp-relay group VLAN1088 relay-option-82 server-id-override
set groups top routing-instances developers forwarding-options dhcp-relay group VLAN1088 route-suppression destination
set groups top routing-instances developers forwarding-options dhcp-relay group VLAN1088 overrides relay-source lo0.2
set groups top routing-instances developers forwarding-options dhcp-relay forward-only
set groups top routing-instances developers route-distinguisher 172.16.254.8:102
set groups top routing-instances developers vrf-target target:65000:102
set groups top routing-instances developers vrf-table-label
set groups top routing-instances developers routing-options auto-export
set groups top routing-instances developers routing-options multipath
set groups top routing-instances developers protocols evpn ip-prefix-routes advertise direct-nexthop
set groups top routing-instances developers protocols evpn ip-prefix-routes encapsulation vxlan
set groups top routing-instances developers protocols evpn ip-prefix-routes vni 15600414
set groups top routing-instances developers protocols evpn ip-prefix-routes export evpn_export_type5
set groups top routing-instances developers interface lo0.2
#
# second VRF (includes IRB and DHCP-Relay)
set groups top routing-instances corp-it instance-type vrf
set groups top routing-instances corp-it interface irb.1099
set groups top routing-instances corp-it forwarding-options dhcp-relay server-group VLAN1099 192.168.10.11
set groups top routing-instances corp-it forwarding-options dhcp-relay group VLAN1099 interface irb.1099
set groups top routing-instances corp-it forwarding-options dhcp-relay group VLAN1099 active-server-group VLAN1099
set groups top routing-instances corp-it forwarding-options dhcp-relay group VLAN1099 relay-option-82 circuit-id vlan-id-only
set groups top routing-instances corp-it forwarding-options dhcp-relay group VLAN1099 relay-option-82 server-id-override
set groups top routing-instances corp-it forwarding-options dhcp-relay group VLAN1099 route-suppression destination
set groups top routing-instances corp-it forwarding-options dhcp-relay group VLAN1099 overrides relay-source lo0.1
set groups top routing-instances corp-it forwarding-options dhcp-relay forward-only
set groups top routing-instances corp-it route-distinguisher 172.16.254.8:101
set groups top routing-instances corp-it vrf-target target:65000:101
set groups top routing-instances corp-it vrf-table-label
set groups top routing-instances corp-it routing-options auto-export
set groups top routing-instances corp-it routing-options multipath
set groups top routing-instances corp-it protocols evpn ip-prefix-routes advertise direct-nexthop
set groups top routing-instances corp-it protocols evpn ip-prefix-routes encapsulation vxlan
set groups top routing-instances corp-it protocols evpn ip-prefix-routes vni 11284517
set groups top routing-instances corp-it protocols evpn ip-prefix-routes export evpn_export_type5
set groups top routing-instances corp-it interface lo0.1
#
# third VRF (includes IRB and DHCP-Relay)
set groups top routing-instances guest-wifi instance-type vrf
set groups top routing-instances guest-wifi interface irb.1033
set groups top routing-instances guest-wifi forwarding-options dhcp-relay server-group VLAN1033 192.168.10.10
set groups top routing-instances guest-wifi forwarding-options dhcp-relay group VLAN1033 interface irb.1033
set groups top routing-instances guest-wifi forwarding-options dhcp-relay group VLAN1033 active-server-group VLAN1033
set groups top routing-instances guest-wifi forwarding-options dhcp-relay group VLAN1033 relay-option-82 circuit-id vlan-id-only
set groups top routing-instances guest-wifi forwarding-options dhcp-relay group VLAN1033 relay-option-82 server-id-override
set groups top routing-instances guest-wifi forwarding-options dhcp-relay group VLAN1033 route-suppression destination
set groups top routing-instances guest-wifi forwarding-options dhcp-relay group VLAN1033 overrides relay-source lo0.3
set groups top routing-instances guest-wifi forwarding-options dhcp-relay forward-only
set groups top routing-instances guest-wifi route-distinguisher 172.16.254.8:103
set groups top routing-instances guest-wifi vrf-target target:65000:103
set groups top routing-instances guest-wifi vrf-table-label
set groups top routing-instances guest-wifi routing-options auto-export
set groups top routing-instances guest-wifi routing-options multipath
set groups top routing-instances guest-wifi protocols evpn ip-prefix-routes advertise direct-nexthop
set groups top routing-instances guest-wifi protocols evpn ip-prefix-routes encapsulation vxlan
set groups top routing-instances guest-wifi protocols evpn ip-prefix-routes vni 15560868
set groups top routing-instances guest-wifi protocols evpn ip-prefix-routes export evpn_export_type5
set groups top routing-instances guest-wifi interface lo0.3
#
# Overlay loopbacks for DHCP-Relay
set groups top interfaces lo0 unit 3 family inet address 172.16.192.6/32
set groups top interfaces lo0 unit 3 family inet6 address fd33:ab00:2::6/128
set groups top interfaces lo0 unit 2 family inet address 172.16.192.5/32
set groups top interfaces lo0 unit 2 family inet6 address fd33:ab00:2::5/128
set groups top interfaces lo0 unit 1 family inet address 172.16.192.4/32
set groups top interfaces lo0 unit 1 family inet6 address fd33:ab00:2::4/128
#
# VXLAN global settings
set groups top protocols evpn encapsulation vxlan
set groups top protocols evpn default-gateway do-not-advertise
set groups top protocols evpn extended-vni-list all
set groups top switch-options vtep-source-interface lo0.0
set groups top switch-options route-distinguisher 172.16.254.8:1
#
# VXLAN tuneing parameters based on device model
set groups top forwarding-options vxlan-routing next-hop 16384
set groups top forwarding-options vxlan-routing interface-num 6144
#
# VLAN to IRB+VNI mapping
set vlans VLAN1033 vlan-id 1033
set vlans VLAN1033 l3-interface irb.1033
set vlans VLAN1033 vxlan vni 11033
set vlans VLAN1088 vlan-id 1088
set vlans VLAN1088 l3-interface irb.1088
set vlans VLAN1088 vxlan vni 11088
set vlans VLAN1099 vlan-id 1099
set vlans VLAN1099 l3-interface irb.1099
set vlans VLAN1099 vxlan vni 11099
set vlans default vlan-id 1
set vlans default l3-interface irb.0
set vlans default vxlan vni 10001
```

Access2 Switch Junos Configuration

```
              # global system housekeeping
set system host-name access2
set system time-zone UTC
set system commit synchronize
set protocols lldp interface all
set protocols lldp port-id-subtype interface-name
set protocols lldp port-description-type interface-alias
set protocols lldp-med interface all
set protocols rstp interface all
set protocols rstp bpdu-block-on-edge
set groups top system commit no-delta-synchronize
set groups top system name-server 8.8.8.8
set groups top system name-server 9.9.9.9
set groups top system ntp server 192.168.10.1
set groups top system syslog file messages authorization any
set groups top system syslog file messages archive files 5
set groups top system syslog file messages archive size 2m
set groups top system syslog file interactive-commands match "!(.*mist.*)"
set groups top system syslog file interactive-commands archive files 5
set groups top system syslog file interactive-commands archive size 2m
set groups top system syslog file escript.log archive files 5
set groups top system syslog file escript.log archive size 2m
set groups top system syslog file op-script.log archive files 5
set groups top system syslog file op-script.log archive size 2m
set groups top system syslog file snapshot archive files 5
set groups top system syslog file snapshot archive size 2m
set apply-groups top
#
# virtual chassis configuration
set protocols layer2-control nonstop-bridging
delete virtual-chassis
set virtual-chassis preprovisioned
set virtual-chassis member 0 role routing-engine serial-number ZG4723340069
set virtual-chassis member 1 role line-card serial-number ZG4723350034
set virtual-chassis member 2 role routing-engine serial-number ZG4723350187
set virtual-chassis member 3 role line-card serial-number ZF4321500037
set chassis redundancy graceful-switchover
set routing-options nonstop-routing
```

```
#
# up or downlink interfaces to other fabric nodes
set interfaces xe-2/2/0 unit 0 family inet address 10.255.240.19/31
set interfaces xe-2/2/0 description evpn_uplink-to-d8539a64a6c0
set interfaces xe-1/2/0 unit 0 family inet address 10.255.240.23/31
set interfaces xe-1/2/0 description evpn_uplink-to-d8539a6519c0
#
# Underlay Loopback interface, router ID, and AS number
set groups top interfaces lo0 unit 0 family inet address 172.16.254.7/32
set groups top routing-options router-id 172.16.254.7
set groups top routing-options autonomous-system 65007
#
# Per-packet load balancing
set groups top policy-options policy-statement ecmp_policy then load-balance per-packet
set groups top policy-options policy-statement ecmp_policy then accept
set groups top routing-options forwarding-table export ecmp_policy
set routing-options forwarding-table ecmp-fast-reroute
set routing-options forwarding-table chained-composite-next-hop ingress evpn
set groups top forwarding-options vxlan-routing overlay-ecmp
#
# BGP underlay network to other fabric nodes
set groups top policy-options policy-statement evpn_underlay_export term 01-loopback from route-filter 172.16.254.0/23 orlonger
set groups top policy-options policy-statement evpn_underlay_export term 01-loopback then accept
set groups top policy-options policy-statement evpn_underlay_export term 02-default then reject
set groups top policy-options policy-statement evpn_underlay_import term 01-loopback from route-filter 172.16.254.0/23 orlonger
set groups top policy-options policy-statement evpn_underlay_import term 01-loopback then accept
set groups top policy-options policy-statement evpn_underlay_import term 02-default then reject
set protocols bgp group evpn_underlay type external
set protocols bgp group evpn_underlay local-as 65007
set protocols bgp group evpn_underlay multipath multiple-as
set protocols bgp group evpn_underlay authentication-key <not-disclosed-here>
set protocols bgp group evpn_underlay family inet unicast
set protocols bgp group evpn_underlay bfd-liveness-detection minimum-interval 1000
set protocols bgp group evpn_underlay bfd-liveness-detection multiplier 3
set protocols bgp group evpn_underlay log-updown
set protocols bgp group evpn_underlay export evpn_underlay_export
set protocols bgp group evpn_underlay import evpn_underlay_import
set protocols bgp group evpn_underlay neighbor 10.255.240.18 peer-as 65005
set protocols bgp group evpn_underlay neighbor 10.255.240.22 peer-as 65006
set protocols bgp graceful-restart
#
```

```
# EVPN signalling to other fabric nodes
set groups top switch-options vrf-target target:65000:1
set protocols bgp group evpn_overlay type external
set protocols bgp group evpn_overlay local-address 172.16.254.7
set protocols bgp group evpn_overlay local-as 65007
set protocols bgp group evpn_overlay multipath multiple-as
set protocols bgp group evpn_overlay authentication-key <not-disclosed-here>
set protocols bgp group evpn_overlay family evpn signaling loops 2
set protocols bgp group evpn_overlay log-updown
set protocols bgp group evpn_overlay bfd-liveness-detection minimum-interval 1000
set protocols bgp group evpn_overlay bfd-liveness-detection multiplier 3
set protocols bgp group evpn_overlay bfd-liveness-detection session-mode automatic
set protocols bgp group evpn_overlay multihop ttl 1
set protocols bgp group evpn_overlay multihop no-nexthop-change
set protocols bgp group evpn_overlay neighbor 172.16.254.5 peer-as 65005
set protocols bgp group evpn_overlay neighbor 172.16.254.6 peer-as 65006
#
# EVPN type2/5 coexistence
set groups top policy-options policy-statement evpn_export_type5 term 01_ipv4 from protocol evpn
set groups top policy-options policy-statement evpn_export_type5 term 01_ipv4 from route-filter 0.0.0.0/0 prefix-length-range /32-/32
set groups top policy-options policy-statement evpn_export_type5 term 01_ipv4 then accept
set groups top policy-options policy-statement evpn_export_type5 term 02_ipv6 from protocol evpn
set groups top policy-options policy-statement evpn_export_type5 term 02_ipv6 from family inet6
set groups top policy-options policy-statement evpn_export_type5 term 02_ipv6 from route-filter 0::0/0 prefix-length-range /128-/128
set groups top policy-options policy-statement evpn_export_type5 term 02_ipv6 then accept
set groups top policy-options policy-statement evpn_export_type5 term 03_direct from protocol direct
set groups top policy-options policy-statement evpn_export_type5 term 03_direct then accept
#
# interface housekeeping
set interfaces interface-range default apply-groups default
set interfaces interface-range default member et-0/1/[0-3]
set interfaces interface-range default member mge-0/0/[0-23]
set interfaces interface-range default member et-1/1/[0-3]
set interfaces interface-range default member mge-1/0/[0-23]
set interfaces interface-range default member et-2/1/[0-3]
set interfaces interface-range default member mge-2/0/[0-23]
set interfaces interface-range default member et-3/1/[0-3]
set interfaces interface-range default member mge-3/0/[0-11]
set interfaces interface-range default member mge-3/0/[14-15]
set interfaces interface-range default member mge-3/0/[17-47]
set interfaces interface-range vlan1088-no-auth apply-groups vlan1088-no-auth
set interfaces interface-range vlan1088-no-auth member mge-3/0/12
set interfaces interface-range vlan1099-no-auth apply-groups vlan1099-no-auth
set interfaces interface-range vlan1099-no-auth member mge-3/0/13
set interfaces interface-range access-point apply-groups access-point
set interfaces interface-range access-point member mge-3/0/16
set interfaces interface-range evpn_uplink apply-groups evpn_uplink
set interfaces interface-range evpn_uplink member xe-2/2/0
set interfaces interface-range evpn_uplink member xe-1/2/0
set interfaces vme unit 0 family inet dhcp vendor-id Juniper
set interfaces vme unit 0 family inet dhcp force-discover
set interfaces vme unit 0 family inet dhcp retransmission-attempt 60
set interfaces vme unit 0 family inet dhcp client-identifier user-id ascii bc0ffe157080-M4aLquH9
set interfaces irb unit 0 family inet dhcp vendor-id Juniper
set interfaces irb unit 0 family inet dhcp force-discover
set interfaces irb unit 0 family inet dhcp retransmission-attempt 60
set interfaces irb unit 0 family inet dhcp client-identifier user-id ascii bc0ffe157080-0
set interfaces irb unit 0 family inet mtu 9000
set interfaces irb unit 0 description default
set groups default interfaces <*> unit 0 family ethernet-switching vlan members [ default ]
set groups vlan1088-no-auth interfaces <*> unit 0 family ethernet-switching vlan members
[ VLAN1088 ]
set groups vlan1099-no-auth interfaces <*> unit 0 family ethernet-switching vlan members
[ VLAN1099 ]
set groups access-point interfaces <*> unit 0 family ethernet-switching interface-mode trunk
set groups access-point interfaces <*> unit 0 family ethernet-switching vlan members [ all ]
set groups access-point interfaces <*> native-vlan-id 1033
set groups evpn_uplink interfaces <*> mtu 9192
set groups top poe interface all
set groups top forwarding-options storm-control-profiles default all
#
# IRB's and default GW's for VLANs
set interfaces irb unit 1033 family inet address 10.33.33.1/24
set interfaces irb unit 1033 family inet mtu 9000
set interfaces irb unit 1033 description VLAN1033
set interfaces irb unit 1033 no-dhcp-flood
set interfaces irb unit 1033 mac 00:00:5e:e4:31:57
set interfaces irb unit 1088 family inet address 10.88.88.1/24
set interfaces irb unit 1088 family inet mtu 9000
set interfaces irb unit 1088 description VLAN1088
set interfaces irb unit 1088 no-dhcp-flood
set interfaces irb unit 1088 mac 00:00:5e:e4:31:57
set interfaces irb unit 1099 family inet address 10.99.99.1/24
set interfaces irb unit 1099 family inet mtu 9000
set interfaces irb unit 1099 description VLAN1099
set interfaces irb unit 1099 no-dhcp-flood
set interfaces irb unit 1099 mac 00:00:5e:e4:31:57
#
# first VRF (includes IRB and DHCP-Relay)
set groups top routing-instances corp-it instance-type vrf
set groups top routing-instances corp-it interface irb.1099
set groups top routing-instances corp-it forwarding-options dhcp-relay server-group VLAN1099 192.168.10.11
set groups top routing-instances corp-it forwarding-options dhcp-relay group VLAN1099 interface irb.1099
set groups top routing-instances corp-it forwarding-options dhcp-relay group VLAN1099 active-server-group VLAN1099
set groups top routing-instances corp-it forwarding-options dhcp-relay group VLAN1099 relay-option-82 circuit-id vlan-id-only
set groups top routing-instances corp-it forwarding-options dhcp-relay group VLAN1099 relay-option-82 server-id-override
set groups top routing-instances corp-it forwarding-options dhcp-relay group VLAN1099 route-suppression destination
set groups top routing-instances corp-it forwarding-options dhcp-relay group VLAN1099 overrides relay-source lo0.1
set groups top routing-instances corp-it forwarding-options dhcp-relay forward-only
set groups top routing-instances corp-it route-distinguisher 172.16.254.7:101
set groups top routing-instances corp-it vrf-target target:65000:101
set groups top routing-instances corp-it vrf-table-label
set groups top routing-instances corp-it routing-options auto-export
set groups top routing-instances corp-it routing-options multipath
set groups top routing-instances corp-it protocols evpn ip-prefix-routes advertise direct-nexthop
set groups top routing-instances corp-it protocols evpn ip-prefix-routes encapsulation vxlan
set groups top routing-instances corp-it protocols evpn ip-prefix-routes vni 11284517
set groups top routing-instances corp-it protocols evpn ip-prefix-routes export evpn_export_type5
set groups top routing-instances corp-it interface lo0.1
#
# second VRF (includes IRB and DHCP-Relay)
set groups top routing-instances developers instance-type vrf
set groups top routing-instances developers interface irb.1088
set groups top routing-instances developers forwarding-options dhcp-relay server-group VLAN1088 192.168.10.11
set groups top routing-instances developers forwarding-options dhcp-relay group VLAN1088 interface irb.1088
set groups top routing-instances developers forwarding-options dhcp-relay group VLAN1088 active-server-group VLAN1088
set groups top routing-instances developers forwarding-options dhcp-relay group VLAN1088 relay-option-82 circuit-id vlan-id-only
set groups top routing-instances developers forwarding-options dhcp-relay group VLAN1088 relay-option-82 server-id-override
set groups top routing-instances developers forwarding-options dhcp-relay group VLAN1088 route-suppression destination
set groups top routing-instances developers forwarding-options dhcp-relay group VLAN1088 overrides relay-source lo0.2
set groups top routing-instances developers forwarding-options dhcp-relay forward-only
set groups top routing-instances developers route-distinguisher 172.16.254.7:102
set groups top routing-instances developers vrf-target target:65000:102
set groups top routing-instances developers vrf-table-label
set groups top routing-instances developers routing-options auto-export
set groups top routing-instances developers routing-options multipath
set groups top routing-instances developers protocols evpn ip-prefix-routes advertise direct-nexthop
set groups top routing-instances developers protocols evpn ip-prefix-routes encapsulation vxlan
set groups top routing-instances developers protocols evpn ip-prefix-routes vni 15600414
set groups top routing-instances developers protocols evpn ip-prefix-routes export evpn_export_type5
set groups top routing-instances developers interface lo0.2
#
# third VRF (includes IRB and DHCP-Relay)
set groups top routing-instances guest-wifi instance-type vrf
set groups top routing-instances guest-wifi interface irb.1033
set groups top routing-instances guest-wifi forwarding-options dhcp-relay server-group VLAN1033 192.168.10.10
set groups top routing-instances guest-wifi forwarding-options dhcp-relay group VLAN1033 interface irb.1033
set groups top routing-instances guest-wifi forwarding-options dhcp-relay group VLAN1033 active-server-group VLAN1033
set groups top routing-instances guest-wifi forwarding-options dhcp-relay group VLAN1033 relay-option-82 circuit-id vlan-id-only
set groups top routing-instances guest-wifi forwarding-options dhcp-relay group VLAN1033 relay-option-82 server-id-override
set groups top routing-instances guest-wifi forwarding-options dhcp-relay group VLAN1033 route-suppression destination
set groups top routing-instances guest-wifi forwarding-options dhcp-relay group VLAN1033 overrides relay-source lo0.3
set groups top routing-instances guest-wifi forwarding-options dhcp-relay forward-only
set groups top routing-instances guest-wifi route-distinguisher 172.16.254.7:103
set groups top routing-instances guest-wifi vrf-target target:65000:103
set groups top routing-instances guest-wifi vrf-table-label
set groups top routing-instances guest-wifi routing-options auto-export
set groups top routing-instances guest-wifi routing-options multipath
set groups top routing-instances guest-wifi protocols evpn ip-prefix-routes advertise direct-nexthop
set groups top routing-instances guest-wifi protocols evpn ip-prefix-routes encapsulation vxlan
set groups top routing-instances guest-wifi protocols evpn ip-prefix-routes vni 15560868
set groups top routing-instances guest-wifi protocols evpn ip-prefix-routes export evpn_export_type5
set groups top routing-instances guest-wifi interface lo0.3
#
# Overlay loopbacks for DHCP-Relay
set groups top interfaces lo0 unit 3 family inet address 172.16.192.9/32
set groups top interfaces lo0 unit 3 family inet6 address fd33:ab00:2::9/128
set groups top interfaces lo0 unit 2 family inet address 172.16.192.8/32
set groups top interfaces lo0 unit 2 family inet6 address fd33:ab00:2::8/128
set groups top interfaces lo0 unit 1 family inet address 172.16.192.7/32
set groups top interfaces lo0 unit 1 family inet6 address fd33:ab00:2::7/128
#
# VXLAN global settings
set groups top protocols evpn encapsulation vxlan
set groups top protocols evpn default-gateway do-not-advertise
set groups top protocols evpn extended-vni-list all
set groups top switch-options vtep-source-interface lo0.0
set groups top switch-options route-distinguisher 172.16.254.7:1
#
# VXLAN tuneing parameters based on device model
set groups top forwarding-options vxlan-routing next-hop 16384
set groups top forwarding-options vxlan-routing interface-num 6144
#
# VLAN to IRB+VNI mapping
set vlans VLAN1033 vlan-id 1033
set vlans VLAN1033 l3-interface irb.1033
set vlans VLAN1033 vxlan vni 11033
set vlans VLAN1088 vlan-id 1088
set vlans VLAN1088 l3-interface irb.1088
set vlans VLAN1088 vxlan vni 11088
set vlans VLAN1099 vlan-id 1099
set vlans VLAN1099 l3-interface irb.1099
set vlans VLAN1099 vxlan vni 11099
set vlans default vlan-id 1
set vlans default l3-interface irb.0
set vlans default vxlan vni 10001
#
```

```
# additional CLI
set chassis fpc 0 pic 2 port 0 speed 10g
set chassis fpc 1 pic 2 port 0 speed 10g
set chassis fpc 2 pic 2 port 0 speed 10g
set chassis fpc 3 pic 2 port 0 speed 10g
```

WAN-Router Junos Configuration

We obtained the following example configuration from the SRX1500 as WAN-Router.

```
root@wanrouter> show configuration | display set | no-more
# global system housekeeping
set system host-name wanrouter
set system services ssh root-login allow
set system services telnet
set system services netconf ssh
set system name-server 8.8.8.8
set system name-server 8.8.4.4
set system syslog user * any emergency
set system syslog file interactive-commands interactive-commands any
set system syslog file messages any notice
set system syslog file messages authorization info
set system max-configurations-on-flash 5
set system license autoupdate url https://ae1.juniper.net/junos/key_retrieval
set security log mode stream
set protocols lldp port-id-subtype interface-name
set protocols lldp port-description-type interface-alias
set protocols lldp interface all
set protocols lldp-med interface all
#
# default screen setting
set security screen ids-option untrust-screen icmp ping-death
set security screen ids-option untrust-screen ip source-route-option
set security screen ids-option untrust-screen ip tear-drop
set security screen ids-option untrust-screen tcp syn-flood alarm-threshold 1024
set security screen ids-option untrust-screen tcp syn-flood attack-threshold 200
set security screen ids-option untrust-screen tcp syn-flood source-threshold 1024
set security screen ids-option untrust-screen tcp syn-flood destination-threshold 2048
set security screen ids-option untrust-screen tcp syn-flood timeout 20
set security screen ids-option untrust-screen tcp land
#
# SNAT with exclusion of 192.168.10.0/24 to DHCP-Server
set security nat source rule-set trust-to-untrust from zone trust
set security nat source rule-set trust-to-untrust to zone untrust
set security nat source rule-set trust-to-untrust rule source-nat-rule1 match destination-address 192.168.10.0/24
set security nat source rule-set trust-to-untrust rule source-nat-rule1 then source-nat off
set security nat source rule-set trust-to-untrust rule source-nat-rule2 match source-address 0.0.0.0/0
set security nat source rule-set trust-to-untrust rule source-nat-rule2 then source-nat interface
#
# default trust and untrust zone communication
set security policies from-zone trust to-zone trust policy default-permit match source-address any
set security policies from-zone trust to-zone trust policy default-permit match destination-address any
set security policies from-zone trust to-zone trust policy default-permit match application any
set security policies from-zone trust to-zone trust policy default-permit then permit
set security policies from-zone trust to-zone untrust policy default-permit match source-address any
set security policies from-zone trust to-zone untrust policy default-permit match destination-address any
set security policies from-zone trust to-zone untrust policy default-permit match application any
set security policies from-zone trust to-zone untrust policy default-permit then permit
set security policies pre-id-default-policy then log session-close
#
# bind interfaces to security zones
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces xe-0/0/16.1099
set security zones security-zone trust interfaces xe-0/0/16.1088
set security zones security-zone trust interfaces xe-0/0/16.1033
set security zones security-zone trust interfaces xe-0/0/17.1099
set security zones security-zone trust interfaces xe-0/0/17.1088
set security zones security-zone trust interfaces xe-0/0/17.1033
set security zones security-zone untrust screen untrust-screen
set security zones security-zone untrust host-inbound-traffic system-services all
set security zones security-zone untrust host-inbound-traffic protocols all
set security zones security-zone untrust interfaces ge-0/0/0.0
#
# uplink to lab and downlink interfaces to fabric nodes
set interfaces ge-0/0/0 unit 0 family inet address 192.168.10.99/24
set interfaces xe-0/0/16 flexible-vlan-tagging
set interfaces xe-0/0/16 mtu 9014
set interfaces xe-0/0/16 unit 1033 description vlan1033
set interfaces xe-0/0/16 unit 1033 vlan-id 1033
set interfaces xe-0/0/16 unit 1033 family inet address 10.255.224.4/31
set interfaces xe-0/0/16 unit 1088 description vlan1088
set interfaces xe-0/0/16 unit 1088 vlan-id 1088
set interfaces xe-0/0/16 unit 1088 family inet address 10.255.224.2/31
set interfaces xe-0/0/16 unit 1099 description vlan1099
set interfaces xe-0/0/16 unit 1099 vlan-id 1099
set interfaces xe-0/0/16 unit 1099 family inet address 10.255.224.0/31
set interfaces xe-0/0/17 flexible-vlan-tagging
set interfaces xe-0/0/17 mtu 9014
set interfaces xe-0/0/17 unit 1033 description vlan1033
set interfaces xe-0/0/17 unit 1033 vlan-id 1033
set interfaces xe-0/0/17 unit 1033 family inet address 10.255.226.4/31
set interfaces xe-0/0/17 unit 1088 description vlan1088
set interfaces xe-0/0/17 unit 1088 vlan-id 1088
set interfaces xe-0/0/17 unit 1088 family inet address 10.255.226.2/31
set interfaces xe-0/0/17 unit 1099 description vlan1099
set interfaces xe-0/0/17 unit 1099 vlan-id 1099
set interfaces xe-0/0/17 unit 1099 family inet address 10.255.226.0/31
#
# Per-packet load balancing
set routing-options forwarding-table export ECMP
set policy-options policy-statement ECMP then load-balance per-packet
set policy-options policy-statement ECMP then accept
#
# filters to announce default route and import fabric routes
set policy-options policy-statement fabric term 1 from protocol bgp
set policy-options policy-statement fabric term 1 from route-filter 0.0.0.0/0 orlonger
set policy-options policy-statement fabric term 1 then accept
set policy-options policy-statement fabric term 2 then reject
set policy-options policy-statement internet term 1 from protocol static
set policy-options policy-statement internet term 1 from route-filter 0.0.0.0/0 exact
set policy-options policy-statement internet term 1 then accept
set policy-options policy-statement internet term 2 then reject
#
# virtual router with up/downlink interfaces
set routing-instances public-int instance-type virtual-router
set routing-instances public-int interface ge-0/0/0.0
set routing-instances public-int interface xe-0/0/16.1033
set routing-instances public-int interface xe-0/0/16.1088
set routing-instances public-int interface xe-0/0/16.1099
set routing-instances public-int interface xe-0/0/17.1033
set routing-instances public-int interface xe-0/0/17.1088
set routing-instances public-int interface xe-0/0/17.1099
set routing-instances public-int routing-options static route 0.0.0.0/0 next-hop 192.168.10.1
#
# BGP peering to first fabric VRF neigbours
set routing-instances public-int protocols bgp group corp-it type external
set routing-instances public-int protocols bgp group corp-it hold-time 90
set routing-instances public-int protocols bgp group corp-it import fabric
set routing-instances public-int protocols bgp group corp-it family inet unicast
set routing-instances public-int protocols bgp group corp-it export internet
set routing-instances public-int protocols bgp group corp-it local-as 64901
set routing-instances public-int protocols bgp group corp-it multipath multiple-as
set routing-instances public-int protocols bgp group corp-it bfd-liveness-detection minimum-interval 1000
set routing-instances public-int protocols bgp group corp-it bfd-liveness-detection multiplier 3
set routing-instances public-int protocols bgp group corp-it bfd-liveness-detection session-mode automatic
set routing-instances public-int protocols bgp group corp-it neighbor 10.255.224.1 peer-as 64911
set routing-instances public-int protocols bgp group corp-it neighbor 10.255.226.1 peer-as 64911
#
# BGP peering to second fabric VRF neigbours
set routing-instances public-int protocols bgp group developers type external
set routing-instances public-int protocols bgp group developers hold-time 90
set routing-instances public-int protocols bgp group developers import fabric
set routing-instances public-int protocols bgp group developers family inet unicast
set routing-instances public-int protocols bgp group developers export internet
set routing-instances public-int protocols bgp group developers local-as 64901
set routing-instances public-int protocols bgp group developers multipath multiple-as
set routing-instances public-int protocols bgp group developers bfd-liveness-detection minimum-interval 1000
set routing-instances public-int protocols bgp group developers bfd-liveness-detection multiplier 3
set routing-instances public-int protocols bgp group developers bfd-liveness-detection session-mode automatic
set routing-instances public-int protocols bgp group developers neighbor 10.255.224.3 peer-as 64911
set routing-instances public-int protocols bgp group developers neighbor 10.255.226.3 peer-as 64911
#
# BGP peering to third fabric VRF neigbours
set routing-instances public-int protocols bgp group guest-wifi type external
set routing-instances public-int protocols bgp group guest-wifi hold-time 90
set routing-instances public-int protocols bgp group guest-wifi import fabric
set routing-instances public-int protocols bgp group guest-wifi family inet unicast
set routing-instances public-int protocols bgp group guest-wifi export internet
set routing-instances public-int protocols bgp group guest-wifi local-as 64901
set routing-instances public-int protocols bgp group guest-wifi multipath multiple-as
set routing-instances public-int protocols bgp group guest-wifi bfd-liveness-detection minimum-interval 1000
set routing-instances public-int protocols bgp group guest-wifi bfd-liveness-detection multiplier 3
set routing-instances public-int protocols bgp group guest-wifi bfd-liveness-detection session-mode automatic
set routing-instances public-int protocols bgp group guest-wifi neighbor 10.255.224.5 peer-as 64911
set routing-instances public-int protocols bgp group guest-wifi neighbor 10.255.226.5 peer-as 64911
```

APPENDIX: Mist Edge Integration into IP-Clos Fabric (Optional)

IN THIS SECTION

An enterprise Wi-Fi access point (AP) can be thought of as an L2 MAC bridge. One side communicates wirelessly with client devices, while the other side connects to the wired Ethernet network. Because of this, APs do not perform L3 gateway functions for a VLAN. Instead, it is expected that somewhere in the network—at the point where the AP's Ethernet port connects—the L3 default gateway for the VLAN is provided. In most cases this means:

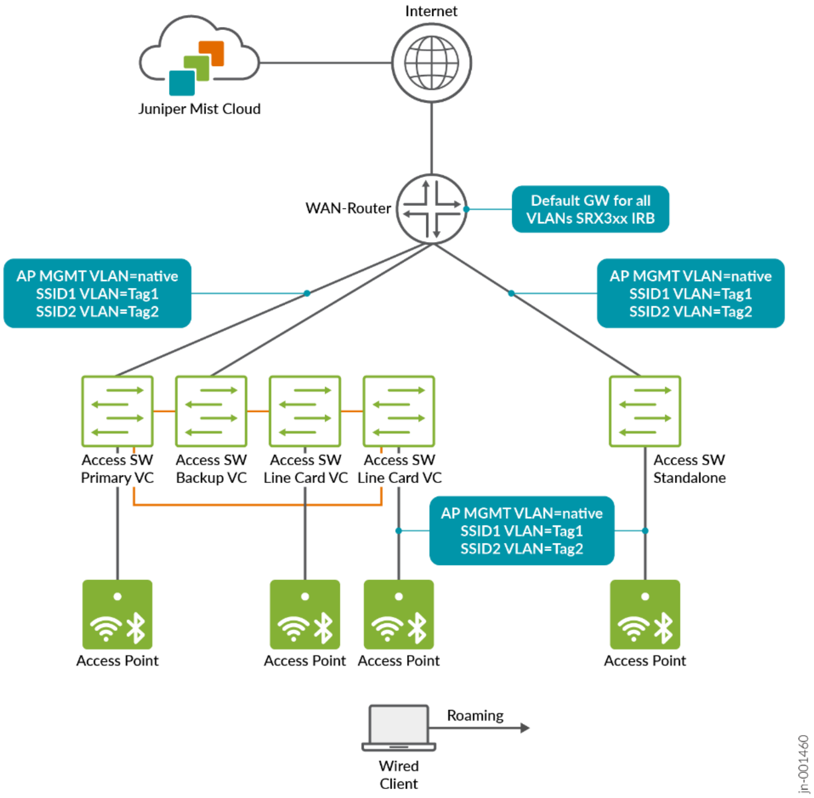
- In simple branch deployments, the Layer 3 gateway for a VLAN is typically located on the WAN router. Both switches and APs connect to this router, and in many cases the DHCP server function is also hosted on the WAN router.

- In campus fabric deployments, the Layer 3 gateway for a VLAN resides in the VRF of the fabric. The exact placement of the VRF depends on the campus fabric design, but in these environments, the fabric usually performs only a DHCP relay function rather than acting as the DHCP server itself.

Juniper Mist Wi-Fi supports two primary models for forwarding traffic from APs into the network:

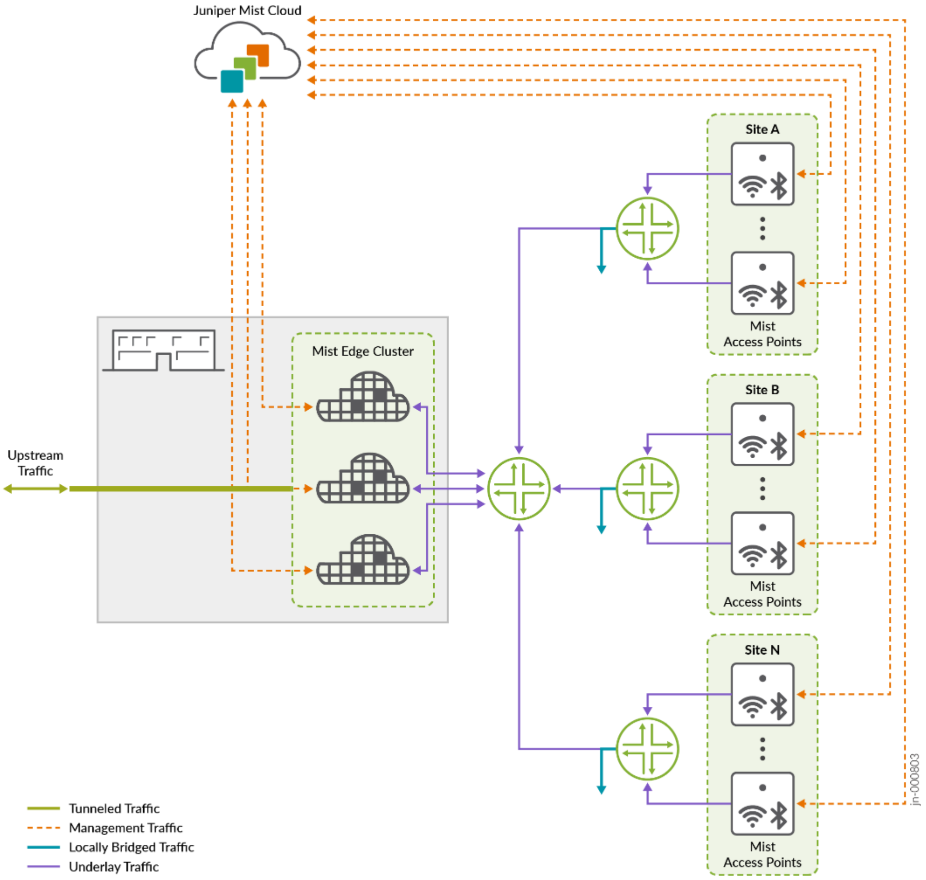
- The first is a distributed model, in which wireless client traffic is broken out locally at the AP. In this approach, the AP broadcasts an SSID and associates clients with it, then maps their traffic to a tagged VLAN on the Ethernet side of the AP. In some cases, client traffic may also be switched directly to another device connected to the same AP and SSID, without leaving the AP.

Figure 77: Wireless Client Roaming at Branch



- The second option is a centralized, or tunneled, approach where wireless client traffic is broken out remotely at a central traffic Data Plane Concentrator known as Mist Edge. In this model, the access point establishes an overlay transport tunnel to the tunneling endpoint using the tunnelling head end's remote IP address. The tunnel can operate across any routed network or VLAN, providing flexibility in deployment. For the tunnel, Juniper Mist uses the L2TPv3 protocol, which can be further secured by enabling IPsec protection. Once the traffic reaches the tunnelling head end, it is broken out into the appropriate VLANs as configured, typically based on the SSID settings. This approach centralizes traffic handling and allows for consistent policy enforcement at a single point in the network.

Figure 78: Mist-Edge Central Data Plane Concentrator



When the centralized approach is used with an EVPN fabric such as campus fabric IP Clos, we recommend the following setup:

- Configure an AP overlay network for campus fabric where:
  - The AP gets a DHCP lease using DHCP relay on the fabric.
  - After getting an IP address, the AP reaches out to the Juniper Mist cloud to get managed.

  - After the tunnel configuration is applied, an unsecured L2TPv3 tunnel can be built to the Mist Edge device's tunnel termination IP address in the same VLAN.

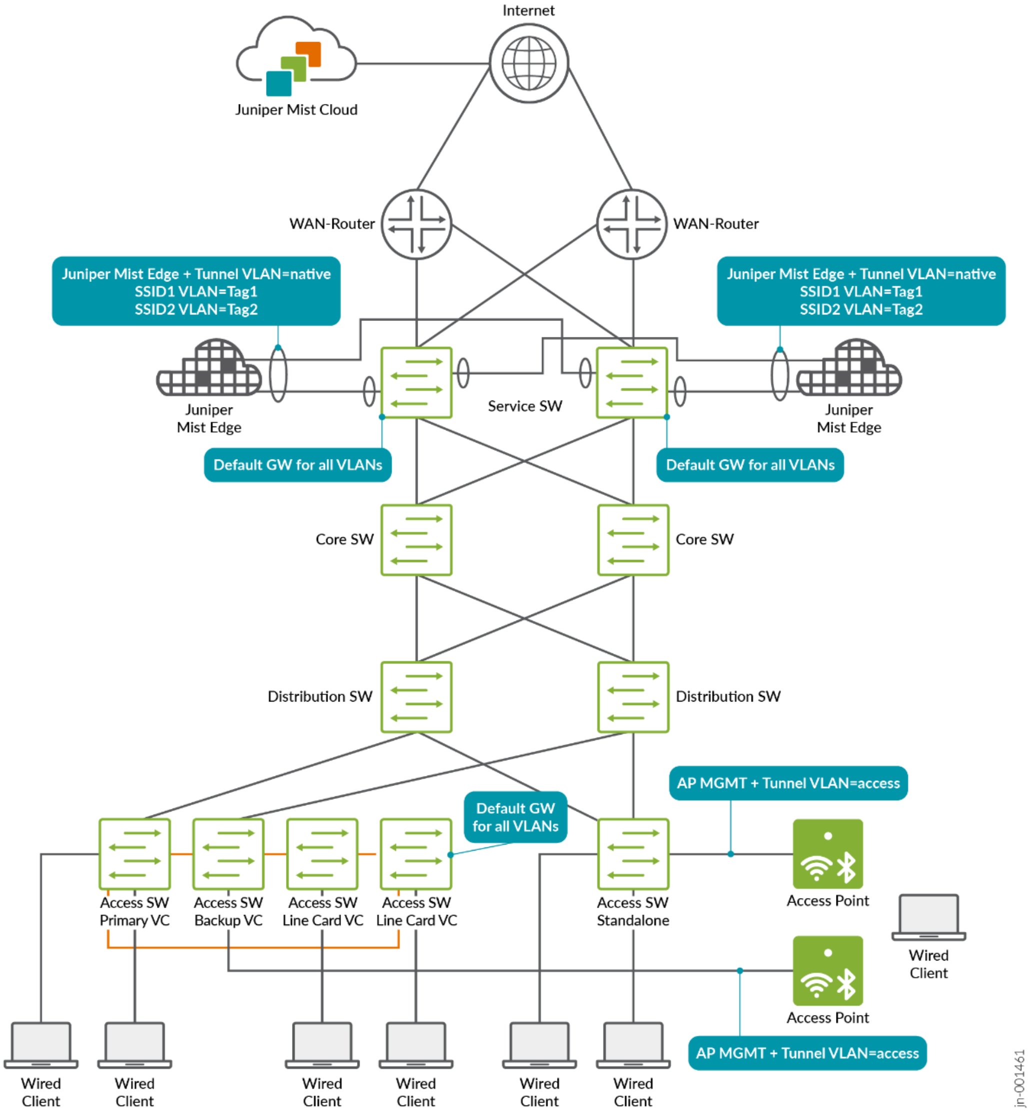
- Configure individual wireless client breakout overlay networks (similar to regular wired networks) for campus fabric according to your requirements and make sure that:
  - The wireless clients mapped to these networks are getting DHCP leases using DHCP relay on the fabric.

- Attach your AP at the access switch and power them up.
  - Only the AP overlay network needs to be configured as access port type.

- Attach at least two (for redundancy) Mist Edge devices at the top of the EVPN fabric at the service block function which is usually the integration point for other network services like the WAN router, firewalls and servers.
  - Each Mist Edge must have an interface to each service block function.
  - On the fabric side towards each Mist Edge, configure a unique ESI-LAG.
  - The Mist Edge interfaces (apart from the OOBM) are using a simple LAG towards the fabric.
  - On both sides, the AP overlay network is configured as a native VLAN without tags.
  - On both sides, the individual wireless client breakout overlay networks are configured as tagged VLANs.

This recommended configuration enables the EVPN fabric to gather wireless client traffic from the APs through the tunnels at the top of the fabric and then reinject it back into the network. This approach ensures seamless communication between wireless and wired clients, while also supporting traffic flows to the Internet outside of the EVPN fabric.

Figure 79: Recommended Mist Edge Integration into EVPN Fabric



The choice of which approach to use should be determined during the design phase of an EVPN fabric. The centralized approach with Mist Edges in an EVPN fabric is recommended when the following conditions apply:

- More than 2.000 wireless clients are expected in the EVPN fabric.
- You have an expectation of “fast” roaming wireless clients through the APs of the network.

NOTE: It is important to remember that the decision to roam is ultimately made by the wireless client itself. Unlike cellular networks, where the base station controls and can trigger roaming, Wi-Fi does not have this capability built into its standard functions. In Wi-Fi, the AP network can only attempt to encourage or guide the client to roam, but the final decision always rests with the client device. Because there are many different Wi-Fi network interface cards (NICs) on the market, each with its own firmware and internal configuration, predicting how a specific client will behave when roaming is extremely difficult. As a result, roaming behavior can vary significantly between devices.When using the centralized model with Juniper Mist Edge, it is considered best practice to use different VLANs for client access at the switch layer than those used for breakout at the Juniper Mist Edge service block at the top of the EVPN fabric. These VLANs may reside within the same VRF but separating them into distinct L2 domains is recommended. If both layers share the same L2 domain, all MAC addresses must be learned by both the access layer and the service block, which can affect scalability. Using separate L2 domains avoids this issue. Refer to the next section for more detailed guidance.

Design for High Scale of Wired and Wireless Clients with EVPN Fabrics

With the right design, using Juniper Mist Edge allows you to support a larger number of wired and wireless clients within a single fabric. To accomplish this, the VLANs used for wired clients at the access layer should be separated from the VLANs used for wireless clients, and the locations of their default gateways (VRFs) should reside on different layers of the fabric, handled by different fabric switches. This capability is planned as a future enhancement that will be applied automatically when creating a new campus fabric.

For now, contact your Juniper representative if you need these details in advance. Once the new feature is available, its announcement will be posted here for review. In the meantime, it is helpful to understand the required design changes ahead of time, so you are prepared when the enhancement becomes available.

The recommended VLAN structure for achieving independent, high-scale client designs is as follows:

1. Plan for one or more VLANs used by APs to obtain DHCP leases, connect to the Juniper Mist cloud, and establish tunnels to the Juniper Mist Edge. These VLANs should exist only at the access switch and are typically configured as native VLANs on the switch ports.

2. Plan for a small VLAN that provides the tunnel termination IP address for the Juniper Mist Edges. APs send remotely-broken-out wireless client traffic to this address. Deploy this VLAN only at the service block where the Juniper Mist Edge is connected, usually as the native VLAN on the Juniper Mist Edge's LAG trunk. Make sure the AP tunnel VLANs and the Mist Edge tunnel VLAN are in the

same VRF; otherwise, traffic will be forced through a WAN router due to VRF isolation. This separation helps ensure that even with many APs, the Juniper Mist Edge only needs to track the default gateway's local IP address instead of maintaining ARP entries for every AP.

3. Plan for one or more VLANs that receive Wi-Fi client traffic forwarded by the AP and handed off at the Juniper Mist Edge. These VLANs are typically trunked and carried across the Juniper Mist Edge's LAG interfaces into the EVPN fabric via ESI-LAG. They should only be configured at the service block to maintain the intended scale separation.

4. Plan for one or more VLANs dedicated to wired client traffic. These VLANs should be defined only at the EVPN fabric access layer. If you expect communication between wired and wireless clients, consider placing their VLANs in the same VRF so that traffic can be exchanged within the EVPN fabric.

5. If additional infrastructure services (such as servers) are connected at the service block, ensure they use their own VLANs that exist only at the service block where the wireless client VLANs reside. Using the same VRF is again recommended when possible.

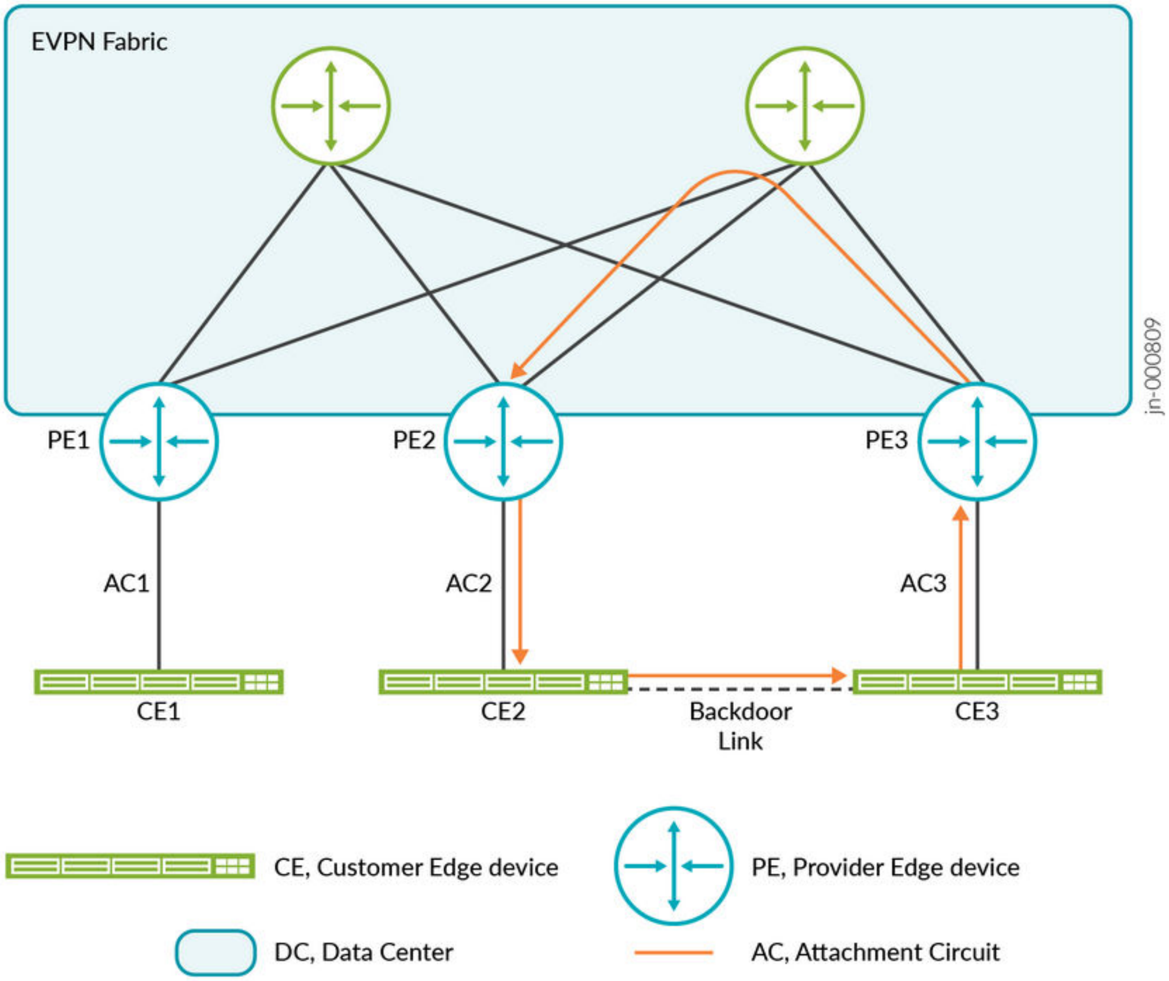
Further details on scaling Juniper Mist Edges are covered in the following sections.

Wireless Client Roaming in EVPN Fabrics

The concepts behind MAC address mobility in EVPN networks are explained in the following link.

By default, EVPN fabrics typically treat MAC address movement as unusual behavior, since with wired clients this usually indicates something abnormal that should not occur. In the example below, a link between two fabric leaf switches could create a loop. To prevent this, one of the strategies used in EVPN fabrics is to detect when the same MAC address appears on multiple links. If this happens, the MAC address may be automatically blocklisted, and traffic associated with it will no longer be forwarded.

Figure 80: EVPN Fabric Duplicate MAC-Address Detection



While automatic blocklisting of duplicate MAC addresses is useful for wired clients, it can create problems for wireless clients. Normal roaming behavior, where a client moves between APs connected to different leaf nodes in the EVPN fabric, may appear as a duplicate MAC event. This can cause the freshly roamed MAC address to be blocklisted, even though the behavior is legitimate.

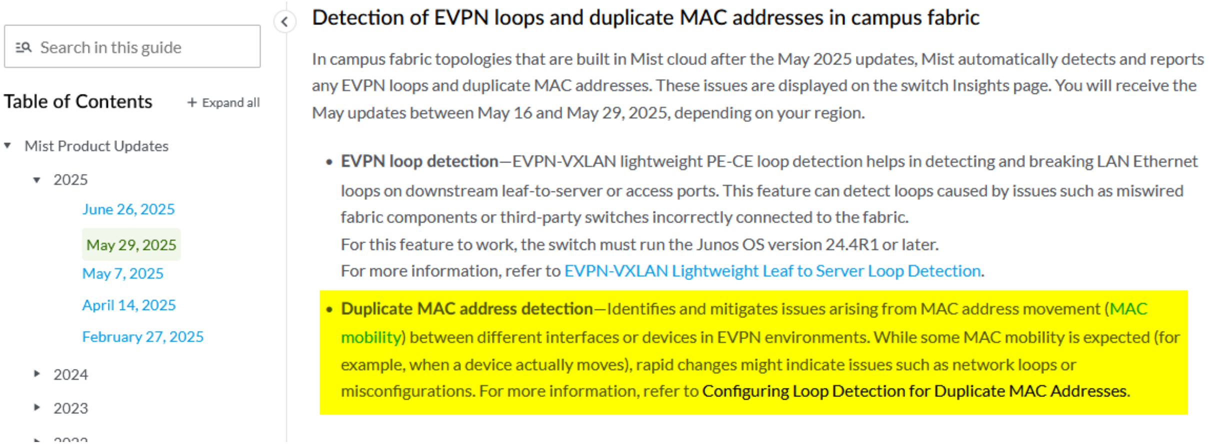
To prevent this issue and ensure proper roaming for wireless clients, two approaches can be considered:

- By design: When using a centralized (tunneled) approach with Juniper Mist Edge integrated into the EVPN fabric, roaming does not appear as a duplicate MAC event. This is because all access points terminate their overlay tunnels on the Juniper Mist Edge, so the EVPN fabric always sees the client MAC address on the same upstream Mist Edge interfaces, regardless of which AP the client is connected to.

- Relaxing the default parameters for duplicate MAC address detection: Juniper EVPN fabrics enable administrators to adjust the detection window and related parameters (please review). Starting May

29, 2025, all newly created EVPN fabrics automatically use relaxed settings based on recommended best practices. With these defaults in place, wireless client roaming is supported without requiring additional configuration.

Figure 81: Fabric Update May 29 2025



Detection of EVPN loops and duplicate MAC addresses in campus fabric

In campus fabric topologies that are built in Mist cloud after the May 2025 updates, Mist automatically detects and reports any EVPN loops and duplicate MAC addresses. These issues are displayed on the switch Insights page. You will receive the May updates between May 16 and May 29, 2025, depending on your region.

- **EVPN loop detection**—EVPN-VXLAN lightweight PE-CE loop detection helps in detecting and breaking LAN Ethernet loops on downstream leaf-to-server or access ports. This feature can detect loops caused by issues such as miswired fabric components or third-party switches incorrectly connected to the fabric.
  For this feature to work, the switch must run the Junos OS version 24.4R1 or later.
  For more information, refer to EVPN-VXLAN Lightweight Leaf to Server Loop Detection.
- **Duplicate MAC address detection**—Identifies and mitigates issues arising from MAC address movement (MAC mobility) between different interfaces or devices in EVPN environments. While some MAC mobility is expected (for example, when a device actually moves), rapid changes might indicate issues such as network loops or misconfigurations. For more information, refer to Configuring Loop Detection for Duplicate MAC Addresses.

Older fabrics need the additional CLI commands listed below depending on fabric underlay and device type. Please do not change the parameters configured here as they are recommended best practice.

```
# Duplicate MAC detection relaxation for fabric ipv4-undelay
set groups top protocols evpn duplicate-mac-detection auto-recovery-time 5 detection-threshold
10 detection-window 20
#
# Duplicate MAC detection relaxation for fabric ipv6-undelay or physical ex92xx or vJunos-switch
set groups top routing-instances evpn_vs protocols evpn duplicate-mac-detection auto-recovery-
time 5 detection-threshold 10 detection-window 20
```

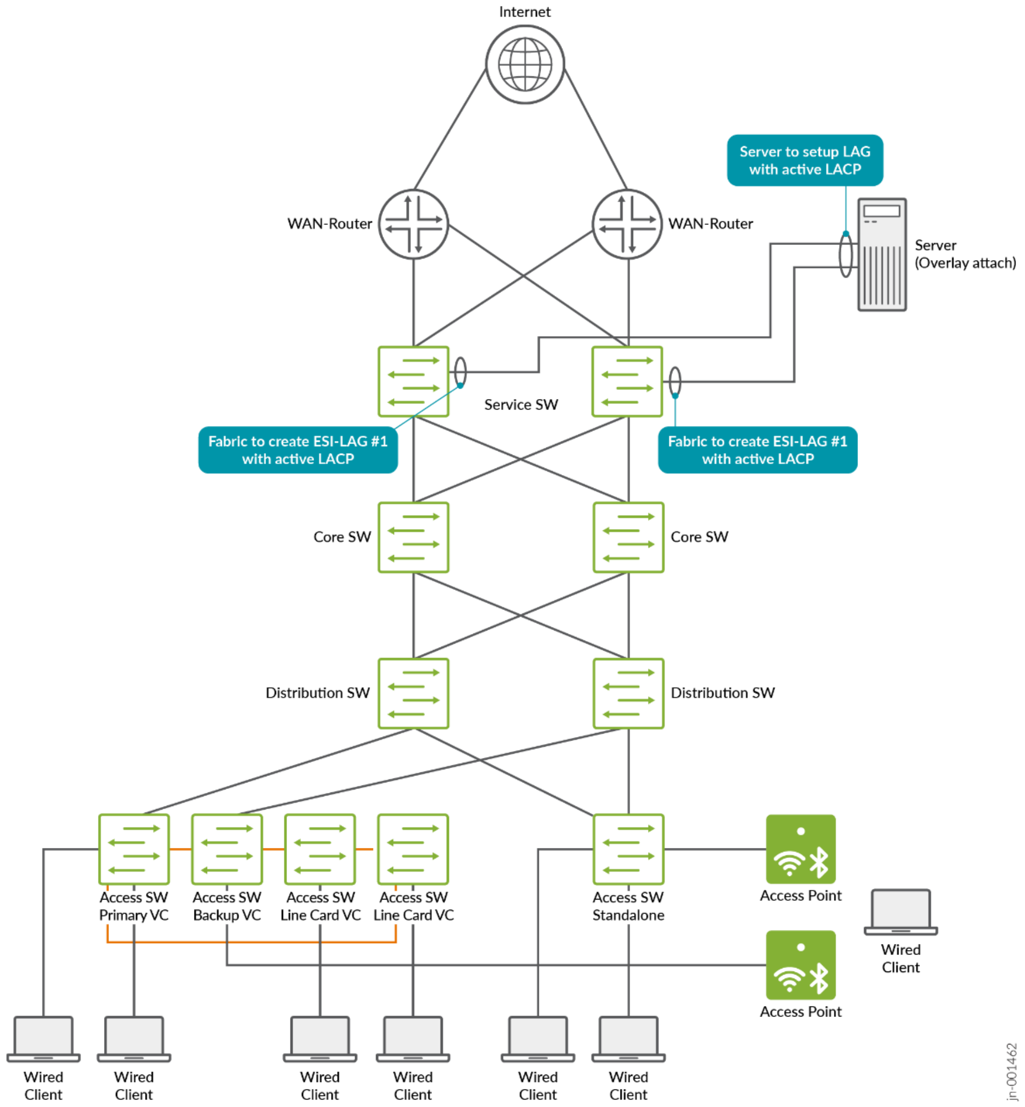
NOTE: Even when using the centralized approach with Juniper Mist Edge, it is recommended to verify that duplicate MAC address relaxation is enabled. This ensures proper handling in cases where APs terminate their tunnels on Juniper Mist Edges in different clusters and wireless clients attempt to roam between them.

Building a Hybrid Lab with Two Juniper Mist Edges and ESI-LAG Fabric Attach

The recommended method for integrating Juniper Mist Edge is to connect it at the service-block function of the fabric on the northbound side. It should not be connected at the access switch layer, since access switches typically lack the resources and capacity to manage the required traffic.

Integration should take advantage of the redundancy features provided by the EVPN fabric. This involves using ESI-LAG connections to the Juniper Mist Edge and enabling active LACP on both ends for link monitoring. The diagram below illustrates this approach using the example of a standard server integration.

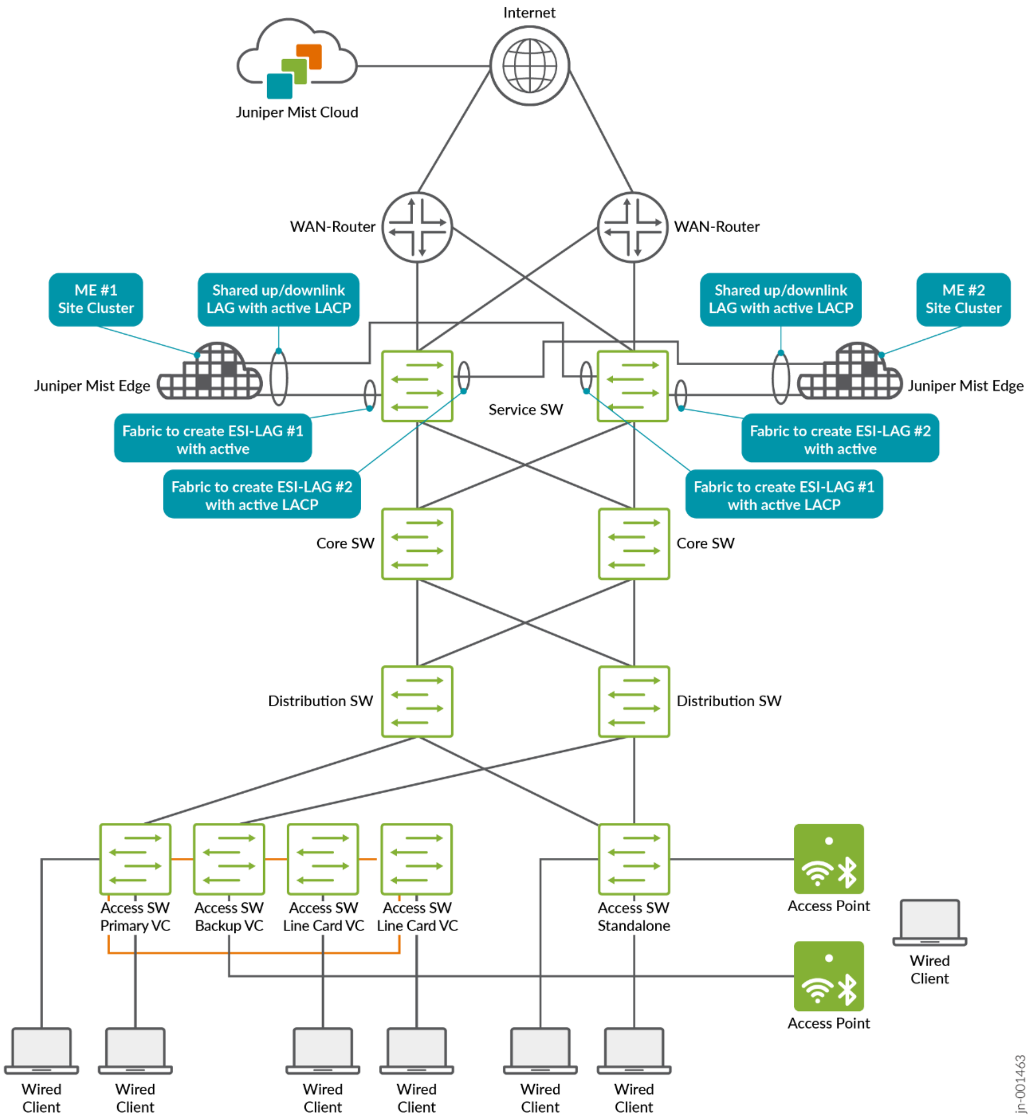
Figure 82: Campus Fabric Server Integration



NOTE: Juniper Mist Edge offers several other integration options, but these should be disregarded for campus fabric deployments. The proper method is to integrate using ESI-LAG on the fabric side and a single LAG on the Juniper Mist Edge, with both upstream and downstream interfaces multiplexed through that connection.In this lab design, we follow the recommended minimum of deploying two Juniper Mist Edges to provide redundancy in case one becomes unavailable, and we place both units in a single cluster. All APs in the EVPN fabric are assigned to a single AP site (the switch site assignments may differ, but that does not matter for this scenario). It is also required to enable the Shuffle-by-Site feature. With this enabled, one of the two Mist Edges is chosen to handle tunnel termination during normal operation, while the other remains on standby and automatically takes over if the primary unit becomes unreachable from the APs.Important note: When all Mist Edges reside in a single cluster, it is essential to prevent APs from distributing their tunnels across both Mist Edges. Although an active/active approach might appear beneficial, it can lead to unnecessary MAC address movement between two ESI-LAGs in the fabric when wireless clients roam between APs anchored on different Mist Edges. Active/active designs should only be used when mobility domains are separated, as explained in the next section on scale-out strategies.

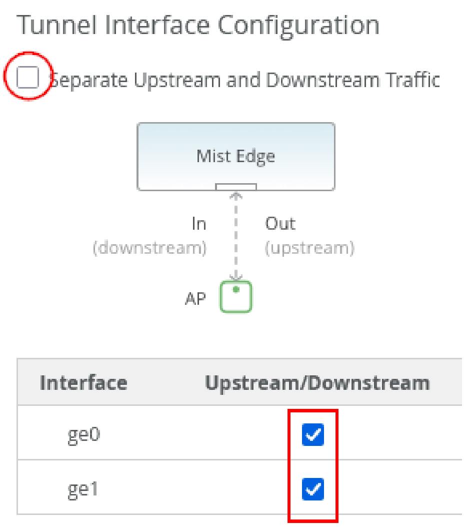
Juniper Mist Edge devices can provide either two or four uplink interfaces for tunnel operation, depending on the model. The setup shown below illustrates a lab design recommended for Juniper Mist Edges that support two uplink interfaces for tunnel operation:

Figure 83: Campus Fabric Mist Edge Integration When Two Revenue Interfaces



When setting up Juniper Mist Edge, the LAG and active LACP configurations are not directly specified in the Juniper Mist portal. To configure this, disable the separate upstream and downstream operation and instead enable both upstream and downstream interfaces together, as shown in the image below:

Figure 84: LAG/LACP Configuration on Mist-Edge



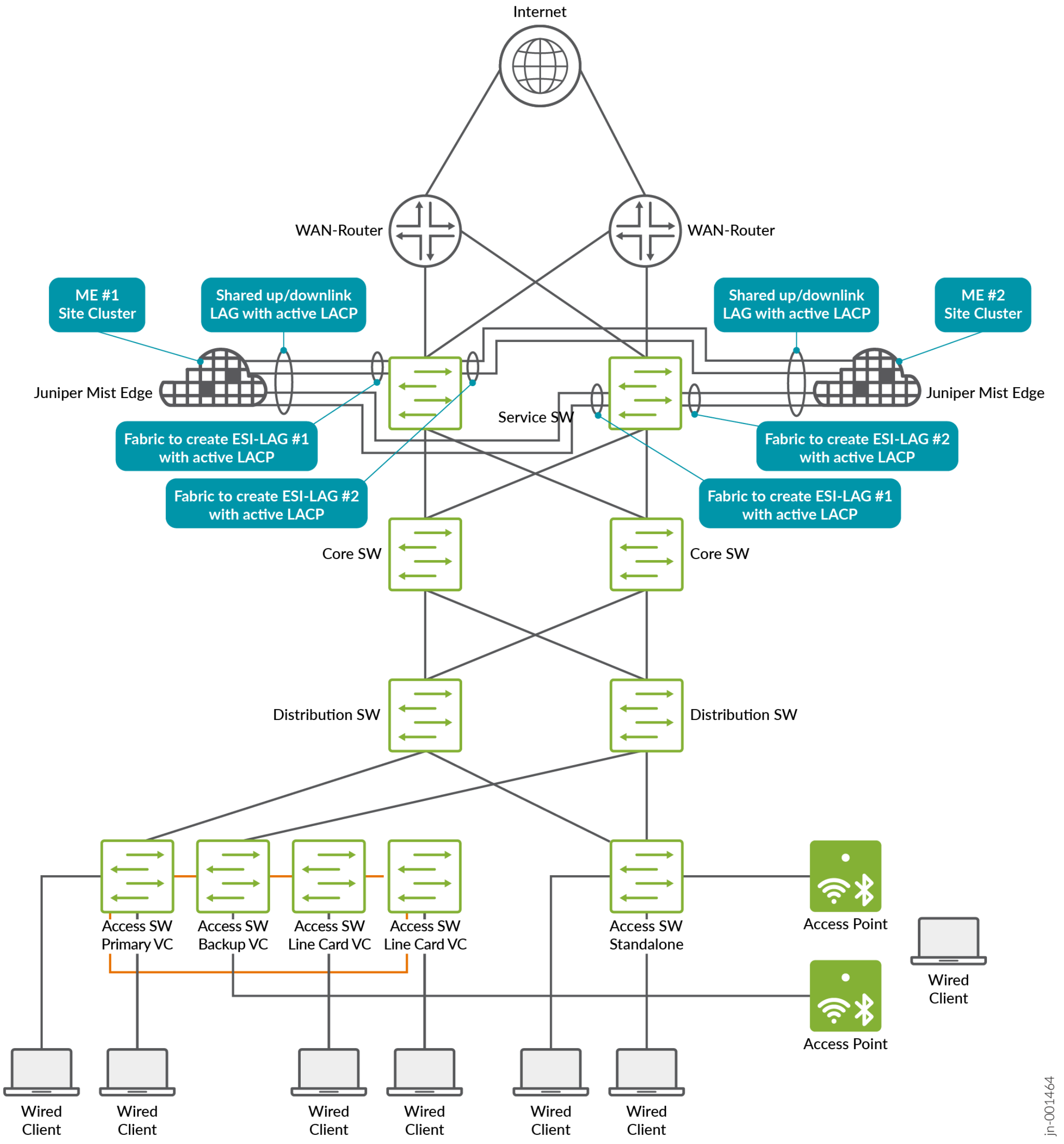
Tunnel Interface Configuration

☐ Separate Upstream and Downstream Traffic

Interface	Upstream/Downstream
ge0	☑
ge1	☑

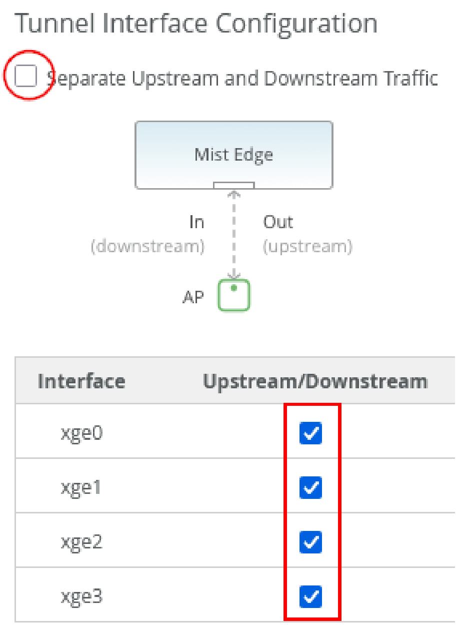
Here's a suggested lab setup for Juniper Mist Edges, which have four revenue interfaces for tunnel operation:

Figure 85: Campus Fabric Mist Edge Integration When Four Revenue Interfaces



When setting up Juniper Mist Edge, the LAG and active LACP configurations are not directly specified in the Juniper Mist portal. To configure this, disable the separate upstream and downstream operation and instead enable both upstream and downstream interfaces together, as shown in the image below:

Figure 86: LAG/LACP Configuration for Mist-Edge



Tunnel Interface Configuration

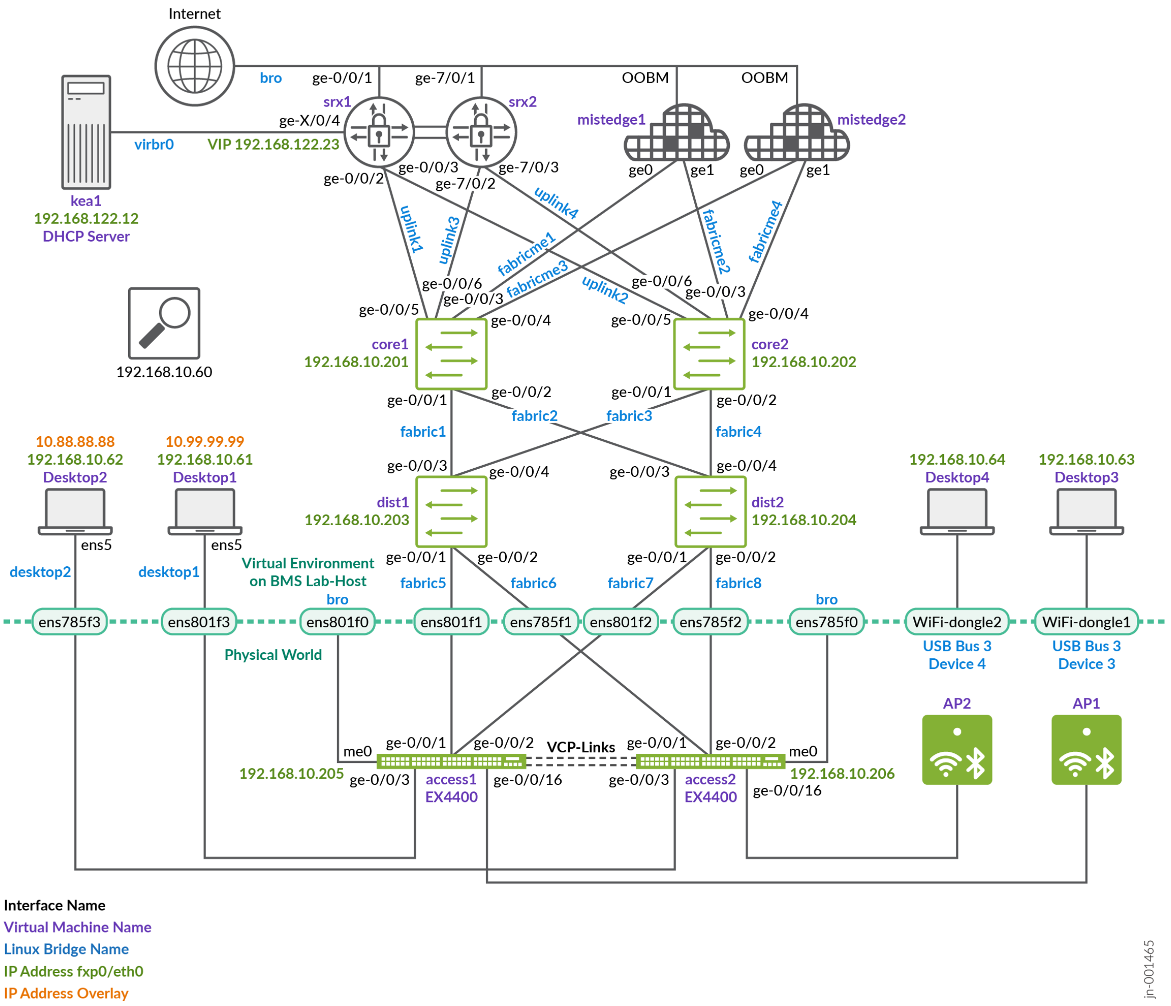
☐ Separate Upstream and Downstream Traffic

Interface	Upstream/Downstream
xge0	☑
xge1	☑
xge2	☑
xge3	☑

Lab Preparation

This lab was constructed to show the functionality with the fewest physical devices required. It is not intended for use in a production environment. The integration of vJunos-switch VMs with Mist Cloud is described here.

Figure 87: Hybrid Lab with Mist Edge Integration



The lab described here was not built with physical Juniper Mist Edge devices. Instead, we used a combination of virtual machines and physical hardware, particularly because APs cannot be virtualized. The two additional access switches in this setup are also physical devices, which allows us to utilize Virtual Chassis and VXLAN group-based policy (GBP) functionality.

NOTE: The Juniper Mist Edge virtual machine does not support LACP on LAGs when deployed as a VM. When LACP is configured on the LAG of the VM, the LACP configuration is automatically suppressed in the backend and not applied to the VM. This limitation exists because LAG configurations are typically handled at the hypervisor level on the host operating system. For our lab environment, we configure it as a physical ME-X1 device to overcome this restriction.

The topology above the service-block switches is part of the core network, with two Juniper Mist Edge devices connected to each core switch. Each Juniper Mist Edge has its own dedicated ESI-LAG configuration. The design concept for this setup is as follows:

- VLAN 1033 with subnet 10.33.33.0/24 is assigned as the native/access VLAN for the APs, allowing them to boot properly. Only a single native VLAN is required for the AP, since all SSID VLANs are tunneled through the Juniper Mist Edge. This VLAN serves multiple purposes:
    - It provides the AP with a DHCP lease for booting.
    - It allows the AP to communicate with the Juniper Mist cloud for management.
    - It enables the AP to establish a tunnel to one of the two Juniper Mist Edge devices, carrying all VLANs associated with SSIDs via default gateway in the next VLAN.
- VLAN 1011 with subnet 10.11.11.0/24. This is the network in which the Juniper Mist Edges have static IP addresses for tunnel termination from the AP. VLANs 1099 and 1088 are assigned to a specific SSID that wireless clients can connect to. The traffic for these VLANs is tunneled from the AP to the Juniper Mist Edge using a proprietary L2TPv3 tunnel. The tunneled traffic is then presented as an L2 VLAN trunk and reinserted into the fabric through a service block, allowing it to be mapped into the appropriate VRFs.
- VLANs 1091 and 1081 will be used for wired clients.

If not already completed, perform the steps for IP Clos fabric creation, DHCP relay configuration, and WAN router integration as described in the appendix above.

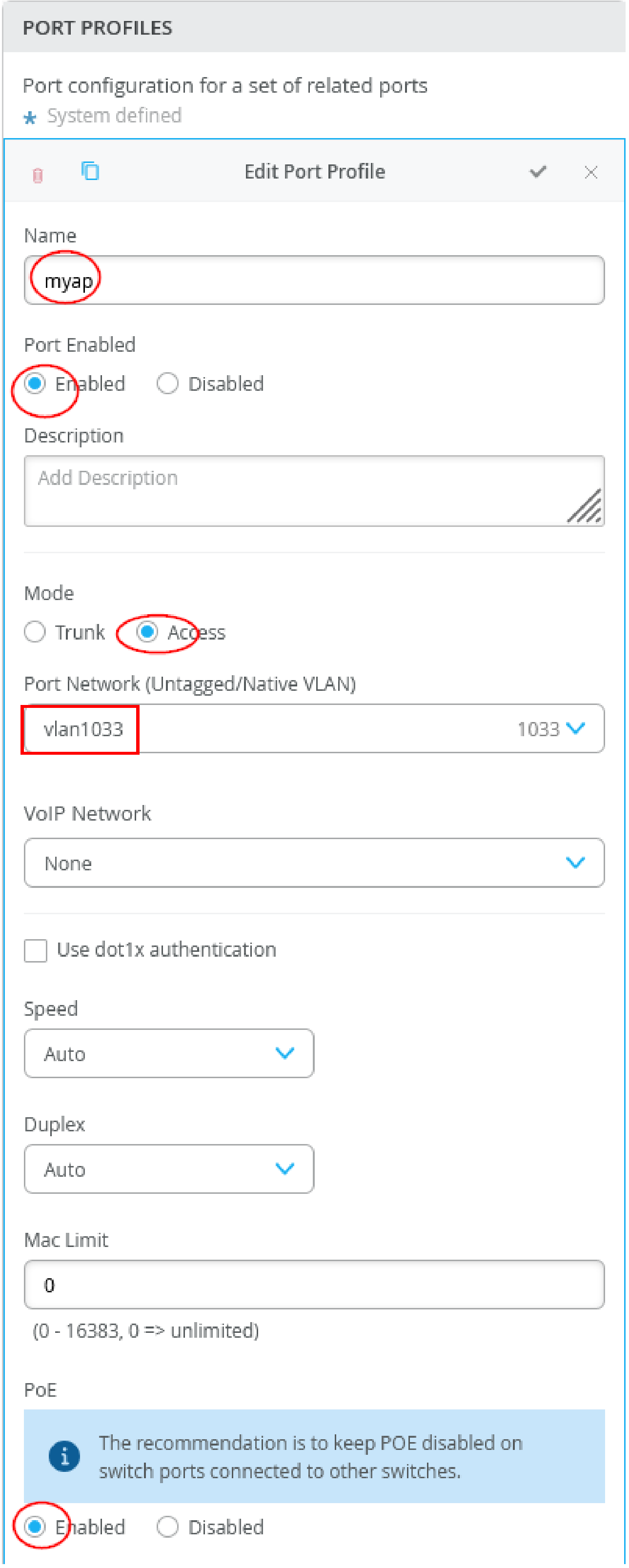
Juniper Mist Edge and Fabric Configuration

In the switch template, a port profile is configured to connect the AP to the access switch. Since forwarding is handled through the Juniper Mist Edge, only VLAN 1033 needs to be set as the access VLAN.

Navigate to **Switch Template** and configure a new port profile:

- Name=`myap`
- Port Enabled=`Enabled`
- Mode=`Access`
- Port Network=`vlan1033`
- PoE=`Enabled`

Figure 88: Port Profile for Access Point



PORT PROFILES

Port configuration for a set of related ports

★ System defined

Edit Port Profile

Name

myap

Port Enabled

● Enabled ○ Disabled

Description

Add Description

Mode

○ Trunk ● Access

Port Network (Untagged/Native VLAN)

vlan1033 1033

VoIP Network

None

□ Use dot1x authentication

Speed

Auto

Duplex

Auto

Mac Limit

0

(0 - 16383, 0 => unlimited)

PoE

The recommendation is to keep POE disabled on switch ports connected to other switches.

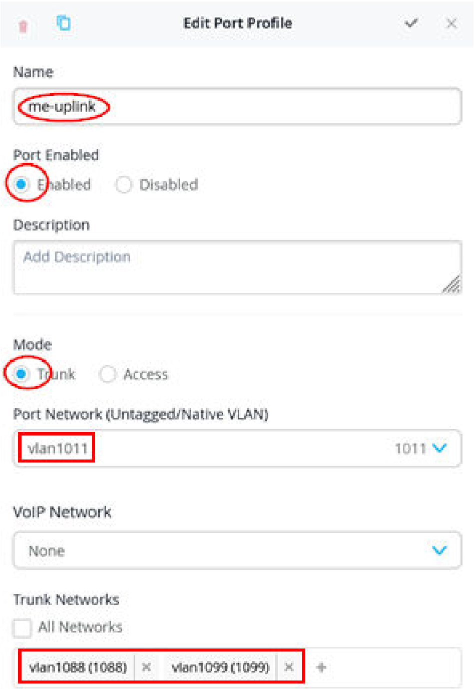
● Enabled ○ Disabled

Next, a port profile is required for the links on the service-block switches (core1 and core2 in this example). The AP tunnel is carried as the native VLAN, while the VLANs for wireless client breakout are configured as tagged VLANs.

Navigate to **Switch Template** and configure a new port profile:

- Name=`me-uplink`
- Port Enabled=`Enabled`
- Mode=`Trunk`
- Port Network=`vlan1011`
- Trunk Networks=`vlan1088` and `vlan1099`

Figure 89: Port Profile for Mist-Edge Up/Downstream



Edit Port Profile

Name

me-uplink

Port Enabled

● Enabled ○ Disabled

Description

Add Description

Mode

● Trunk ○ Access

Port Network (Untagged/Native VLAN)

vlan1011 1011

VoIP Network

None

Trunk Networks

□ All Networks

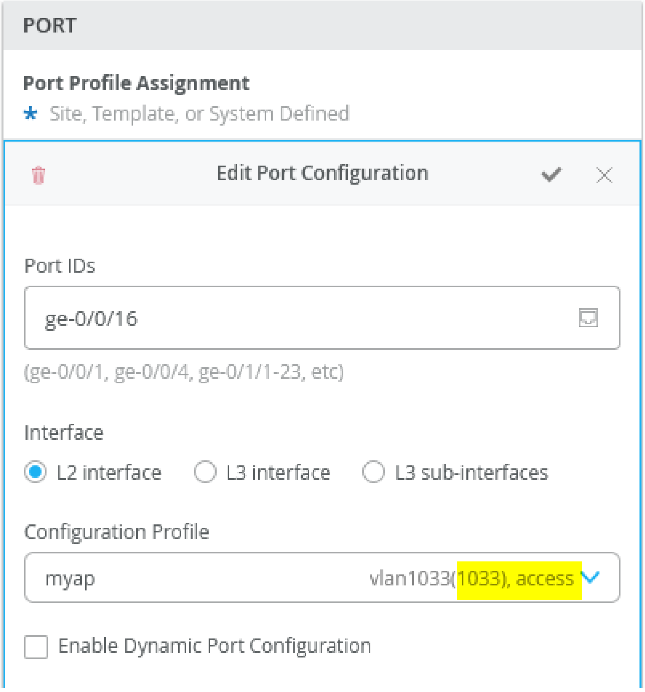
vlan1088 (1088) × vlan1099 (1099) × +

The configuration of **Access-Switch1** and **Access-Switch2** for wired and access ports have the following port profile created:

Then, on Access-Switch1 and Access-Switch2, apply this port profile on all ports connected to an AP:

- Port IDs=`ge-0/0/16`
- Configuration Profile=`myap`

Figure 90: Port Config for Access Point



PORT

Port Profile Assignment

* Site, Template, or System Defined

Edit Port Configuration

Port IDs

ge-0/0/16

(ge-0/0/1, ge-0/0/4, ge-0/1/1-23, etc)

Interface

● L2 interface ○ L3 interface ○ L3 sub-interfaces

Configuration Profile

myap vlan1033(1033), access

□ Enable Dynamic Port Configuration

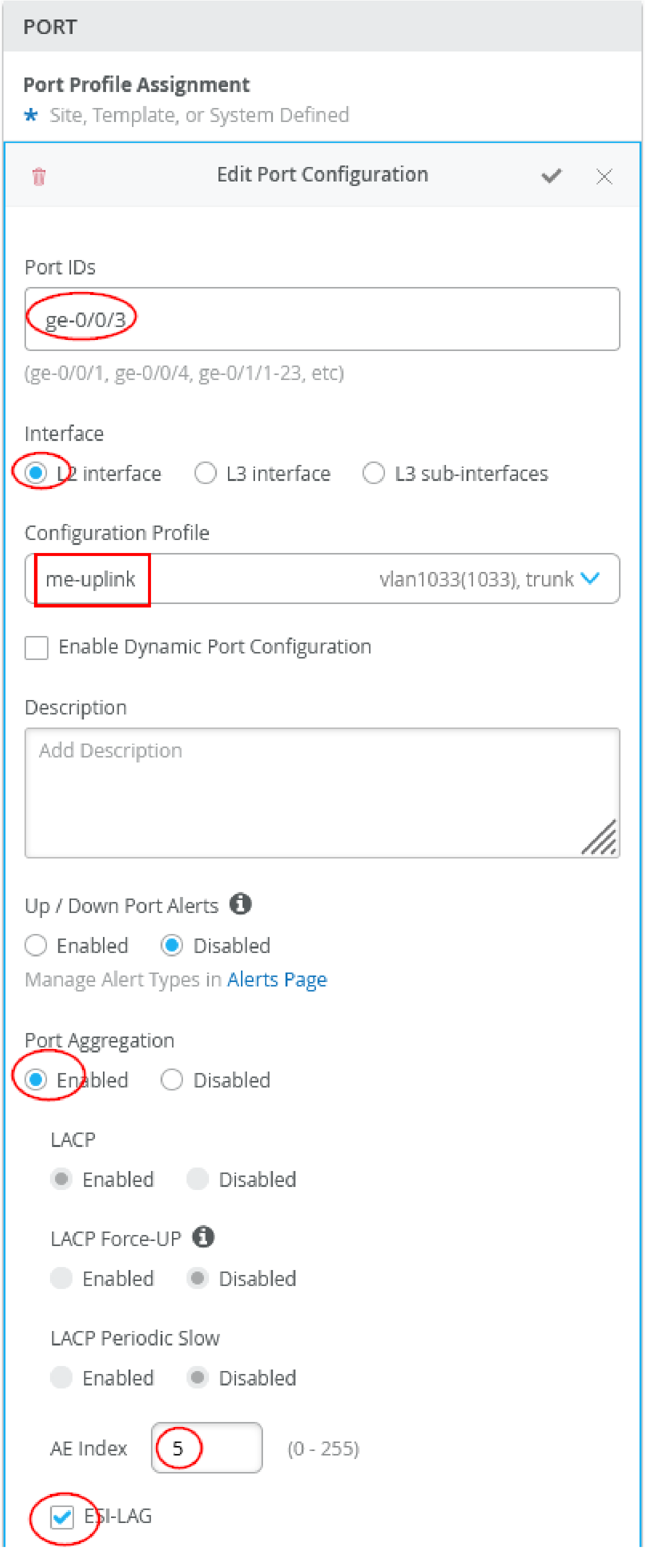
Check that your APs are coming up now and are seen in Juniper Mist cloud as they should now get a DHCP lease and are able to contact Juniper Mist cloud.

Next, configure **core1** and **core2** which perform the service-block functions.

The configuration of Core1-Switch and Core2-Switch for the first Juniper Mist Edge is:

- Port ID=`ge-0/0/3`
- Interface=`L2 interface`
- Configuration Profile=`me-uplink`
- Port Aggregation=`Enabled`
  - AE Index=`5` (you need to ensure this index value is unique and only used on the interface towards the same Juniper Mist Edge)
  - ESI-LAG=`Enabled` (enabling this is mandatory)

Figure 91: Both Service-Blocks First Mist-Edge ESI-LAG



PORT

Port Profile Assignment

★ Site, Template, or System Defined

Edit Port Configuration

Port IDs

ge-0/0/3

(ge-0/0/1, ge-0/0/4, ge-0/1/1-23, etc)

Interface

● L2 interface ○ L3 interface ○ L3 sub-interfaces

Configuration Profile

me-uplink vlan1033(1033), trunk

□ Enable Dynamic Port Configuration

Description

Add Description

Up / Down Port Alerts

○ Enabled ● Disabled

Manage Alert Types in Alerts Page

Port Aggregation

● Enabled ○ Disabled

LACP

● Enabled ○ Disabled

LACP Force-UP

○ Enabled ● Disabled

LACP Periodic Slow

○ Enabled ● Disabled

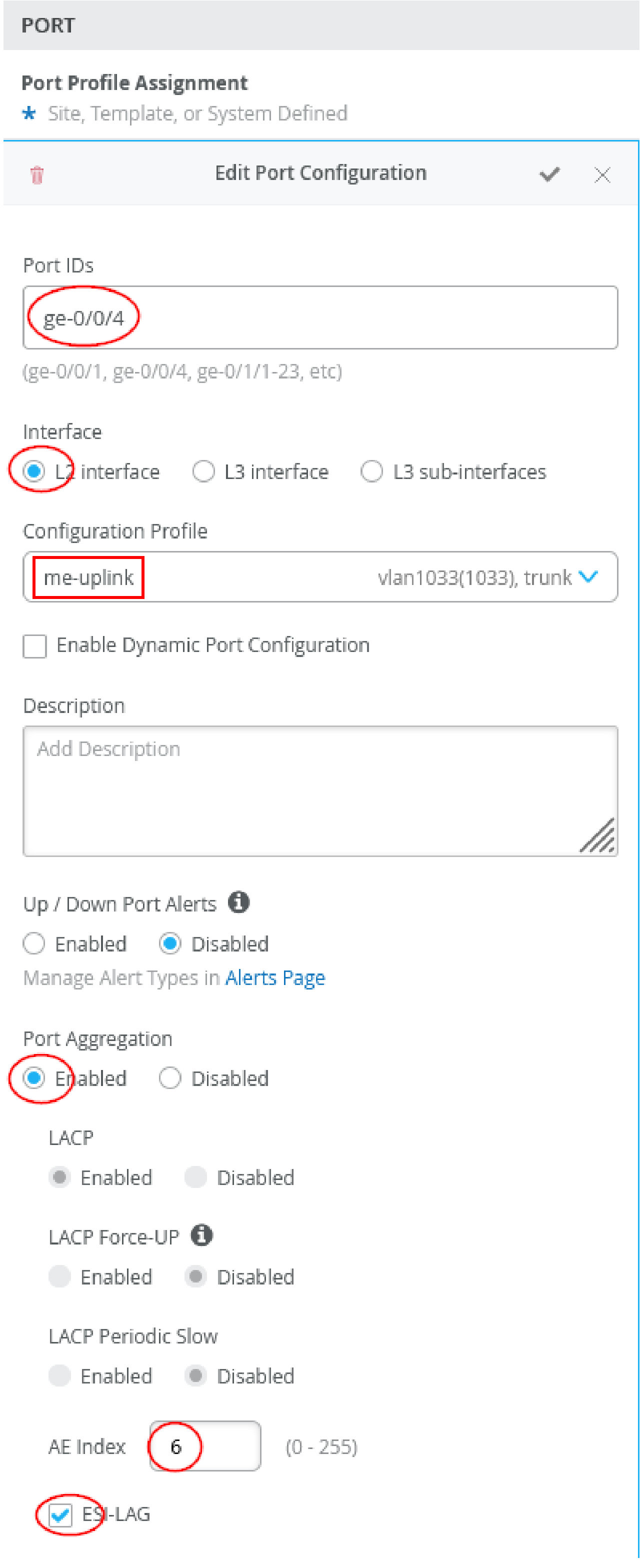
AE Index 5 (0 - 255)

☑ ESI-LAG

The configuration of Core1-Switch and Core2-Switch for the second Juniper Mist Edge is:

- Port ID=`ge-0/0/4`
- Interface=`L2 interface`
- Configuration Profile=`me-uplink`
- Port Aggregation=`Enabled`
  - AE Index=`6` (you need to ensure this index value is only used on the interface towards the same Juniper Mist Edge)
  - ESI-LAG=`Enabled` (Enabling this is mandatory)

Figure 92: Both Service-Blocks Second Mist-Edge ESI-LAG



PORT

Port Profile Assignment

* Site, Template, or System Defined

Edit Port Configuration

Port IDs

ge-0/0/4

(ge-0/0/1, ge-0/0/4, ge-0/1/1-23, etc)

Interface

● L2 interface ○ L3 interface ○ L3 sub-interfaces

Configuration Profile

me-uplink vlan1033(1033), trunk

□ Enable Dynamic Port Configuration

Description

Add Description

Up / Down Port Alerts

○ Enabled ● Disabled

Manage Alert Types in Alerts Page

Port Aggregation

● Enabled ○ Disabled

LACP

● Enabled ○ Disabled

LACP Force-UP

○ Enabled ● Disabled

LACP Periodic Slow

○ Enabled ● Disabled

AE Index 6 (0 - 255)

☑ ESI-LAG

As of July 2025, the Juniper Mist cloud does not automatically configure default gateways or DHCP relay for a VLAN and VRF at the service-block function. This feature may be included in a future release, but for now, these configurations must be added manually using additional CLI commands. To determine what needs to be configured, it is recommended that for IP Clos fabrics you first set up a VLAN for wireless clients on a port of one of your access switches. You can then copy the IRB and DHCP relay settings to the service-block switches. The following example illustrates this process for our fabric:

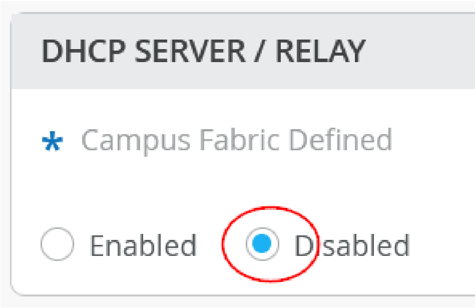
> **NOTE**: Consider a maintenance window of your EVPN Fabric when testing the below additional Junos CLI.

For VLAN 1011, which is used to establish the tunnel from the AP to the Juniper Mist Edge, configuring the default gateway is required. DHCP relay is not necessary, since the connected Juniper Mist Edges are always assigned static IP addresses.

```
set interfaces irb unit 1011 family inet address 10.11.11.1/24
set interfaces irb unit 1011 family inet mtu 9000
set interfaces irb unit 1011 description vlan1011
set interfaces irb unit 1011 no-dhcp-flood
set interfaces irb unit 1011 mac 00:00:5e:e4:31:57
#
set groups top routing-instances evpn_vs vlans vlan1011 l3-interface irb.1011
set groups top routing-instances devices interface irb.1011
```

Depending on your service block function configuration and what is attached, you might be able to automatically include large portions of the required DHCP relay configuration onto the VRF at the service-block function. Usually, when you check the configuration of your service-block switch, the DHCP relay will be disabled as shown in the image below.

Figure 93: DHCP Relay Local Configuration on Service Block Function



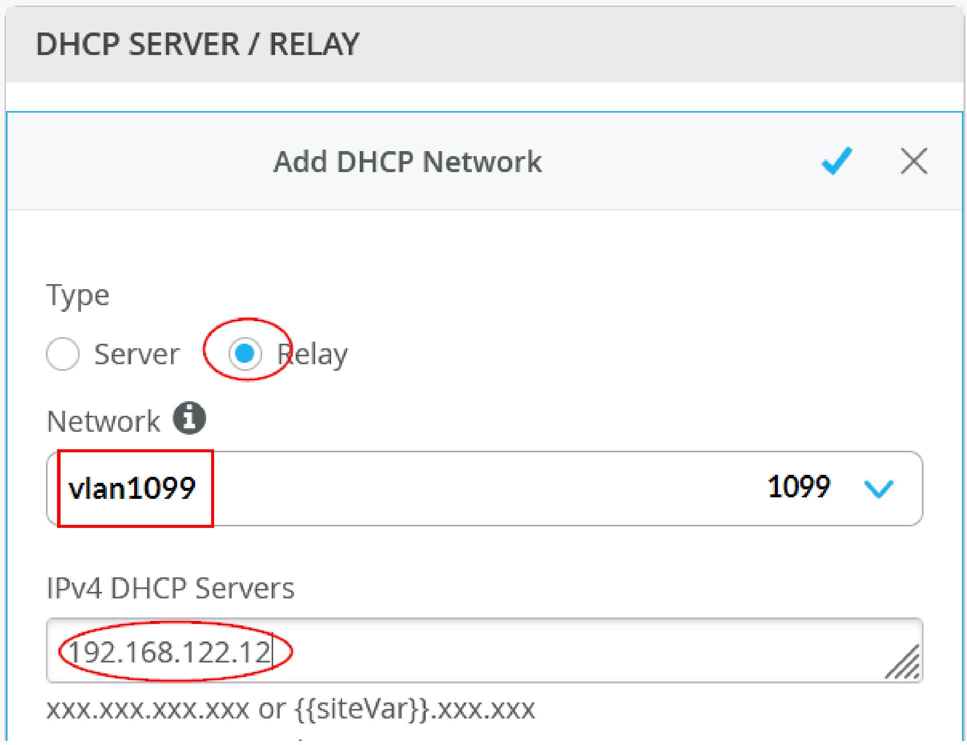
DHCP SERVER / RELAY

* Campus Fabric Defined

○ Enabled ● Disabled

This is due to the assumption that clients typically obtain DHCP leases only through an access switch. Servers connected through an ESI-LAG at a service-block function usually rely on static IP addresses. However, as an exception, you can override this behavior by enabling the option and then manually configuring DHCP relay for your wireless client breakout networks.

Figure 94: DHCP Relay Local Configuration



DHCP SERVER / RELAY

Add DHCP Network

Type

○ Server ● Relay

Network

vlan1099 1099

IPv4 DHCP Servers

192.168.122.12

xxx.xxx.xxx.xxx or {{siteVar}}.xxx.xxx

NOTE: An issue that can occur when enabling DHCP relay on a VRF is that clients connected to the EVPN fabric access switches using that same VRF may suddenly be unable to obtain a DHCP lease!

This behavior can occur depending on the Junos release in use, because enabling the DHCP relay agent installs hidden traffic filters within the VRF. These filters may inadvertently block DHCP relay packets coming from access switches, even when they are simply passing through the VRF along with locally generated traffic. To avoid or resolve this, you can use one of the following approaches:

- If the service-block function switch is a physical device, add an additional Junos configuration such as

  ```
  set groups top
  routing-instances <vrf> forwarding-options dhcp-relay
  no-snoop
  ```

  to the affected VRF on the service-block switches. This removes the hidden filters that could block DHCP relay traffic originating from access switches.

- If the service-block function is running on a virtual switch VM and the above adjustment does not work, it is recommended to separate the VLAN used by clients on access switches into one VRF, and place all wireless clients into another VRF. By doing this, the DHCP relay configuration on the service block will not interfere with other VRFs.

This is the configuration needed for wireless client breakout for VLAN 1099:

```
set interfaces irb unit 1099 family inet address 10.99.99.1/24
set interfaces irb unit 1099 family inet mtu 9000
set interfaces irb unit 1099 description vlan1099
set interfaces irb unit 1099 no-dhcp-flood
set interfaces irb unit 1099 mac 00:00:5e:e4:31:57
#
set groups top routing-instances evpn_vs vlans vlan1099 l3-interface irb.1099
set groups top routing-instances customera interface irb.1099
#
set groups top routing-instances customera forwarding-options dhcp-relay server-group vlan1099
192.168.122.12
set groups top routing-instances customera forwarding-options dhcp-relay group vlan1099
interface irb.1099
set groups top routing-instances customera forwarding-options dhcp-relay group vlan1099 active-
server-group vlan1099
set groups top routing-instances customera forwarding-options dhcp-relay group vlan1099 relay-
option-82 circuit-id vlan-id-only
set groups top routing-instances customera forwarding-options dhcp-relay group vlan1099 relay-
option-82 server-id-override
set groups top routing-instances customera forwarding-options dhcp-relay group vlan1099 route-
suppression destination
set groups top routing-instances customera forwarding-options dhcp-relay group vlan1099
overrides relay-source lo0.1
set groups top routing-instances customera forwarding-options dhcp-relay forward-only
```

```
set groups top routing-instances customera forwarding-options dhcp-relay no-snoop
```

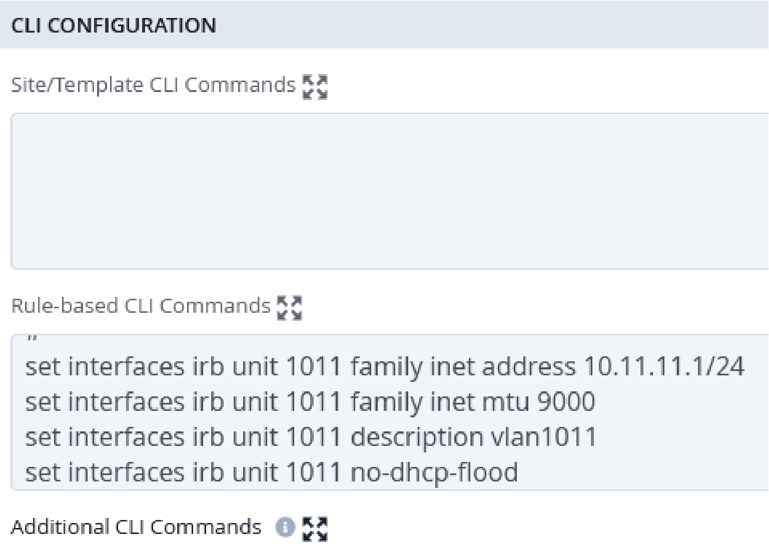
This is the configuration needed for wireless client breakout for VLAN 1088:

```
set interfaces irb unit 1088 family inet address 10.88.88.1/24
set interfaces irb unit 1088 family inet mtu 9000
set interfaces irb unit 1088 description vlan1088
set interfaces irb unit 1088 no-dhcp-flood
set interfaces irb unit 1088 mac 00:00:5e:e4:31:57
#
set groups top routing-instances evpn_vs vlans vlan1088 l3-interface irb.1088
set groups top routing-instances customerb interface irb.1088
#
set groups top routing-instances customerb forwarding-options dhcp-relay server-group vlan1088 192.168.122.12
set groups top routing-instances customerb forwarding-options dhcp-relay group vlan1088 interface irb.1088
set groups top routing-instances customerb forwarding-options dhcp-relay group vlan1088 active-server-group vlan1088
set groups top routing-instances customerb forwarding-options dhcp-relay group vlan1088 relay-option-82 circuit-id vlan-id-only
set groups top routing-instances customerb forwarding-options dhcp-relay group vlan1088 relay-option-82 server-id-override
set groups top routing-instances customerb forwarding-options dhcp-relay group vlan1088 route-suppression destination
set groups top routing-instances customerb forwarding-options dhcp-relay group vlan1088 overrides relay-source lo0.2
set groups top routing-instances customerb forwarding-options dhcp-relay forward-only
set groups top routing-instances customerb forwarding-options dhcp-relay no-snoop
```

NOTE: The line containing "l3-interface" includes the statement "routing-instances evpn_vs" only because our core switches are vJunos-switch VMs. Be sure to verify the correct configuration for your own fabric and deployment.

In our setup, this is added by defining a role in the switch template for the additional CLI commands and assigning that role to core1 and core2. The statements then appear as rule-based CLI commands. Alternatively, additional CLI commands can be added individually on each switch.

Figure 95: Additional Junos CLI via Role



CLI CONFIGURATION

Site/Template CLI Commands

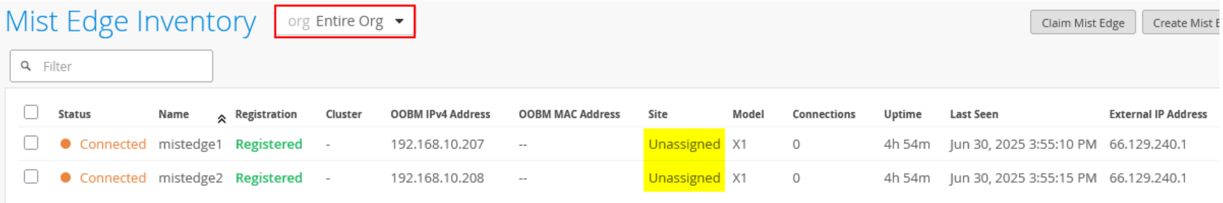
Rule-based CLI Commands

```
set interfaces irb unit 1011 family inet address 10.11.11.1/24
set interfaces irb unit 1011 family inet mtu 9000
set interfaces irb unit 1011 description vlan1011
set interfaces irb unit 1011 no-dhcp-flood
```

Additional CLI Commands

Next, we will configure the Juniper Mist Edge. The image below shows the state after the lab has been set up, with the two Juniper Mist Edge VMs launched and registered in the Juniper Mist cloud using the provided registration code.

Figure 96: New Mist-Edge Globally Assgined



Mist Edge Inventory

☐	Status	Name	Registration	Cluster	OOBM IPv4 Address	OOBM MAC Address	Site	Model	Connections	Uptime	Last Seen	External IP Address
☐	● Connected	mistedge1	Registered	-	192.168.10.207	--	Unassigned	X1	0	4h 54m	Jun 30, 2025 3:55:10 PM	66.129.240.1
☐	● Connected	mistedge2	Registered	-	192.168.10.208	--	Unassigned	X1	0	4h 54m	Jun 30, 2025 3:55:15 PM	66.129.240.1

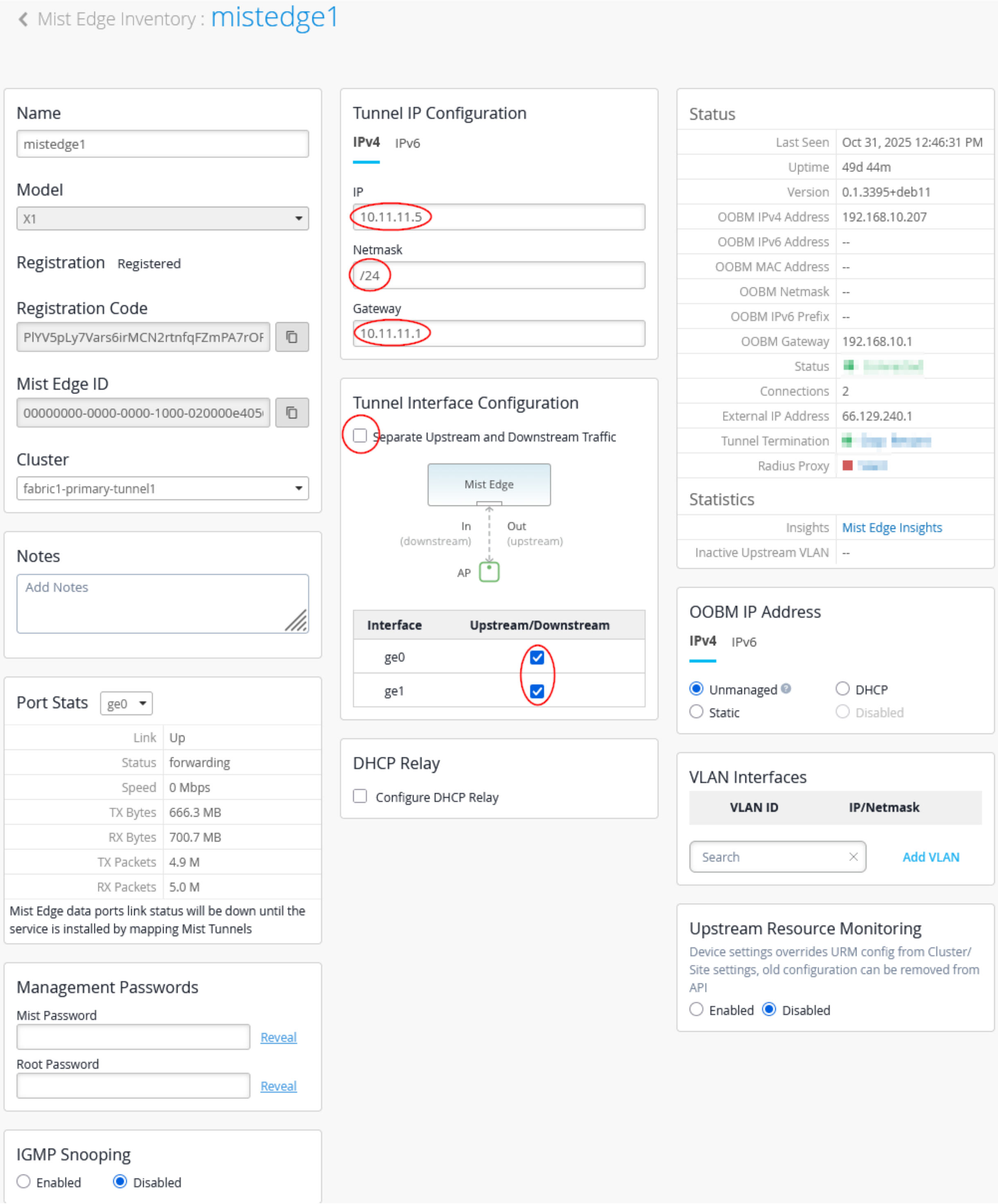
NOTE: In this example, the Juniper Mist Edges are used at the global level and are not assigned to a specific site, so they appear as “unassigned.” Larger organizations may choose to assign Mist Edges to a site and configure tunnels through the **Organization -> Site Configuration** option. This approach was not implemented in our example because we wanted to demonstrate how the Shuffle-by-Site feature operates.

Next, we configure the first Juniper Mist Edge as follows:

- Name=`mistedge1`
- IP=`10.11.11.5` (remember our AP’s vlan should be different from Mist-Edge VLAN used)

- Netmask=`/24`
- Gateway=`10.11.11.1`
- Separate Upstream and Downstream Traffic=`Unchecked`
- Interface ge0=`Checked`
- Interface ge1=`Checked`
- DHCP Relay=`Unchecked`

Figure 97: First Mist-Edge Configuration



‹ Mist Edge Inventory : mistedge1

Name

mistedge1

Model

X1

Registration Registered

Registration Code

PlYV5pLy7Vars6irMCN2rtnfqFZmPA7rOF

Mist Edge ID

00000000-0000-0000-1000-020000e405

Cluster

fabric1-primary-tunnel1

Notes

Add Notes

Port Stats ge0

Link	Up
Status	forwarding
Speed	0 Mbps
TX Bytes	666.3 MB
RX Bytes	700.7 MB
TX Packets	4.9 M
RX Packets	5.0 M

Mist Edge data ports link status will be down until the service is installed by mapping Mist Tunnels

Management Passwords

Mist Password Reveal

Root Password Reveal

IGMP Snooping

○ Enabled ● Disabled

Tunnel IP Configuration

IPv4 IPv6

IP

10.11.11.5

Netmask

/24

Gateway

10.11.11.1

Tunnel Interface Configuration

□ Separate Upstream and Downstream Traffic

Interface	Upstream/Downstream
ge0	☑
ge1	☑

DHCP Relay

□ Configure DHCP Relay

Status

Last Seen	Oct 31, 2025 12:46:31 PM
Uptime	49d 44m
Version	0.1.3395+deb11
OOBM IPv4 Address	192.168.10.207
OOBM IPv6 Address	--
OOBM MAC Address	--
OOBM Netmask	--
OOBM IPv6 Prefix	--
OOBM Gateway	192.168.10.1
Status	🟩 [redacted]
Connections	2
External IP Address	66.129.240.1
Tunnel Termination	🟩 [redacted]
Radius Proxy	🟥 [redacted]

Statistics

Insights	Mist Edge Insights
Inactive Upstream VLAN	--

OOBM IP Address

IPv4 IPv6

● Unmanaged ○ DHCP
○ Static ○ Disabled

VLAN Interfaces

VLAN ID	IP/Netmask

Search Add VLAN

Upstream Resource Monitoring

Device settings overrides URM config from Cluster/ Site settings, old configuration can be removed from API

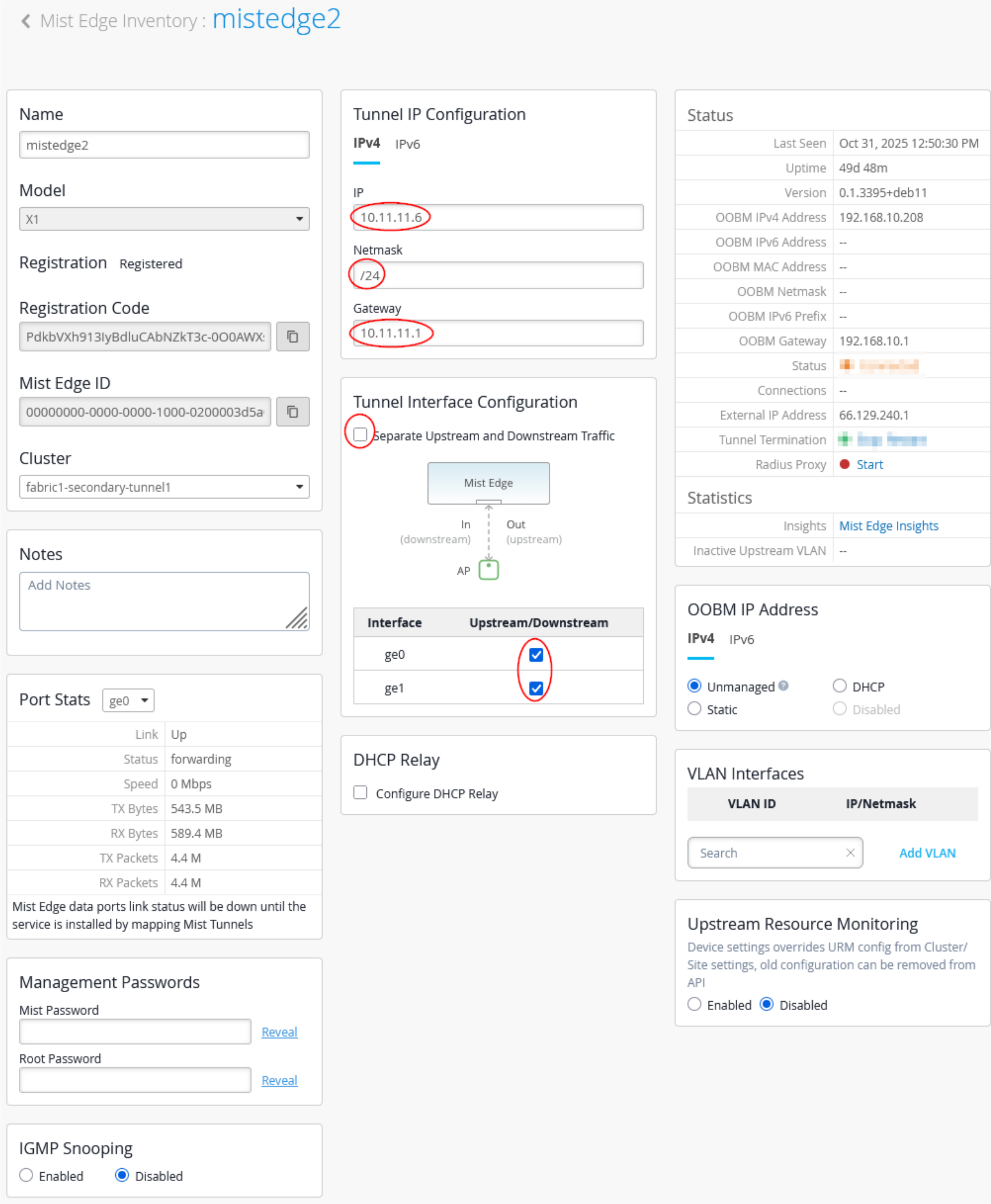
○ Enabled ● Disabled

NOTE: Do not configure DHCP relay directly on the Juniper Mist Edge. The fabric is set up to handle this traffic through the VRF at the service-block function.

Next, we configure the second Juniper Mist Edge as follows:

- Name=`mistedge2`
- IP=`10.11.11.6` (remember our AP's vlan should be different from Mist-Edge VLAN used)
- Netmask=`/24`
- Gateway=`10.33.33.1`
- Separate Upstream and Downstream Traffic=`Unchecked`
- Interface ge0=`Checked`
- Interface ge1=`Checked`
- DHCP Relay=`Unchecked`

Figure 98: Second Mist-Edge Configuration



‹ Mist Edge Inventory : mistedge2

Name

mistedge2

Model

X1

Registration Registered

Registration Code

PdkbVXh913lyBdluCAbNZkT3c-0O0AWX

Mist Edge ID

00000000-0000-0000-1000-0200003d5a

Cluster

fabric1-secondary-tunnel1

Notes

Add Notes

Port Stats ge0

Link	Up
Status	forwarding
Speed	0 Mbps
TX Bytes	543.5 MB
RX Bytes	589.4 MB
TX Packets	4.4 M
RX Packets	4.4 M

Mist Edge data ports link status will be down until the service is installed by mapping Mist Tunnels

Management Passwords

Mist Password Reveal

Root Password Reveal

IGMP Snooping

○ Enabled ● Disabled

Tunnel IP Configuration

IPv4 IPv6

IP

10.11.11.6

Netmask

/24

Gateway

10.11.11.1

Tunnel Interface Configuration

□ Separate Upstream and Downstream Traffic

Interface	Upstream/Downstream
ge0	☑
ge1	☑

DHCP Relay

□ Configure DHCP Relay

Status

Last Seen	Oct 31, 2025 12:50:30 PM
Uptime	49d 48m
Version	0.1.3395+deb11
OOBM IPv4 Address	192.168.10.208
OOBM IPv6 Address	--
OOBM MAC Address	--
OOBM Netmask	--
OOBM IPv6 Prefix	--
OOBM Gateway	192.168.10.1
Status	[redacted]
Connections	--
External IP Address	66.129.240.1
Tunnel Termination	[redacted]
Radius Proxy	● Start

Statistics

Insights	Mist Edge Insights
Inactive Upstream VLAN	--

OOBM IP Address

IPv4 IPv6

● Unmanaged ○ DHCP

○ Static ○ Disabled

VLAN Interfaces

VLAN ID	IP/Netmask

Search Add VLAN

Upstream Resource Monitoring

Device settings overrides URM config from Cluster/ Site settings, old configuration can be removed from API

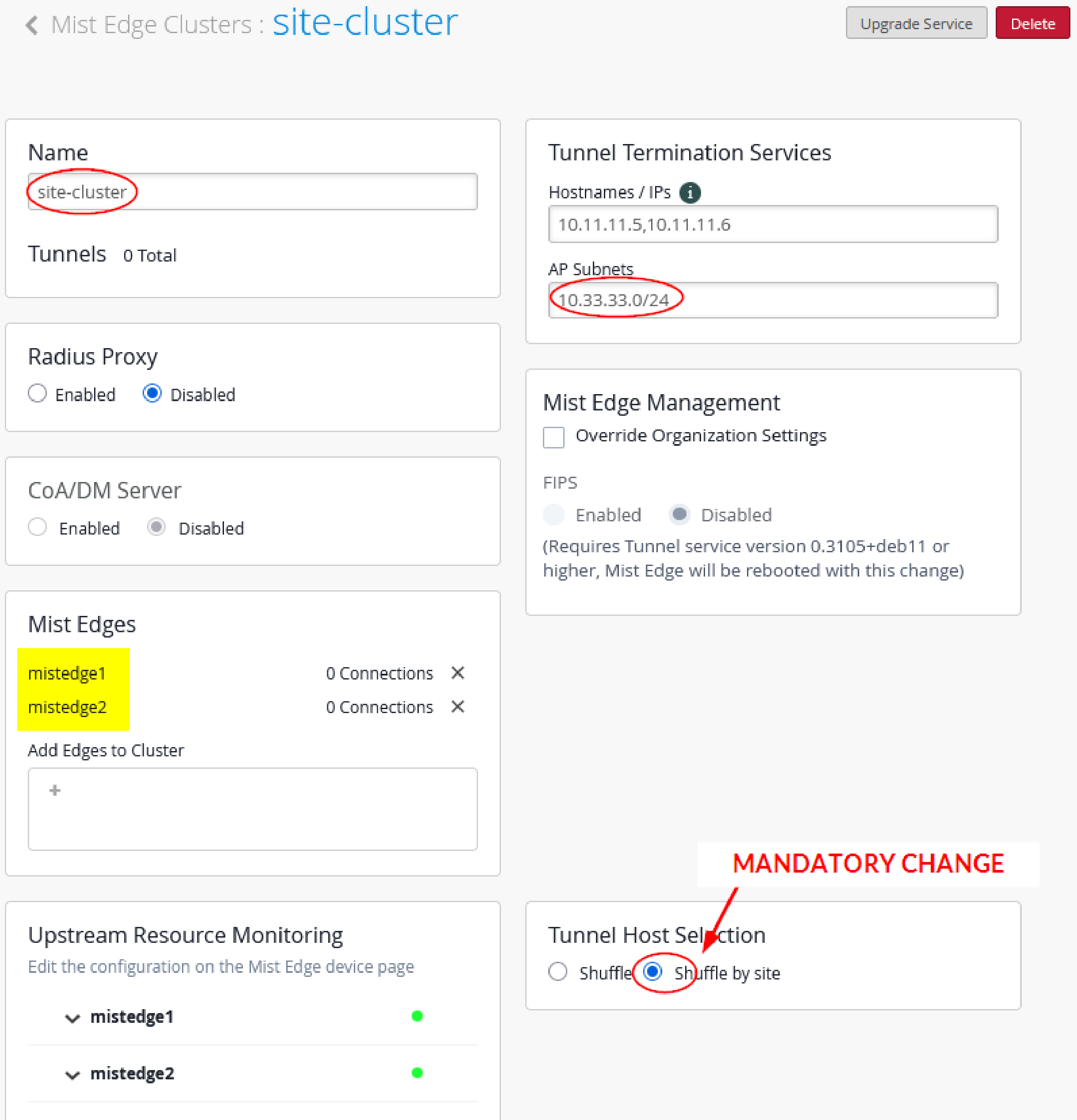
○ Enabled ● Disabled

Then, we configure our Mist Edge cluster with the two Mist Edges inside.

- Name=`site-cluster`

- Mist Edges=`mistedge1` and `mistedge2`
- Hostnames / IPs=`10.11.11.5,10.11.11.6` (happens automatically when assigning the Juniper Mist Edge)
- AP Subnets=`10.33.33.0/24` (This configuration is optional. With it enabled, the Juniper Mist Edge will only accept tunnels if the AP's source IP address falls within the range assigned to VLAN 1033)
- Tunnel Host Selection=`Shuffle by Site` (This configuration is required and ensures that all APs within the same site or fabric use the same Juniper Mist Edge)

Figure 99: Site Mist-Edge Cluster Creation



Mist Edge Clusters : site-cluster

Name: site-cluster

Tunnels 0 Total

Radius Proxy: ○ Enabled ● Disabled

CoA/DM Server: ○ Enabled ● Disabled

Mist Edges

mistedge1	0 Connections ✕
mistedge2	0 Connections ✕

Add Edges to Cluster

Upstream Resource Monitoring

Edit the configuration on the Mist Edge device page

- mistedge1 ●
- mistedge2 ●

Tunnel Termination Services

Hostnames / IPs: 10.11.11.5,10.11.11.6

AP Subnets: 10.33.33.0/24

Mist Edge Management

□ Override Organization Settings

FIPS: ○ Enabled ● Disabled

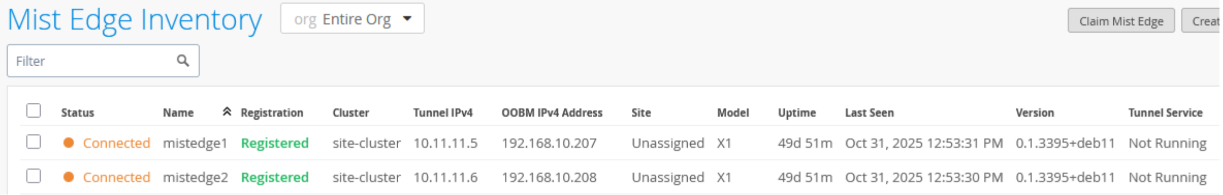
(Requires Tunnel service version 0.3105+deb11 or higher, Mist Edge will be rebooted with this change)

MANDATORY CHANGE

Tunnel Host Selection: ○ Shuffle ● Shuffle by site

Your cluster assignment should now be visible on the Juniper Mist Edges:

Figure 100: Mist-Edge Assignment to Clusters



Mist Edge Inventory   org Entire Org ▾

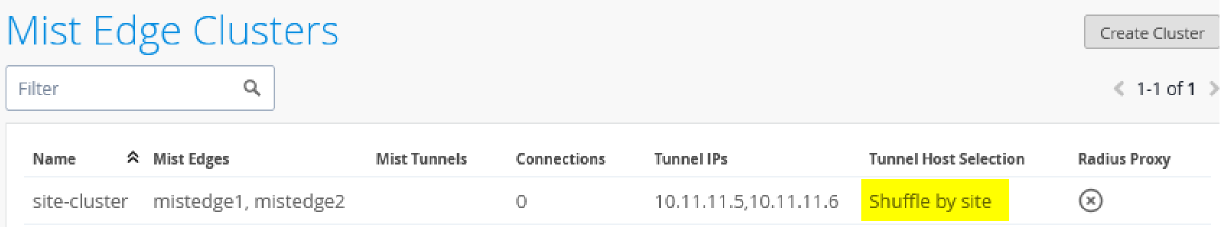
Filter

Claim Mist Edge   Creat

□	Status	Name	Registration	Cluster	Tunnel IPv4	OOBM IPv4 Address	Site	Model	Uptime	Last Seen	Version	Tunnel Service
□	● Connected	mistedge1	Registered	site-cluster	10.11.11.5	192.168.10.207	Unassigned	X1	49d 51m	Oct 31, 2025 12:53:31 PM	0.1.3395+deb11	Not Running
□	● Connected	mistedge2	Registered	site-cluster	10.11.11.6	192.168.10.208	Unassigned	X1	49d 51m	Oct 31, 2025 12:53:30 PM	0.1.3395+deb11	Not Running

The created cluster should look like the image shown below:

Figure 101: Created Mist-Edge Cluster



Mist Edge Clusters

Filter

Create Cluster

‹ 1-1 of 1 ›

Name	Mist Edges	Mist Tunnels	Connections	Tunnel IPs	Tunnel Host Selection	Radius Proxy
site-cluster	mistedge1, mistedge2		0	10.11.11.5,10.11.11.6	Shuffle by site	⊗

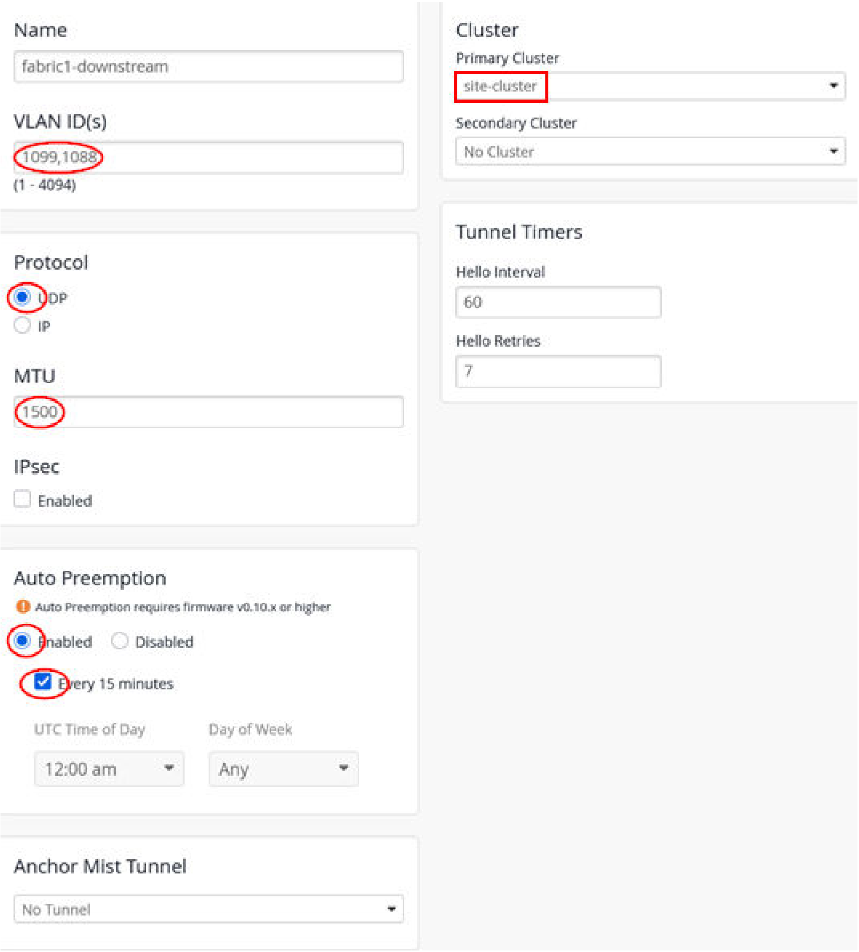
NOTE: Do not continue without selecting “Shuffle-by-site” on your cluster! It is important to instruct the APs to use a single Juniper Mist Edge as long as possible in this design.

Next, the configuration of the overlay tunnel from the AP to the Juniper Mist Edge (in our case in vlan1033):

- Name=`fabric1-downstream`
- VLAN ID=`1099, 1088` (depending on all your breakout VLANs for wireless client traffic)
- Primary Cluster=`site-cluster`
- Secondary Cluster=`No Cluster`
- Protocol=`UDP`
- MTU=`1500`
- IPsec=`Unchecked`
- Auto Preemption=`Enabled` (If an AP accidentally roams to the second Juniper Mist Edge while the primary Edge is still active and managing the other APs, the AP will automatically roam back to the primary.)

- Every 15 minutes=`Checked` (roaming back should happen as fast as possible, hence this configuration)

Figure 102: Fabric Tunnel Creation Towards Mist-Edge Clusters



Name
fabric1-downstream

VLAN ID(s)
1099,1088
(1 - 4094)

Protocol
● UDP
○ IP

MTU
1500

IPsec
☐ Enabled

Auto Preemption
Auto Preemption requires firmware v0.10.x or higher
● Enabled ○ Disabled
☑ Every 15 minutes
UTC Time of Day: 12:00 am
Day of Week: Any

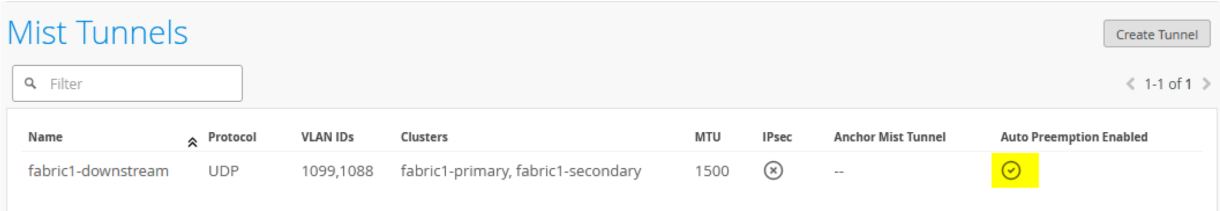
Anchor Mist Tunnel
No Tunnel

Cluster
Primary Cluster: site-cluster
Secondary Cluster: No Cluster

Tunnel Timers
Hello Interval: 60
Hello Retries: 7

The configuration you entered should result in the overview shown below:

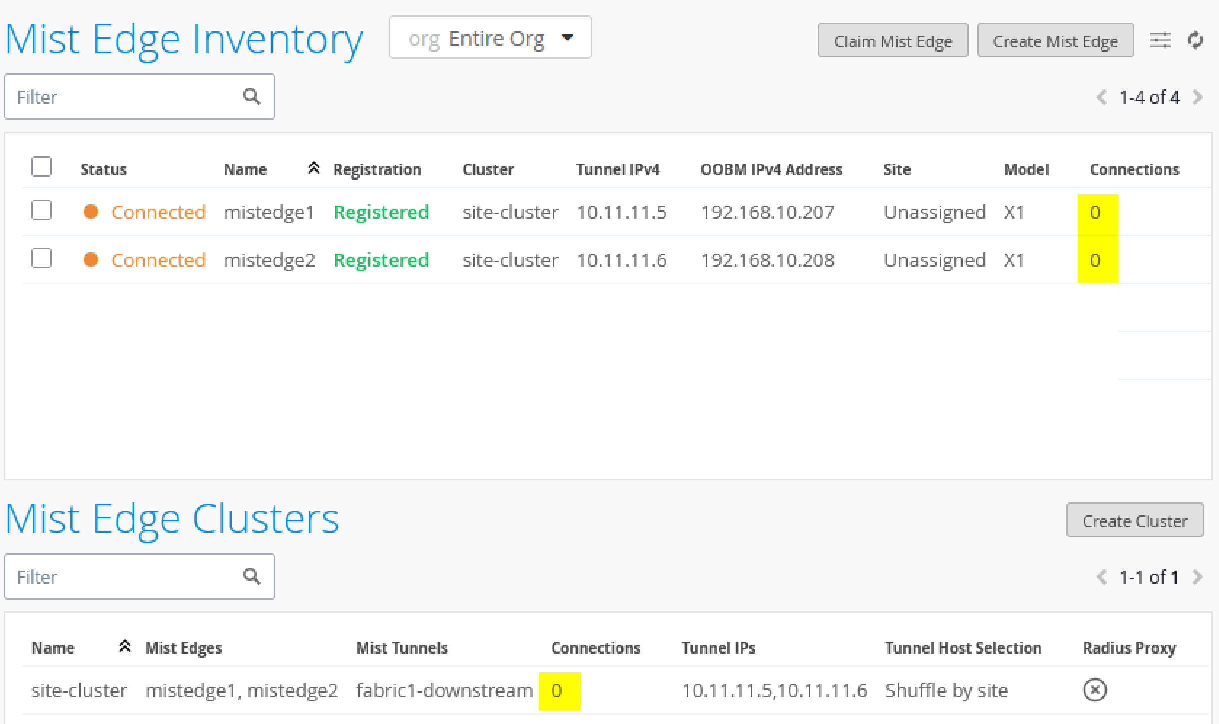
Figure 103: Mist-Edge Tunnel



Name	Protocol	VLAN IDs	Clusters	MTU	IPsec	Anchor Mist Tunnel	Auto Preemption Enabled
fabric1-downstream	UDP	1099,1088	fabric1-primary, fabric1-secondary	1500	⊗	--	✓

At this point, no APs are connecting to the Juniper Mist Edges because the tunnel configuration has not yet been specified for them:

Figure 104: Mist-Edge Tunnels Not Built Yet

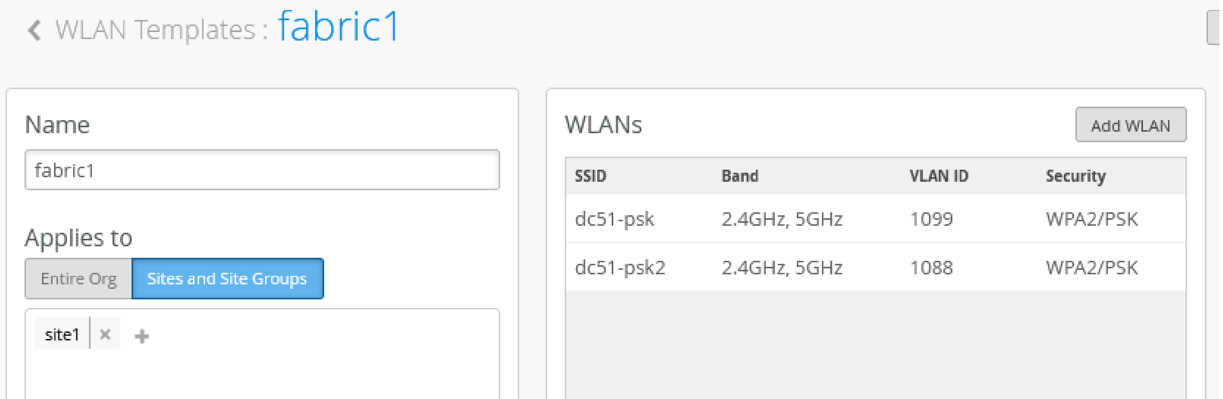


Status	Name	Registration	Cluster	Tunnel IPv4	OOBM IPv4 Address	Site	Model	Connections
● Connected	mistedge1	Registered	site-cluster	10.11.11.5	192.168.10.207	Unassigned	X1	0
● Connected	mistedge2	Registered	site-cluster	10.11.11.6	192.168.10.208	Unassigned	X1	0

Name	Mist Edges	Mist Tunnels	Connections	Tunnel IPs	Tunnel Host Selection	Radius Proxy
site-cluster	mistedge1, mistedge2	fabric1-downstream	0	10.11.11.5,10.11.11.6	Shuffle by site	⊗

As a last configuration step, we need to configure the WLAN template with the information about the tunnel configuration. Here is our fabric template with two SSIDs configured:

Figure 105: WLAN Template Config



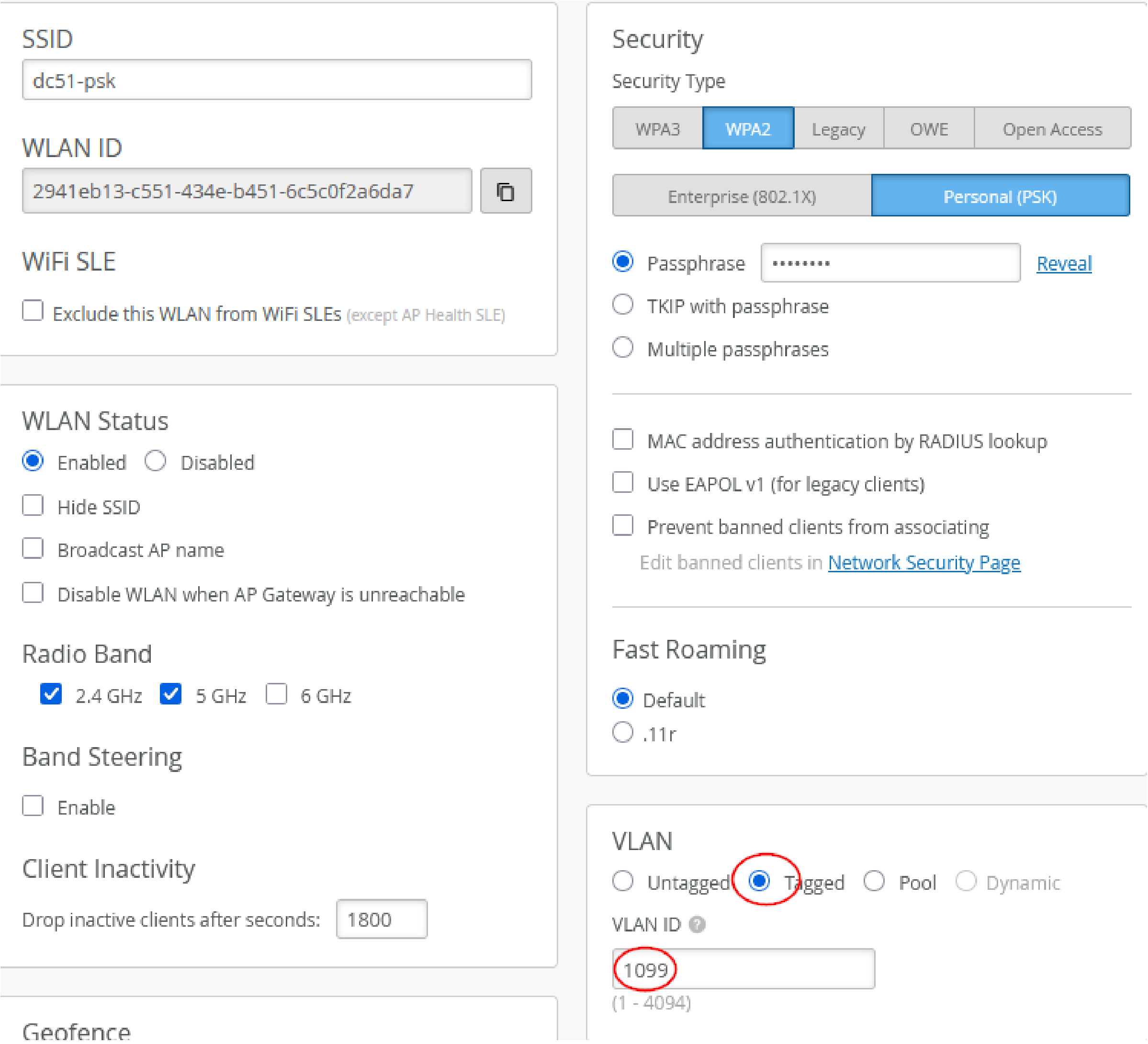
Name: fabric1

SSID	Band	VLAN ID	Security
dc51-psk	2.4GHz, 5GHz	1099	WPA2/PSK
dc51-psk2	2.4GHz, 5GHz	1088	WPA2/PSK

The configuration for our first SSID is a simple PSK. You need to configure one of the tunnel breakout VLANs as tagged as shown below:

- VLAN=`Tagged`
- VLAN ID=`1099`

Figure 106: First SSID Configuration



SSID

dc51-psk

WLAN ID

2941eb13-c551-434e-b451-6c5c0f2a6da7

WiFi SLE

☐ Exclude this WLAN from WiFi SLEs (except AP Health SLE)

WLAN Status

● Enabled ○ Disabled

☐ Hide SSID

☐ Broadcast AP name

☐ Disable WLAN when AP Gateway is unreachable

Radio Band

☑ 2.4 GHz ☑ 5 GHz ☐ 6 GHz

Band Steering

☐ Enable

Client Inactivity

Drop inactive clients after seconds: 1800

Geofence

Security

Security Type

WPA3 | **WPA2** | Legacy | OWE | Open Access

Enterprise (802.1X) | **Personal (PSK)**

● Passphrase •••••••• Reveal

○ TKIP with passphrase

○ Multiple passphrases

☐ MAC address authentication by RADIUS lookup

☐ Use EAPOL v1 (for legacy clients)

☐ Prevent banned clients from associating

Edit banned clients in Network Security Page

Fast Roaming

● Default

○ .11r

VLAN

○ Untagged ● Tagged ○ Pool ○ Dynamic

VLAN ID

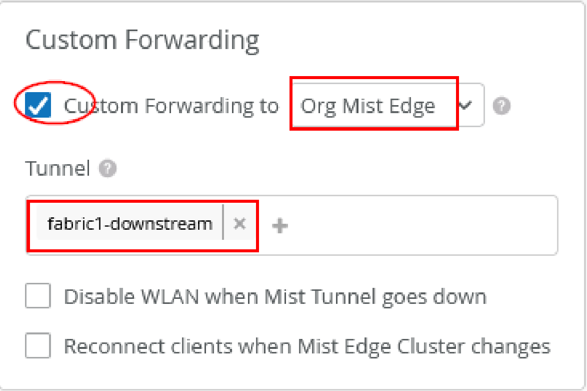
1099

(1 - 4094)

Now, the tunnel configuration for this SSID would look like:

- Custom Forwarding=`Checked`
- To=`Org Mist Edge`
- Tunnel=`fabric1-downstream`

Figure 107: Tunnel Configuration on SSID1



Custom Forwarding

☑ Custom Forwarding to Org Mist Edge

Tunnel

fabric1-downstream

□ Disable WLAN when Mist Tunnel goes down

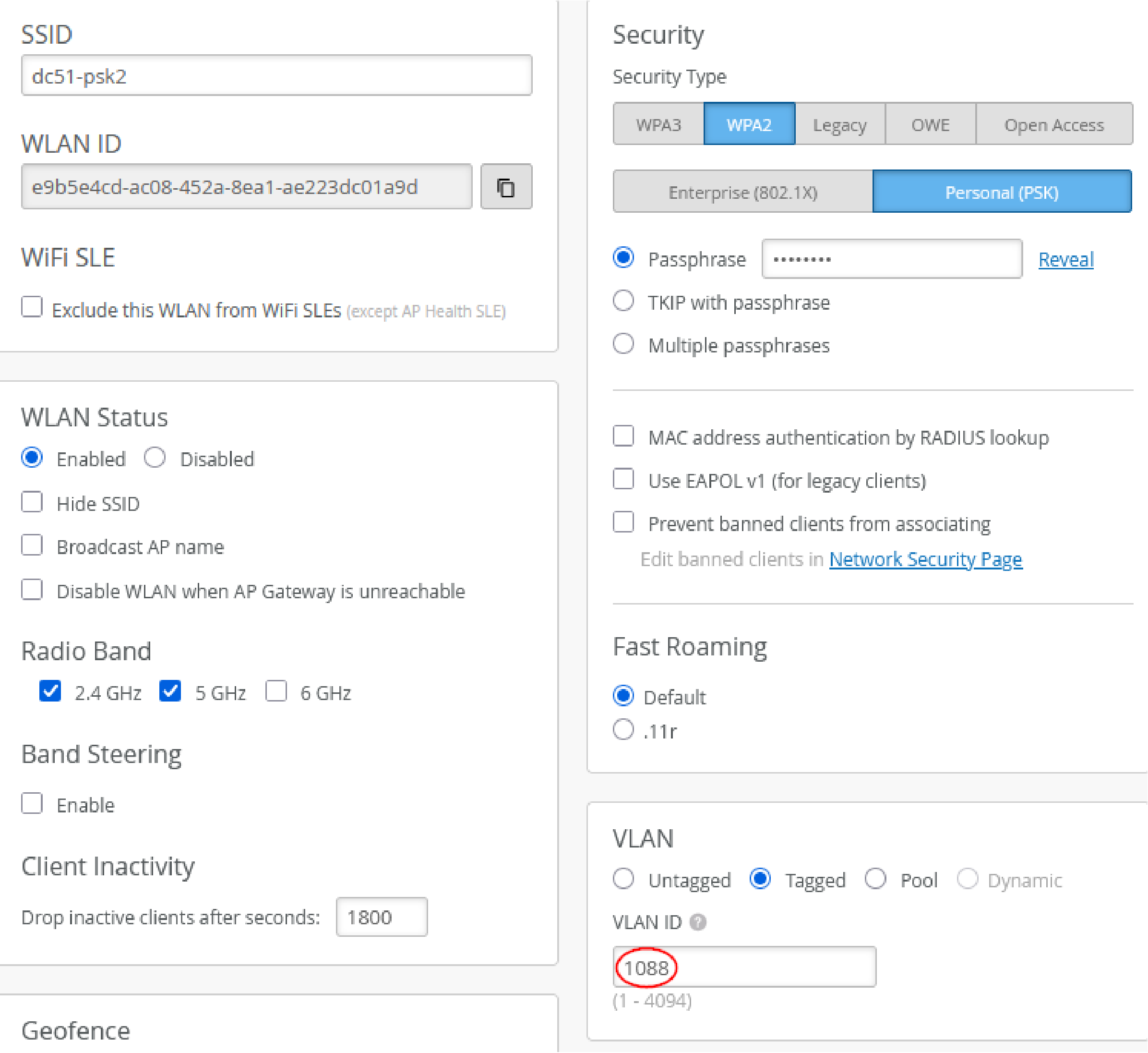
□ Reconnect clients when Mist Edge Cluster changes

NOTE: Do not enable the two additional features unless there is a specific need. Leave "Disable WLAN when Mist Tunnel goes down" and "Reconnect clients when Mist Edge Cluster changes" disabled as they are by default. These options are not required when Juniper Mist Edge is operating within an EVPN fabric, since tunnel distances are short and the new cluster typically uses upstream configurations where such features might otherwise be beneficial.

The configuration for our second SSID is again a simple PSK. You need to configure one of the tunnel breakout VLANs as tagged as shown below:

- VLAN=`Tagged`
- VLAN ID=`1088`

Figure 108: Second SSID Configuration



SSID

dc51-psk2

WLAN ID

e9b5e4cd-ac08-452a-8ea1-ae223dc01a9d

WiFi SLE

□ Exclude this WLAN from WiFi SLEs (except AP Health SLE)

WLAN Status

● Enabled ○ Disabled

□ Hide SSID

□ Broadcast AP name

□ Disable WLAN when AP Gateway is unreachable

Radio Band

☑ 2.4 GHz ☑ 5 GHz □ 6 GHz

Band Steering

□ Enable

Client Inactivity

Drop inactive clients after seconds: 1800

Geofence

Security

Security Type

WPA3 | **WPA2** | Legacy | OWE | Open Access

Enterprise (802.1X) | **Personal (PSK)**

● Passphrase •••••••• Reveal

○ TKIP with passphrase

○ Multiple passphrases

□ MAC address authentication by RADIUS lookup

□ Use EAPOL v1 (for legacy clients)

□ Prevent banned clients from associating

Edit banned clients in Network Security Page

Fast Roaming

● Default

○ .11r

VLAN

○ Untagged ● Tagged ○ Pool ○ Dynamic

VLAN ID

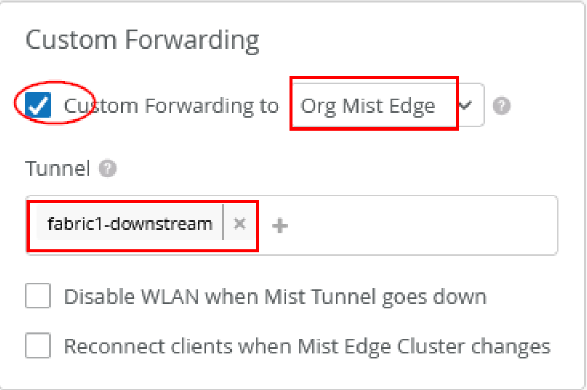
1088

(1 - 4094)

Now, the tunnel configuration for this SSID should again look like the following:

- Custom Forwarding=`Checked`
- To=`Org Mist Edge`
- Tunnel=`fabric1-downstream`

Figure 109: Tunnel Configuration on SSID2



Custom Forwarding

☑ Custom Forwarding to Org Mist Edge

Tunnel

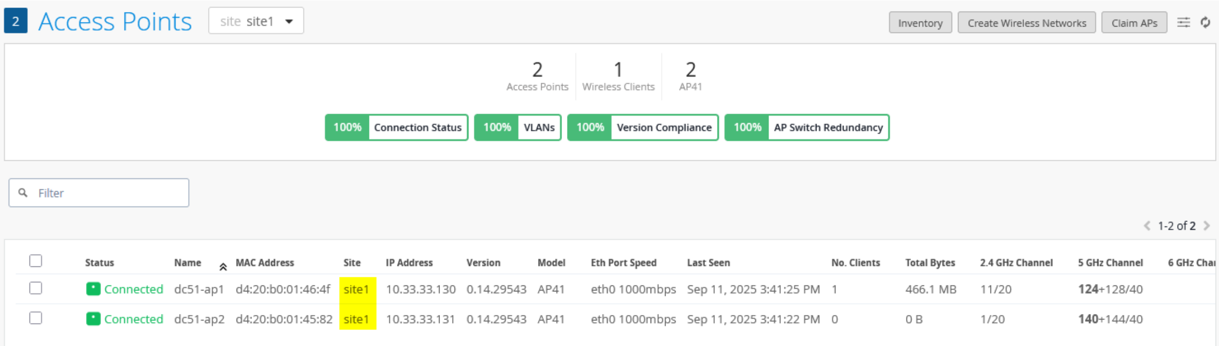
fabric1-downstream

□ Disable WLAN when Mist Tunnel goes down

□ Reconnect clients when Mist Edge Cluster changes

Also ensure that all your Access Points really belong to the same site in this design.

Figure 110: All APs Must be at Same Site



Access Points — site site1

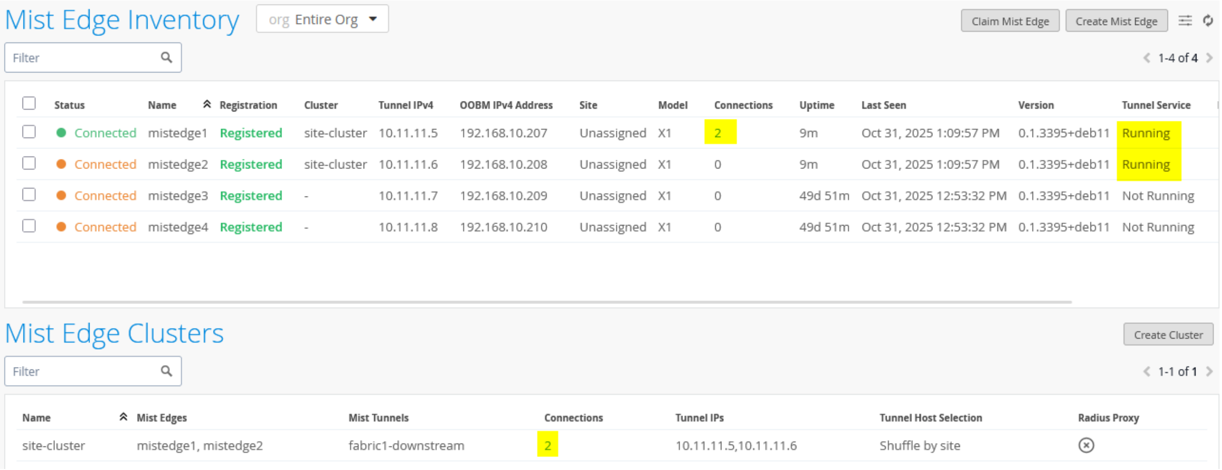
2 Access Points | 1 Wireless Clients | 2 AP41

100% Connection Status | 100% VLANs | 100% Version Compliance | 100% AP Switch Redundancy

□	Status	Name	MAC Address	Site	IP Address	Version	Model	Eth Port Speed	Last Seen	No. Clients	Total Bytes	2.4 GHz Channel	5 GHz Channel	6 GHz Cha
□	Connected	dc51-ap1	d4:20:b0:01:46:4f	site1	10.33.33.130	0.14.29543	AP41	eth0 1000mbps	Sep 11, 2025 3:41:25 PM	1	466.1 MB	11/20	**124**+128/40	
□	Connected	dc51-ap2	d4:20:b0:01:45:82	site1	10.33.33.131	0.14.29543	AP41	eth0 1000mbps	Sep 11, 2025 3:41:22 PM	0	0 B	1/20	**140**+144/40	

Once this final step is complete, the configuration is done. You should now see both APs connected to only one of the Juniper Mist Edges, as expected, since they are part of the same site and the Shuffle-by-Site option designates a specific Mist Edge for them. This confirms that the APs remain anchored to a single Mist Edge. You should also see that the tunnel service is active, indicating that the configuration is valid.

Figure 111: Access Points Now Building Tunnels with Mist-Edges

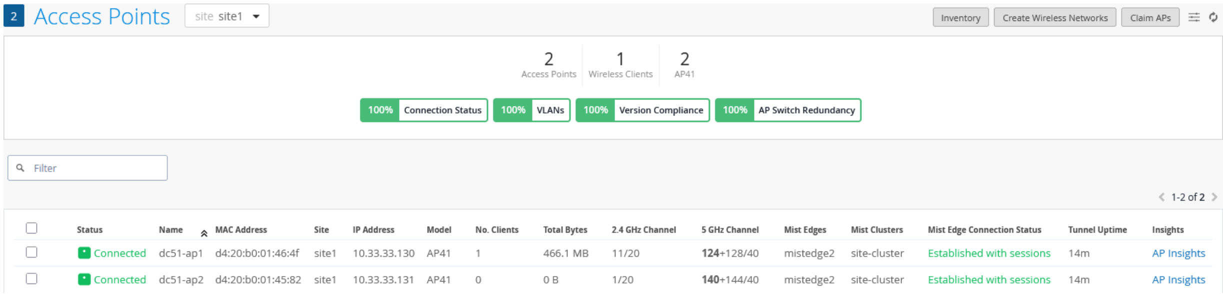


☐	Status	Name	Registration	Cluster	Tunnel IPv4	OOBM IPv4 Address	Site	Model	Connections	Uptime	Last Seen	Version	Tunnel Service
☐	● Connected	mistedge1	Registered	site-cluster	10.11.11.5	192.168.10.207	Unassigned	X1	2	9m	Oct 31, 2025 1:09:57 PM	0.1.3395+deb11	Running
☐	● Connected	mistedge2	Registered	site-cluster	10.11.11.6	192.168.10.208	Unassigned	X1	0	9m	Oct 31, 2025 1:09:57 PM	0.1.3395+deb11	Running
☐	● Connected	mistedge3	Registered	-	10.11.11.7	192.168.10.209	Unassigned	X1	0	49d 51m	Oct 31, 2025 12:53:32 PM	0.1.3395+deb11	Not Running
☐	● Connected	mistedge4	Registered	-	10.11.11.8	192.168.10.210	Unassigned	X1	0	49d 51m	Oct 31, 2025 12:53:32 PM	0.1.3395+deb11	Not Running

Name	Mist Edges	Mist Tunnels	Connections	Tunnel IPs	Tunnel Host Selection	Radius Proxy
site-cluster	mistedge1, mistedge2	fabric1-downstream	2	10.11.11.5,10.11.11.6	Shuffle by site	⊗

You also now have the AP View showing Juniper Mist Edge and Cluster:

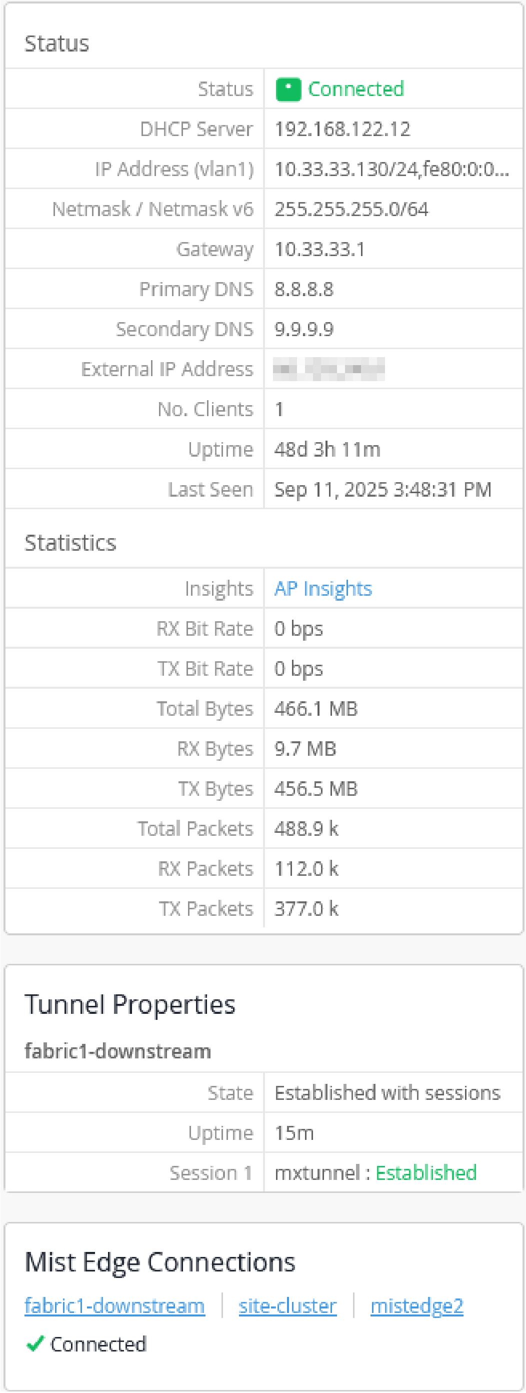
Figure 112: Mist-Edge Visibility on Access Points



☐	Status	Name	MAC Address	Site	IP Address	Model	No. Clients	Total Bytes	2.4 GHz Channel	5 GHz Channel	Mist Edges	Mist Clusters	Mist Edge Connection Status	Tunnel Uptime	Insights
☐	Connected	dc51-ap1	d4:20:b0:01:46:4f	site1	10.33.33.130	AP41	1	466.1 MB	11/20	**124**+128/40	mistedge2	site-cluster	Established with sessions	14m	AP Insights
☐	Connected	dc51-ap2	d4:20:b0:01:45:82	site1	10.33.33.131	AP41	0	0 B	1/20	**140**+144/40	mistedge2	site-cluster	Established with sessions	14m	AP Insights

And a report from a selected individual AP:

Figure 113: Mist-Edge Tunnel on AP Seen



Status

Status	🟩 Connected
DHCP Server	192.168.122.12
IP Address (vlan1)	10.33.33.130/24,fe80:0:0...
Netmask / Netmask v6	255.255.255.0/64
Gateway	10.33.33.1
Primary DNS	8.8.8.8
Secondary DNS	9.9.9.9
External IP Address	[redacted]
No. Clients	1
Uptime	48d 3h 11m
Last Seen	Sep 11, 2025 3:48:31 PM

Statistics

Insights	AP Insights
RX Bit Rate	0 bps
TX Bit Rate	0 bps
Total Bytes	466.1 MB
RX Bytes	9.7 MB
TX Bytes	456.5 MB
Total Packets	488.9 k
RX Packets	112.0 k
TX Packets	377.0 k

Tunnel Properties

fabric1-downstream

State	Established with sessions
Uptime	15m
Session 1	mxtunnel : Established

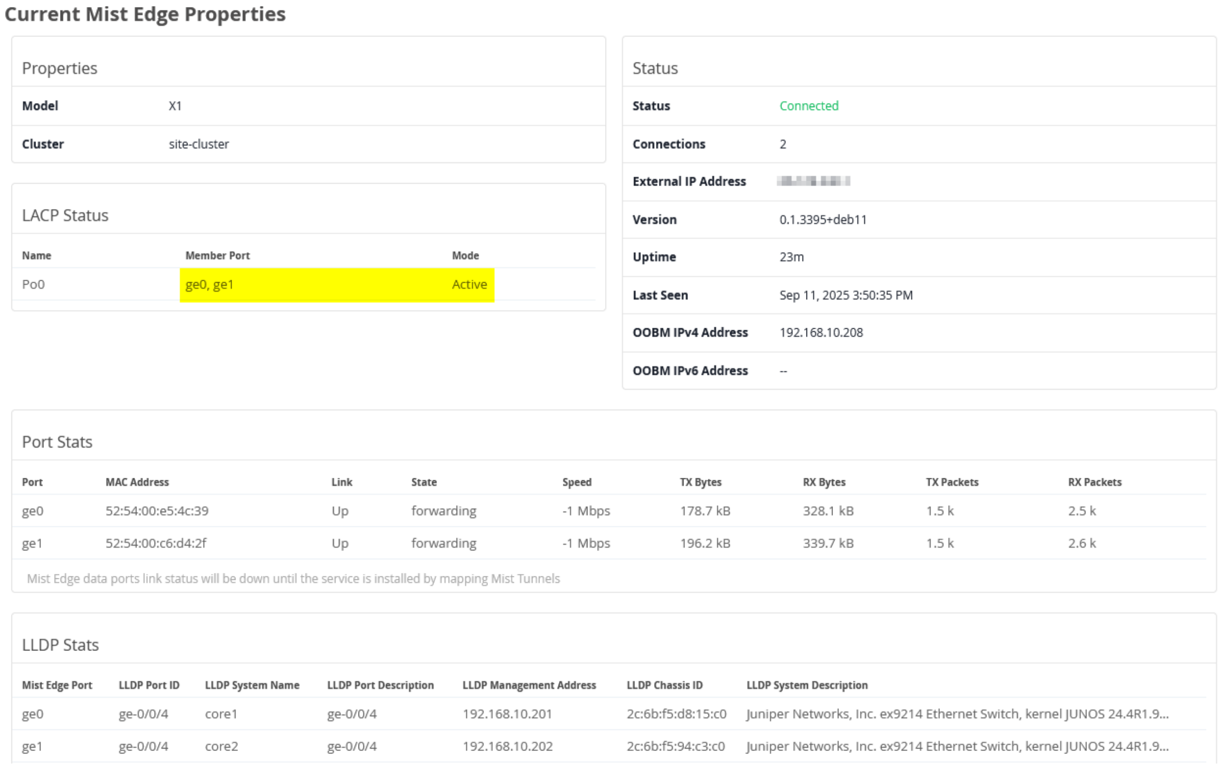
Mist Edge Connections

fabric1-downstream | site-cluster | mistedge2

✓ Connected

Also, review the Juniper Mist Edge insights as shown in the example below:

Figure 114: Mist-Edge Insights



Current Mist Edge Properties

Properties

Model	X1
Cluster	site-cluster

Status

Status	Connected
Connections	2
External IP Address	[redacted]
Version	0.1.3395+deb11
Uptime	23m
Last Seen	Sep 11, 2025 3:50:35 PM
OOBM IPv4 Address	192.168.10.208
OOBM IPv6 Address	--

LACP Status

Name	Member Port	Mode
Po0	ge0, ge1	Active

Port Stats

Port	MAC Address	Link	State	Speed	TX Bytes	RX Bytes	TX Packets	RX Packets
ge0	52:54:00:e5:4c:39	Up	forwarding	-1 Mbps	178.7 kB	328.1 kB	1.5 k	2.5 k
ge1	52:54:00:c6:d4:2f	Up	forwarding	-1 Mbps	196.2 kB	339.7 kB	1.5 k	2.6 k

Mist Edge data ports link status will be down until the service is installed by mapping Mist Tunnels

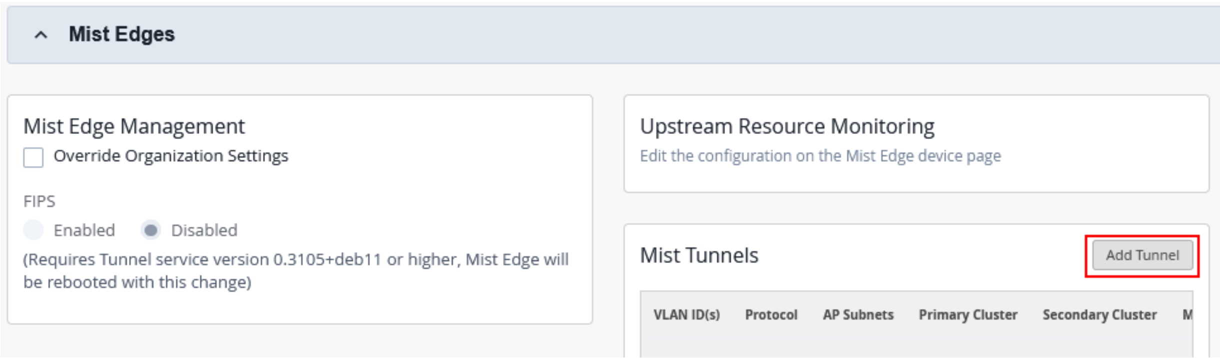
LLDP Stats

Mist Edge Port	LLDP Port ID	LLDP System Name	LLDP Port Description	LLDP Management Address	LLDP Chassis ID	LLDP System Description
ge0	ge-0/0/4	core1	ge-0/0/4	192.168.10.201	2c:6b:f5:d8:15:c0	Juniper Networks, Inc. ex9214 Ethernet Switch, kernel JUNOS 24.4R1.9...
ge1	ge-0/0/4	core2	ge-0/0/4	192.168.10.202	2c:6b:f5:94:c3:c0	Juniper Networks, Inc. ex9214 Ethernet Switch, kernel JUNOS 24.4R1.9...

Alternative Configuration Option by Site Tunnel Creation

In this example, we used global Juniper Mist Edges along with a cluster that has the Shuffle-by-Site option enabled. However, this is not the only way to configure the system. You can also deploy the Mist Edges and establish the tunnel directly through the site settings without creating a cluster. To do this, navigate to **Organization** → **Site Configuration** → **<site of your APs>** and choose **Add Tunnel**.

Figure 115: Create Mist Edge Tunnel Under Site View



Mist Edge Management

□ Override Organization Settings

FIPS

○ Enabled ● Disabled

(Requires Tunnel service version 0.3105+deb11 or higher, Mist Edge will be rebooted with this change)

Upstream Resource Monitoring

Edit the configuration on the Mist Edge device page

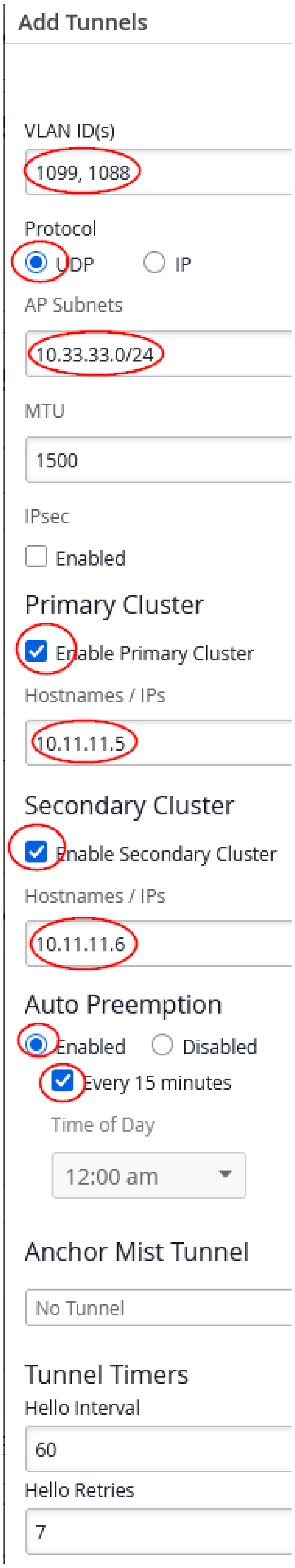
Mist Tunnels

VLAN ID(s)	Protocol	AP Subnets	Primary Cluster	Secondary Cluster	M

This is our site tunnel configuration:

- VLAN ID=`1099, 1088` (depending on all your breakout VLANs used for wireless client traffic)
- Protocol=`UDP`
- AP Subnets=`10.33.33.0/24` (optionally restricts tunnel termination to only those APs using this source IP)
- MTU=`1500`
- IPsec=`Unchecked`
- Primary Cluster Enabled=`Checked/Enabled`
  - Host IP=`10.11.11.5`
- Secondary Cluster Enabled=`Checked/Enabled`
  - Host IP=`10.11.11.6`
- Auto Preemption=`Enabled` (if an AP unintentionally roams to a second Mist Edge while the original remains active, it should return to the original Mist Edge along with the other APs)
  - Every 15 minutes=`Checked` (roaming back should happen as fast as possible hence this configuration)

Figure 116: Site Mist Edge Tunnel



Add Tunnels

VLAN ID(s)

1099, 1088

Protocol

● UDP ○ IP

AP Subnets

10.33.33.0/24

MTU

1500

IPsec

☐ Enabled

Primary Cluster

☑ Enable Primary Cluster

Hostnames / IPs

10.11.11.5

Secondary Cluster

☑ Enable Secondary Cluster

Hostnames / IPs

10.11.11.6

Auto Preemption

● Enabled ○ Disabled

☑ Every 15 minutes

Time of Day

12:00 am

Anchor Mist Tunnel

No Tunnel

Tunnel Timers

Hello Interval

60

Hello Retries

7

NOTE: Since the Shuffle-by-Site option is not available at the site configuration level, you must manually specify a primary and secondary Juniper Mist Edge. This setup ensures that APs terminate their tunnels on a single designated Mist Edge within the EVPN fabric.

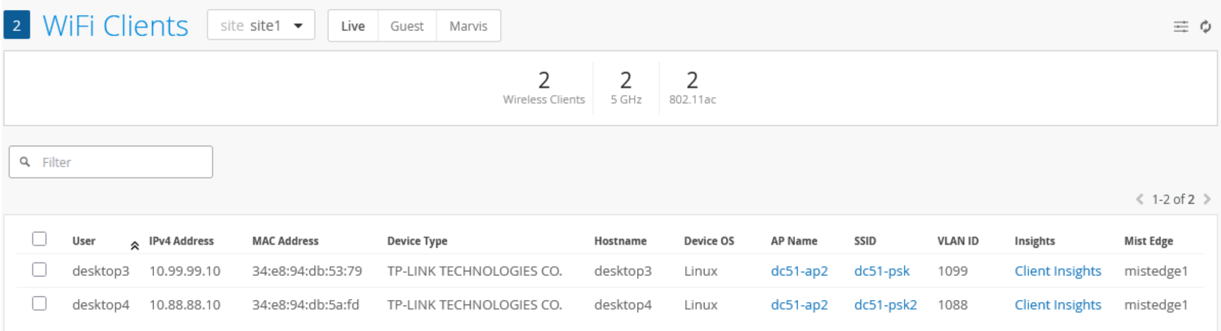
Testing Your Configuration

At this point, all required configurations are completed and you can launch your wireless client for testing. In our case:

- The Desktop3 VM connects to the SSID dc51-psk, is assigned to VLAN 1099 on the first VRF, and receives a DHCP lease.
- The Desktop4 VM connects to the SSID dc51-psk2, is assigned to VLAN 1088 on the second VRF, and receives a DHCP lease.

The two connected wireless clients are reported like this:

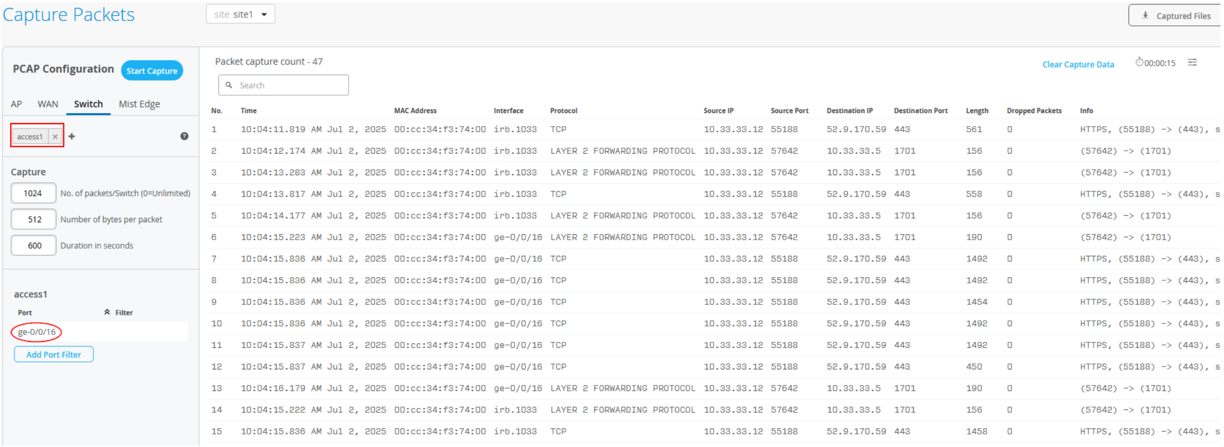
Figure 117: Wireless Client Report



□	User	IPv4 Address	MAC Address	Device Type	Hostname	Device OS	AP Name	SSID	VLAN ID	Insights	Mist Edge
□	desktop3	10.99.99.10	34:e8:94:db:53:79	TP-LINK TECHNOLOGIES CO.	desktop3	Linux	dc51-ap2	dc51-psk	1099	Client Insights	mistedge1
□	desktop4	10.88.88.10	34:e8:94:db:5a:fd	TP-LINK TECHNOLOGIES CO.	desktop4	Linux	dc51-ap2	dc51-psk2	1088	Client Insights	mistedge1

When using **Site -> Switch Packet Captures** in the Juniper Mist portal, you can capture traffic ingressing your switch port. Below, you see a trace taken from the interface where the AP is attached to the Access1 switch. You can see the HTTPS communication with the AP towards the Juniper Mist cloud as well as the L2TPv3 tunnel towards the local Juniper Mist Edge.

Figure 118: Packet Capture on the Access Switch Port



Capture

1024 No. of packets/Switch (0=Unlimited)

512 Number of bytes per packet

600 Duration in seconds

access1

Port: ge-0/0/16

Packet capture count - 47

No.	Time	MAC Address	Interface	Protocol	Source IP	Source Port	Destination IP	Destination Port	Length	Dropped Packets	Info
1	10:04:11.819 AM Jul 2, 2025	00:cc:34:f3:74:00	irb.1033	TCP	10.33.33.12	55188	52.9.170.59	443	561	0	HTTPS, (55188) -> (443), s
2	10:04:12.174 AM Jul 2, 2025	00:cc:34:f3:74:00	irb.1033	LAYER 2 FORWARDING PROTOCOL	10.33.33.12	57642	10.33.33.5	1701	156	0	(57642) -> (1701)
3	10:04:13.283 AM Jul 2, 2025	00:cc:34:f3:74:00	irb.1033	LAYER 2 FORWARDING PROTOCOL	10.33.33.12	57642	10.33.33.5	1701	156	0	(57642) -> (1701)
4	10:04:13.817 AM Jul 2, 2025	00:cc:34:f3:74:00	irb.1033	TCP	10.33.33.12	55188	52.9.170.59	443	558	0	HTTPS, (55188) -> (443), s
5	10:04:14.177 AM Jul 2, 2025	00:cc:34:f3:74:00	irb.1033	LAYER 2 FORWARDING PROTOCOL	10.33.33.12	57642	10.33.33.5	1701	156	0	(57642) -> (1701)
6	10:04:15.223 AM Jul 2, 2025	00:cc:34:f3:74:00	ge-0/0/16	LAYER 2 FORWARDING PROTOCOL	10.33.33.12	57642	10.33.33.5	1701	190	0	(57642) -> (1701)
7	10:04:15.836 AM Jul 2, 2025	00:cc:34:f3:74:00	ge-0/0/16	TCP	10.33.33.12	55188	52.9.170.59	443	1492	0	HTTPS, (55188) -> (443), s
8	10:04:15.836 AM Jul 2, 2025	00:cc:34:f3:74:00	ge-0/0/16	TCP	10.33.33.12	55188	52.9.170.59	443	1492	0	HTTPS, (55188) -> (443), s
9	10:04:15.836 AM Jul 2, 2025	00:cc:34:f3:74:00	ge-0/0/16	TCP	10.33.33.12	55188	52.9.170.59	443	1454	0	HTTPS, (55188) -> (443), s
10	10:04:15.836 AM Jul 2, 2025	00:cc:34:f3:74:00	ge-0/0/16	TCP	10.33.33.12	55188	52.9.170.59	443	1492	0	HTTPS, (55188) -> (443), s
11	10:04:15.837 AM Jul 2, 2025	00:cc:34:f3:74:00	ge-0/0/16	TCP	10.33.33.12	55188	52.9.170.59	443	1492	0	HTTPS, (55188) -> (443), s
12	10:04:15.837 AM Jul 2, 2025	00:cc:34:f3:74:00	ge-0/0/16	TCP	10.33.33.12	55188	52.9.170.59	443	450	0	HTTPS, (55188) -> (443), s
13	10:04:16.179 AM Jul 2, 2025	00:cc:34:f3:74:00	ge-0/0/16	LAYER 2 FORWARDING PROTOCOL	10.33.33.12	57642	10.33.33.5	1701	190	0	(57642) -> (1701)
14	10:04:15.222 AM Jul 2, 2025	00:cc:34:f3:74:00	irb.1033	LAYER 2 FORWARDING PROTOCOL	10.33.33.12	57642	10.33.33.5	1701	156	0	(57642) -> (1701)
15	10:04:15.836 AM Jul 2, 2025	00:cc:34:f3:74:00	irb.1033	TCP	10.33.33.12	55188	52.9.170.59	443	1458	0	HTTPS, (55188) -> (443), s

The output below is gathered from the Core1 switch acting as the service block for the integration of the two Juniper Mist Edges:

```
root@core1> show lldp neighbors
Local Interface    Parent Interface    Chassis Id                               Port
.
ge-0/0/1           -                   2c:6b:f5:5e:9a:c0                        evpn_uplink-
to-0200043ef581 dist1
ge-0/0/2           -                   2c:6b:f5:d9:94:c0                        evpn_uplink-
to-0200043ef581 dist2
ge-0/0/3           ae5                 562537d5-60c9-45cd-9883-93bd614382d3
port0               mistedge1
ge-0/0/4           ae6                 fbed3379-875a-4329-a957-03c64788d8ef
port0               mistedge2
.
root@core1> show lacp interfaces
.
.
Aggregated interface: ae5
    LACP state:           Role   Exp   Def  Dist  Col  Syn  Aggr  Timeout  Activity
      ge-0/0/3           Actor    No    No   Yes  Yes  Yes   Yes     Fast    Active
      ge-0/0/3         Partner    No    No   Yes  Yes  Yes   Yes     Fast    Active
    LACP protocol:        Receive State  Transmit State          Mux State
      ge-0/0/3                  Current   Fast periodic Collecting distributing
Aggregated interface: ae6
    LACP state:           Role   Exp   Def  Dist  Col  Syn  Aggr  Timeout  Activity
      ge-0/0/4           Actor    No    No   Yes  Yes  Yes   Yes     Fast    Active
      ge-0/0/4         Partner    No    No   Yes  Yes  Yes   Yes     Fast    Active
    LACP protocol:        Receive State  Transmit State          Mux State
      ge-0/0/4                  Current   Fast periodic Collecting distributing
.
root@core1> 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,
           B - Blocked MAC)
.
Ethernet switching table : 14 entries, 14 learned
Routing instance : evpn_vs
   Vlan                MAC                 MAC       GBP    Logical                SVLBNH/
Active
   name                address             flags     tag    interface              VENH Index
source
.
   vlan1033            d4:20:b0:01:45:82   DR               vtep.32769
172.16.254.6
   vlan1033            d4:20:b0:01:46:4f   DR               vtep.32770
172.16.254.5
   vlan1033            fa:13:47:b8:a3:c6   DLR              ae5.0
   vlan1033            fe:54:00:42:27:e4   DR               ae6.0
   vlan1088            34:e8:94:db:5a:fd   DLR              ae5.0
   vlan1088            52:54:00:3d:91:08   DR               vtep.32769
172.16.254.6
   vlan1099            1c:fd:08:77:93:4b   DR               vtep.32770
172.16.254.5
   vlan1099            52:54:00:2c:80:63   DR               vtep.32770
172.16.254.5
```

The below is gathered from the Core2 switch acting as the service block for the integration of the two Juniper Mist Edges:

```
root@core2> show lldp neighbors
Local Interface    Parent Interface    Chassis Id                                Port
info          System Name
.
ge-0/0/1           -                   2c:6b:f5:5e:9a:c0                         evpn_uplink-
to-020004070deb dist1
ge-0/0/2           -                   2c:6b:f5:d9:94:c0                         evpn_uplink-
to-020004070deb dist2
ge-0/0/3           ae5                 562537d5-60c9-45cd-9883-93bd614382d3
```

```
port1               mistedge1
ge-0/0/4           ae6                 fbed3379-875a-4329-a957-03c64788d8ef
port1               mistedge2
.
root@core2> show lacp interfaces
.
.
Aggregated interface: ae5
    LACP state:           Role   Exp   Def  Dist  Col  Syn  Aggr  Timeout  Activity
      ge-0/0/3           Actor    No    No   Yes  Yes  Yes   Yes     Fast    Active
      ge-0/0/3         Partner    No    No   Yes  Yes  Yes   Yes     Fast    Active
    LACP protocol:        Receive State  Transmit State          Mux State
      ge-0/0/3                  Current   Fast periodic Collecting distributing
Aggregated interface: ae6
    LACP state:           Role   Exp   Def  Dist  Col  Syn  Aggr  Timeout  Activity
      ge-0/0/4           Actor    No    No   Yes  Yes  Yes   Yes     Fast    Active
      ge-0/0/4         Partner    No    No   Yes  Yes  Yes   Yes     Fast    Active
    LACP protocol:        Receive State  Transmit State          Mux State
      ge-0/0/4                  Current   Fast periodic Collecting distributing
root@core2> 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,
           B - Blocked MAC)
.
Ethernet switching table : 15 entries, 15 learned
Routing instance : evpn_vs
   Vlan                MAC                 MAC       GBP    Logical                SVLBNH/
Active
   name                address             flags     tag    interface              VENH Index
source
.
   vlan1033            d4:20:b0:01:45:82   DR               vtep.32769
172.16.254.6
   vlan1033            d4:20:b0:01:46:4f   DR               vtep.32770
172.16.254.5
   vlan1033            fa:13:47:b8:a3:c6   DLR              ae5.0
   vlan1033            fe:54:00:42:27:e4   DL               ae6.0
   vlan1088            34:e8:94:db:5a:fd   DLR              ae5.0
   vlan1088            52:54:00:3d:91:08   DR               vtep.32769
172.16.254.6
   vlan1099            1c:fd:08:77:93:4b   DR               vtep.32770
```

```
172.16.254.5
   vlan1099           34:e8:94:db:53:79   DLR              ae5.0
   vlan1099           52:54:00:2c:80:63   DR               vtep.32770
172.16.254.5
```

Building a Hybrid Lab with Four Juniper Mist Edges and ESI-LAG Fabric Attach for Scale

Sometimes relying on one Juniper Mist Edge with another acting as standby is not sufficient for larger deployments. In such cases, you need to consider how to increase capacity and scale the system.

Overall, there are two main expansion strategies:

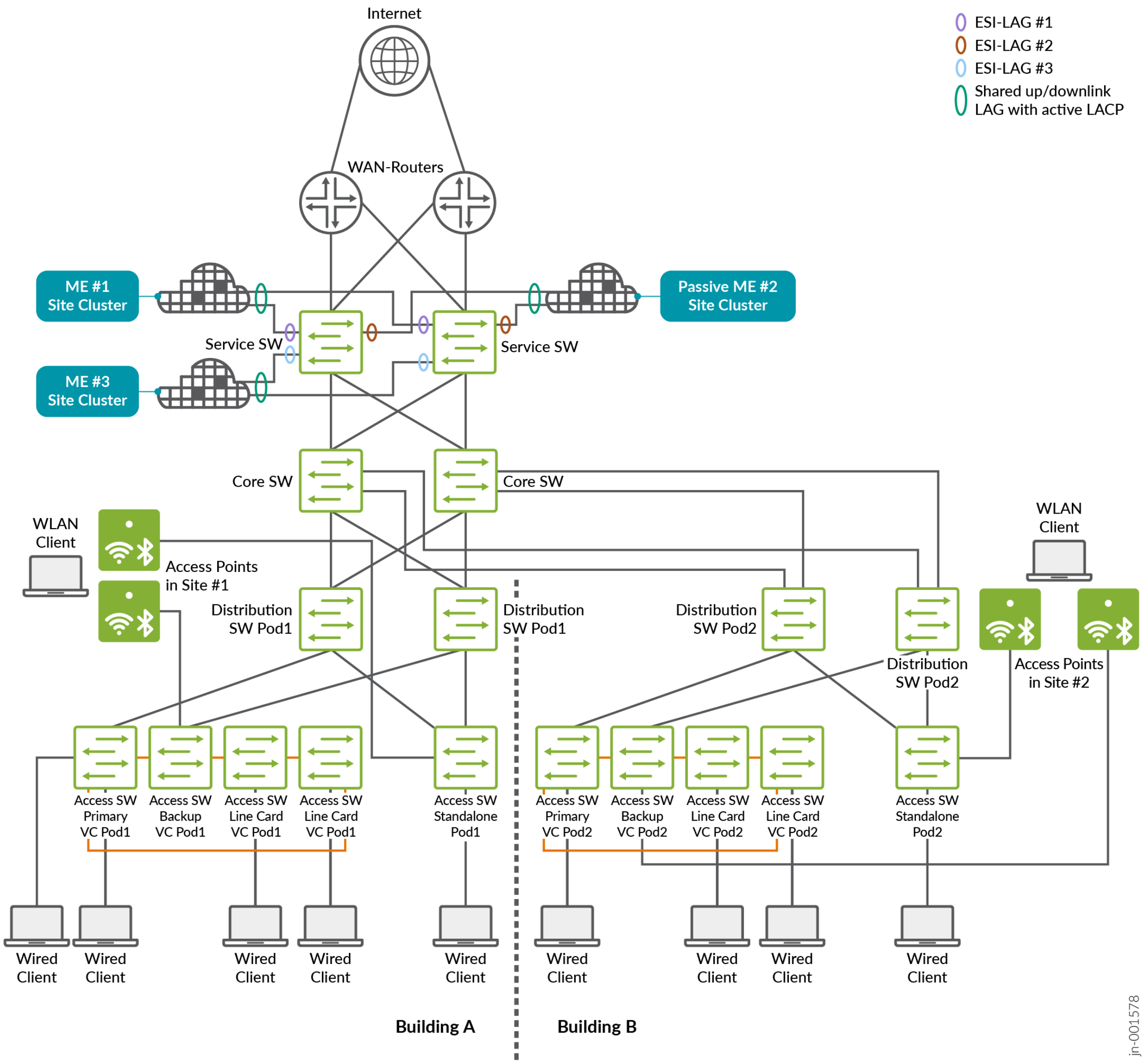
- Expanding the single cluster: This involves adding more Juniper Mist Edges to the current cluster described in the previous chapter and dividing the APs across multiple sites, so they begin using different Mist Edges based on the shuffle-by-site feature. This follows an N+1 redundancy model, where N is defined per site. With this method, all Mist Edges in the cluster must be the same model or SKU, since you cannot control which Mist Edge each AP selects within a single cluster.
- Expanding by introducing pairs of clusters: In this method, the system is scaled by adding two new clusters at a time, designated as primary and secondary. Only one Mist Edge in each cluster is active. This uses an N (Primary Cluster) + N (Secondary Cluster) redundancy model, where N is defined as a mobility domain (explained further below). The Mist Edges within a cluster pair must be the same model or SKU, but different cluster pairs can use different models, giving you more flexibility.

Expanding the Single Cluster

This method is straightforward: you expand the existing single-device cluster by adding more Mist Edges, then distribute the APs across several sites instead of placing them all in one. The key requirement is that each site should have roughly the same number of APs. You also need to ensure that the number of AP sites does not exceed the number of Mist Edges in the cluster minus the spare unit. Following these guidelines helps keep the design manageable and effective. Additionally, when dividing APs among different sites, make sure that client roaming does not occur between those sites; otherwise, client traffic will move away from the Mist Edge it is normally anchored to.

In the image below, the capacity of one Mist Edge along with its redundancy backup was reached, so a third Mist Edge was added. The APs were then divided into two separate sites. With this setup, and by enabling the shuffle-by-site option, the APs in the first site will anchor their traffic to one specific Mist Edge in the cluster, while the APs in the second site will use one of the two remaining Mist Edges for traffic anchoring.

Figure 119: Growing a Single Cluster Example



Expanding by Introducing Pairs of Clusters

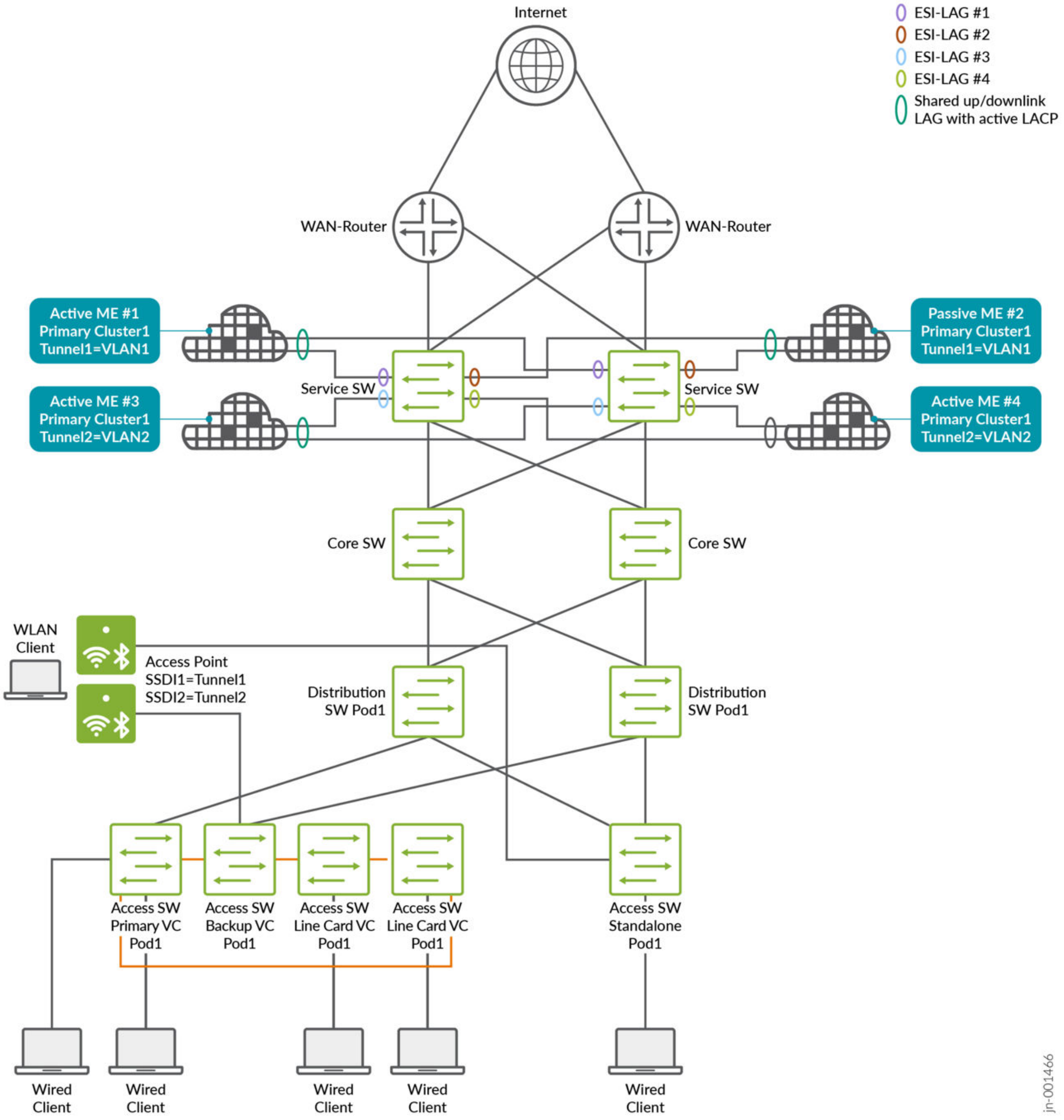
We suggest this design when a single cluster reaches its scaling limits. While slightly more complex, it provides greater flexibility and finer control over resource usage. Rather than expanding by dividing sites, you should focus on the traffic distribution needs and begin planning around the idea of a mobility domain.

To expand capacity, the network must be segmented based on traffic breakout destinations, such as specific clusters. In the last chapter the entire fabric was treated as a single unit, without introducing the concept of a mobility domain. At higher scale, however, it becomes important to evaluate how wireless clients use network resources, considering both their locations and VLAN boundaries.

As a convention, a mobility domain in most designs allows a client to roam inside a single SSID without the need for refreshing its IP address for further network communication.

A mobility domain can be defined at its smallest level as a single VLAN within an SSID, where client traffic exits upstream into the fabric network. In the example in the last chapter, two client VLANs were multiplexed into one tunnel from the access point to a Juniper Mist Edge cluster with primary and secondary devices. In the updated design, two separate tunnels are used, with each VLAN mapped to one tunnel. This approach makes it possible to introduce a new pair of Juniper Mist Edges that only handle the traffic of the second tunnel and its associated VLAN. This change effectively splits the traffic load and provides the ability to scale by adding more Juniper Mist Edge pairs as needed.

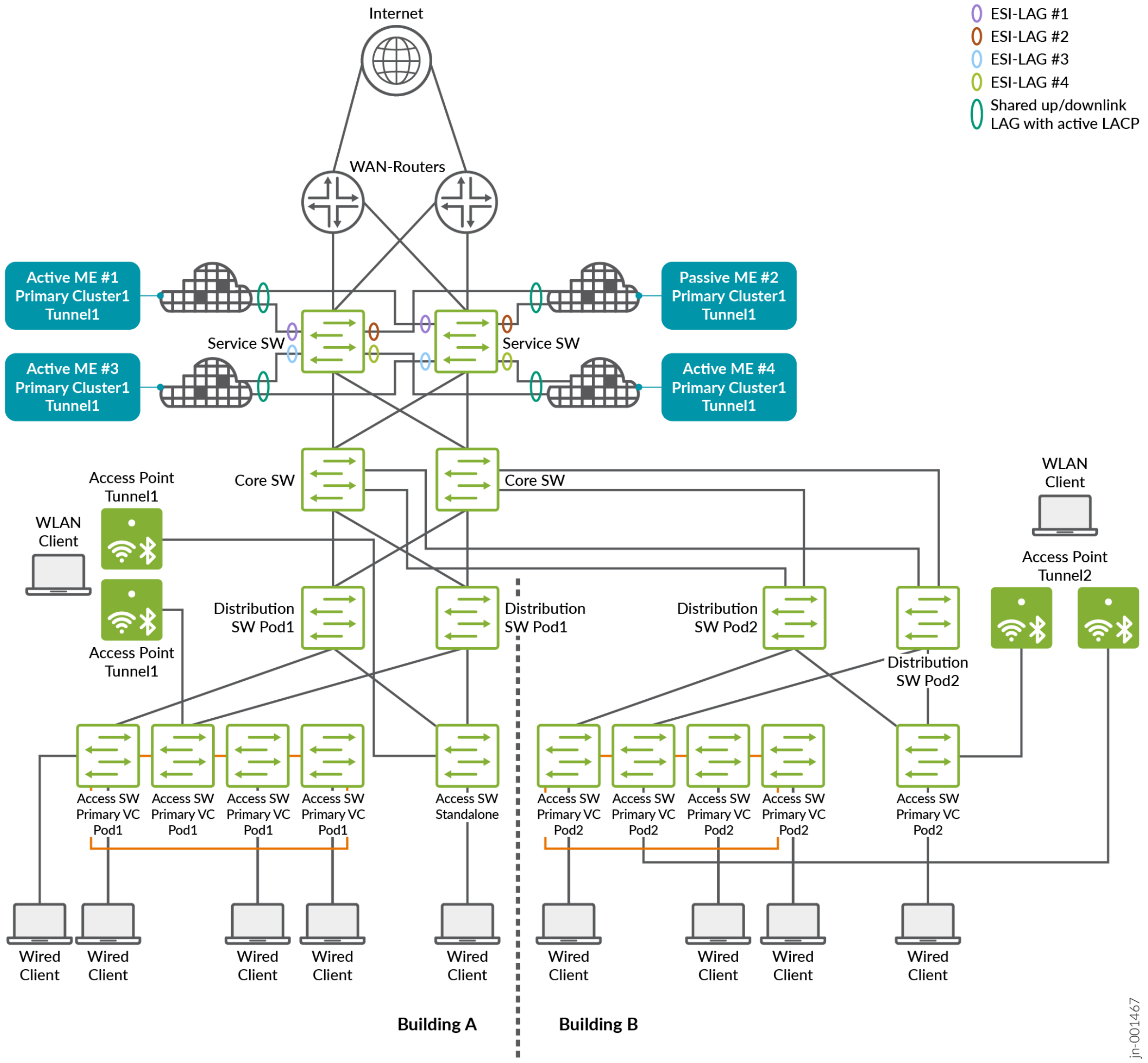
Figure 120: Two Mist Edge Cluster Pairs for Scale


ESI-LAG #1
ESI-LAG #2
ESI-LAG #3
ESI-LAG #4
Shared up/downlink LAG with active LACP
Internet
WAN-Router
WAN-Router
Active ME #1 Primary Cluster1 Tunnel1=VLAN1
Active ME #3 Primary Cluster1 Tunnel2=VLAN2
Passive ME #2 Primary Cluster1 Tunnel1=VLAN1
Active ME #4 Primary Cluster1 Tunnel2=VLAN2
Service SW
Service SW
Core SW
Core SW
WLAN Client
Access Point SSDI1=Tunnel1 SSDI2=Tunnel2
Distribution SW Pod1
Distribution SW Pod1
Access SW Primary VC Pod1
Access SW Backup VC Pod1
Access SW Line Card VC Pod1
Access SW Line Card VC Pod1
Access SW Standalone Pod1
Wired Client
Wired Client
Wired Client
Wired Client
Wired Client
jn-001466

Another way to view mobility domains and their scalability is by considering the geographic location of wireless clients. For example, in a larger campus network using a PoD design (like that shown in the image below), a second set of distribution and access switches may be deployed for an additional building connected to the same fabric. In this scenario, we assume that for technical reasons the same wireless client VLANs must be supported across both PoDs simultaneously. However, clients are unlikely to roam, or will rarely roam, between the two buildings, with each PoD representing one building.

To increase scalability in this setup, the workload can be divided by assigning a dedicated pair of Juniper Mist Edges to each PoD, allowing them to scale independently. This is accomplished by creating two tunnels that each carry all VLANs. Access points in each PoD are then configured to use only the tunnel associated with their respective PoD, which is tied to the Juniper Mist Edge cluster serving that location.

Figure 121: Mist Edge Scale Design When Using PoDs


Internet
ESI-LAG #1
ESI-LAG #2
ESI-LAG #3
ESI-LAG #4
Shared up/downlink LAG with active LACP
WAN-Routers
Active ME #1 Primary Cluster1 Tunnel1
Passive ME #2 Primary Cluster1 Tunnel1
Service SW
Service SW
Active ME #3 Primary Cluster1 Tunnel1
Active ME #4 Primary Cluster1 Tunnel1
Access Point Tunnel1
Core SW
Core SW
WLAN Client
WLAN Client
Access Point Tunnel2
Distribution SW Pod1
Distribution SW Pod1
Distribution SW Pod2
Distribution SW Pod2
Access Point Tunnel1
Access SW Primary VC Pod1
Access SW Primary VC Pod1
Access SW Primary VC Pod1
Access SW Primary VC Pod1
Access SW Standalone
Access SW Primary VC Pod2
Access SW Primary VC Pod2
Access SW Primary VC Pod2
Access SW Primary VC Pod2
Access SW Primary VC Pod2
Wired Client
Wired Client
Wired Client
Wired Client
Wired Client
Wired Client
Wired Client
Wired Client
Wired Client
Building A
Building B
jn-001467

There are many other ways to define how the Juniper Mist Edge cluster resources are used and how one can segment the traffic to accommodate higher scale. We cannot list all of them here as they change individually for each customers' network requirements.

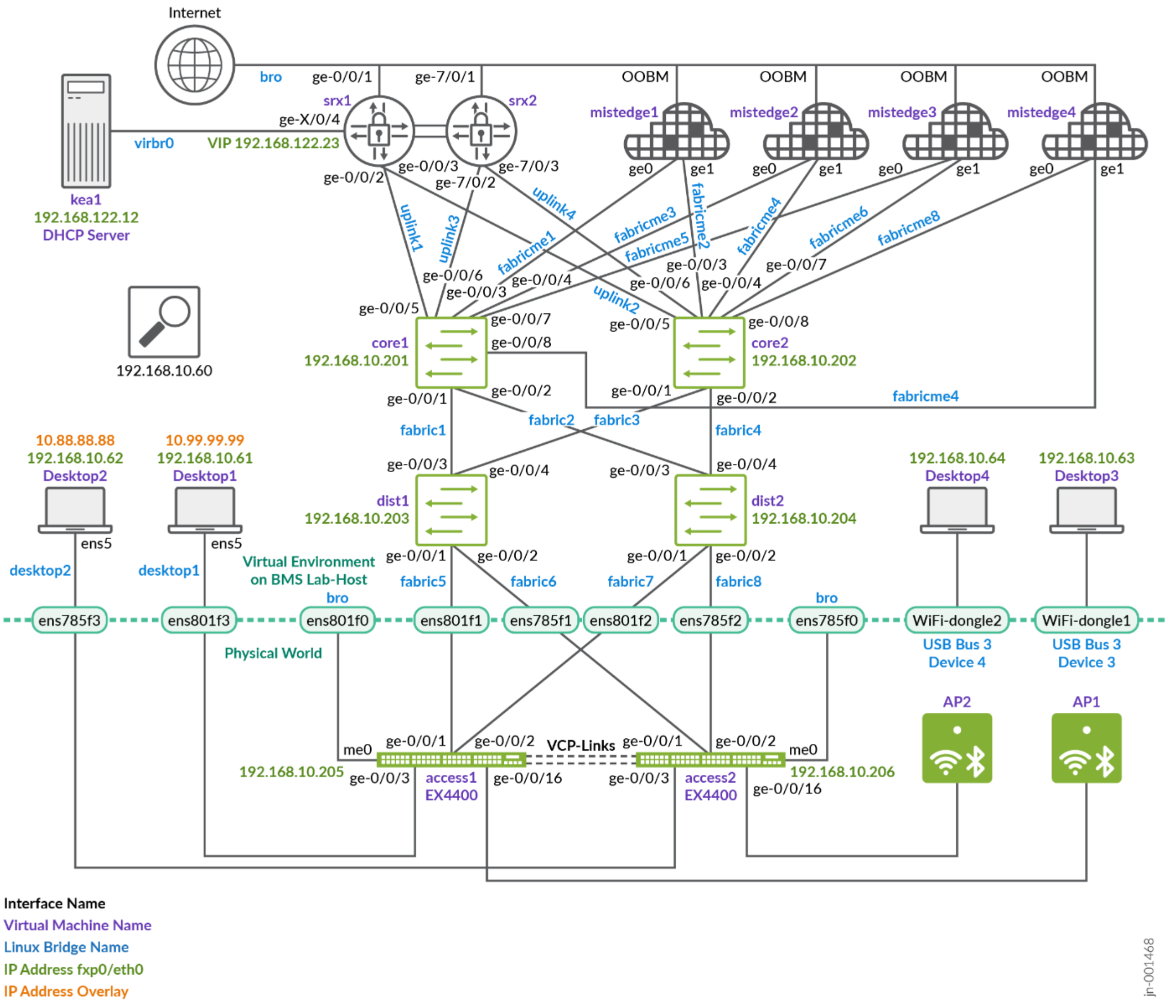
NOTE: A single mobility domain implemented by a pair of redundant Juniper Mist Edges shall not exceed 4,000 connected APs.

The lab example seen below shows the more common approach in distributing VLANs across pairs of Juniper Mist Edge clusters.

Lab Preparation

In our lab we added the two additional Juniper Mist Edge switches as shown below:

Figure 122: Lab with Four Mist Edges for Scale



Juniper Mist Edge and Fabric Configuration

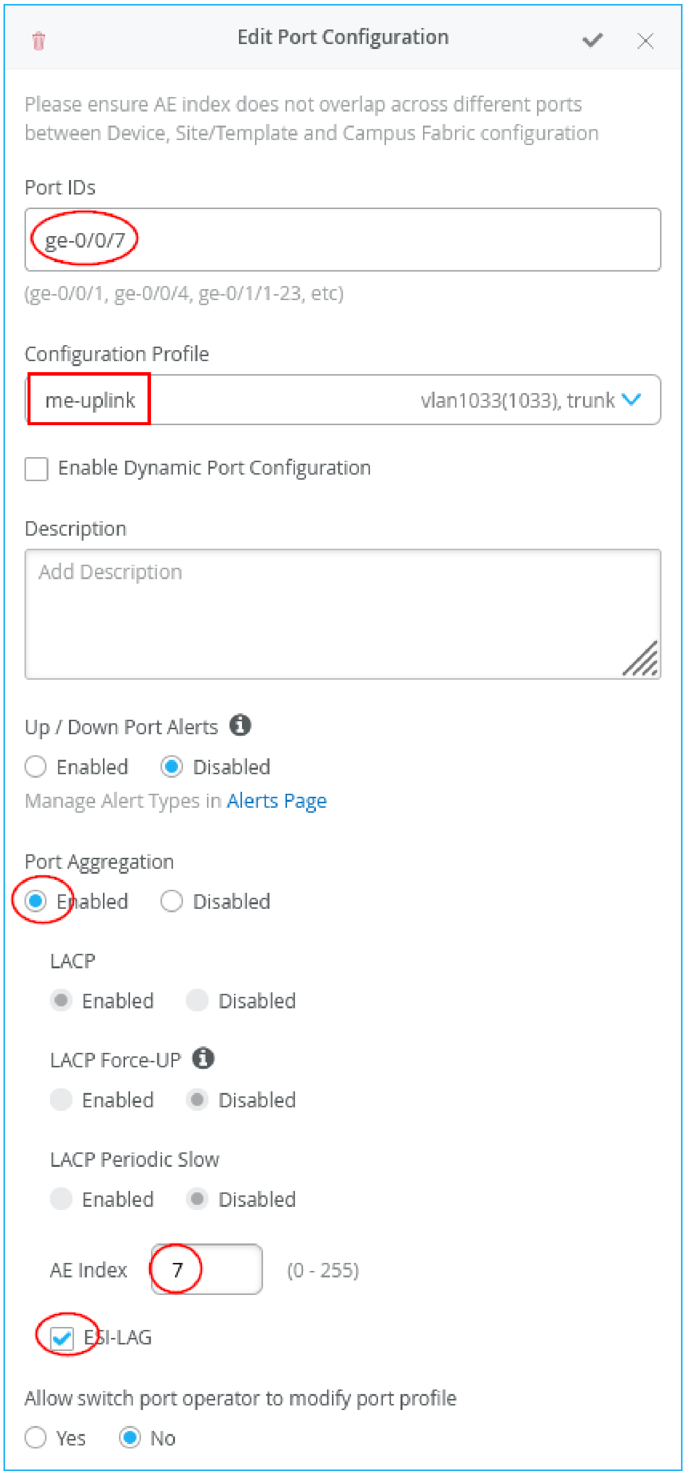
We repeat the exact same configuration for fabric, service blocks, Juniper Mist Edges, APs and access switch as in the previous "Building a Hybrid Lab with Two Juniper Mist Edges and ESI-LAG Fabric Attach" on page 174. The changes needed are:

- Add ESI-LAGs on the service-block function
- Add two Juniper Mist Edge clusters
- Add one more tunnel
- Change the cluster and tunnel for split mobility support
- Optional: Change the AP site assignment for split mobility support

The configuration of Core1-Switch and Core2-Switch for the **third Juniper Mist Edge** is:

- Port ID=`ge-0/0/7`
- Interface=`L2 interface`
- Configuration Profile=`me-uplink`
- Port Aggregation=`Enabled`
  - AE Index=`7` (you need to ensure this index value is unique and only used on the interface towards the same Juniper Mist Edge)
  - ESI-LAG=`Enabled` (enabling this is mandatory)

Figure 123: ESI-LAG for Mist-Edge Three



Edit Port Configuration

Please ensure AE index does not overlap across different ports between Device, Site/Template and Campus Fabric configuration

Port IDs: ge-0/0/7

(ge-0/0/1, ge-0/0/4, ge-0/1/1-23, etc)

Configuration Profile: me-uplink — vlan1033(1033), trunk

□ Enable Dynamic Port Configuration

Description: Add Description

Up / Down Port Alerts: ○ Enabled ● Disabled

Manage Alert Types in Alerts Page

Port Aggregation: ● Enabled ○ Disabled

LACP: ● Enabled ○ Disabled

LACP Force-UP: ○ Enabled ● Disabled

LACP Periodic Slow: ○ Enabled ● Disabled

AE Index: 7 (0 - 255)

☑ ESI-LAG

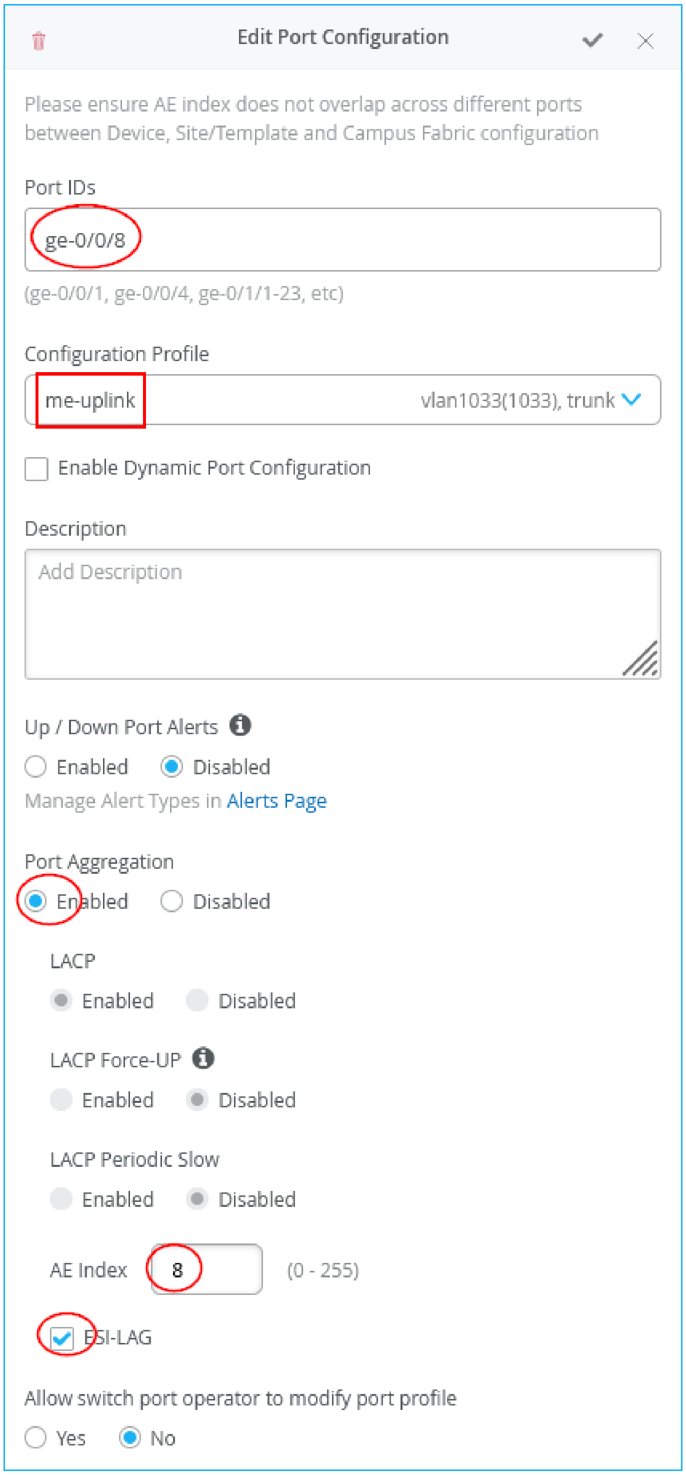
Allow switch port operator to modify port profile: ○ Yes ● No

The configuration of Core1-Switch and Core2-Switch for the fourth Juniper Mist Edge is:

- Port ID=`ge-0/0/8`

- Interface=`L2 interface`
- Configuration Profile=`me-uplink`
- Port Aggregation=`Enabled`
  - AE Index=`8` (you need to ensure this index value is unique and only used on the interface towards the same Juniper Mist Edge)
  - ESI-LAG=Enabled (enabling this is mandatory)

Figure 124: ESI-LAG for Mist-Edge Four



Edit Port Configuration

Please ensure AE index does not overlap across different ports between Device, Site/Template and Campus Fabric configuration

Port IDs

ge-0/0/8

(ge-0/0/1, ge-0/0/4, ge-0/1/1-23, etc)

Configuration Profile

me-uplink vlan1033(1033), trunk

□ Enable Dynamic Port Configuration

Description

Add Description

Up / Down Port Alerts

○ Enabled ● Disabled

Manage Alert Types in Alerts Page

Port Aggregation

● Enabled ○ Disabled

LACP

● Enabled ○ Disabled

LACP Force-UP

○ Enabled ● Disabled

LACP Periodic Slow

○ Enabled ● Disabled

AE Index 8 (0 - 255)

☑ ESI-LAG

Allow switch port operator to modify port profile

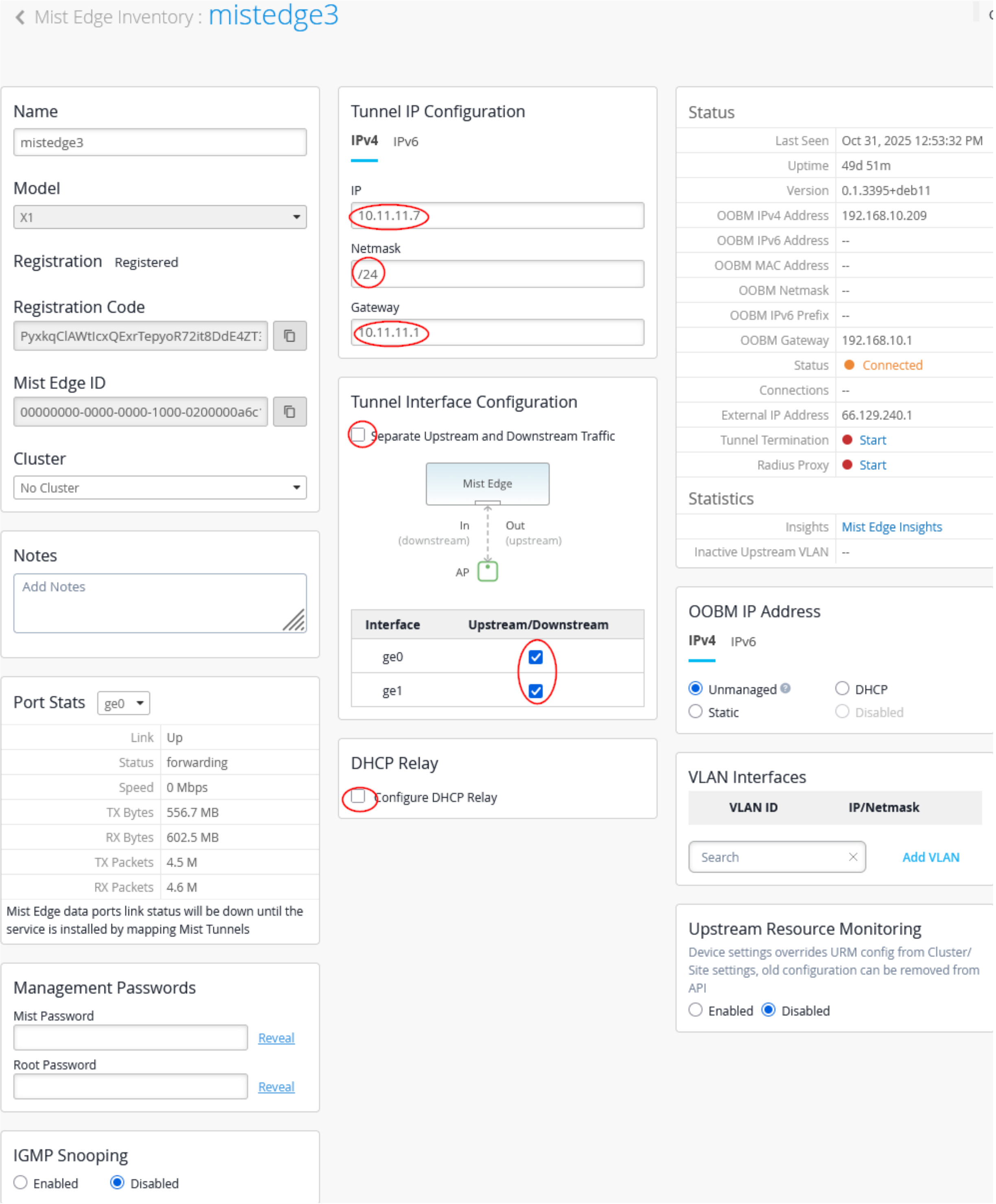
○ Yes ● No

Next, we configure the third Juniper Mist Edge as follows:

- Name=`mistedge3`

- IP=`10.11.11.7` (remember the AP VLAN should be different from the VLAN used for Juniper Mist Edges) (remember our APs are in the same VLAN1033 but their DHCP lease range starts at 10.33.33.10)
- Netmask=`/24`
- Gateway=`10.11.11.1`
- Separate Upstream and Downstream Traffic=`Unchecked`
- Interface ge0=`Checked`
- Interface ge1=`Checked`
- DHCP Relay=`Unchecked`

Figure 125: Mist-Edge Three Configuration



< Mist Edge Inventory : mistedge3

Name
mistedge3

Model
X1

Registration Registered

Registration Code
PyxkqClAWtIcxQExrTepyoR72it8DdE4ZT:

Mist Edge ID
00000000-0000-0000-1000-0200000a6c

Cluster
No Cluster

Notes
Add Notes

Port Stats ge0

Link	Up
Status	forwarding
Speed	0 Mbps
TX Bytes	556.7 MB
RX Bytes	602.5 MB
TX Packets	4.5 M
RX Packets	4.6 M

Mist Edge data ports link status will be down until the service is installed by mapping Mist Tunnels

Management Passwords

Mist Password Reveal

Root Password Reveal

IGMP Snooping
○ Enabled ● Disabled

Tunnel IP Configuration
IPv4 IPv6

IP
10.11.11.7

Netmask
/24

Gateway
10.11.11.1

Tunnel Interface Configuration
□ Separate Upstream and Downstream Traffic

Interface	Upstream/Downstream
ge0	☑
ge1	☑

DHCP Relay
□ Configure DHCP Relay

Status

Last Seen	Oct 31, 2025 12:53:32 PM
Uptime	49d 51m
Version	0.1.3395+deb11
OOBM IPv4 Address	192.168.10.209
OOBM IPv6 Address	--
OOBM MAC Address	--
OOBM Netmask	--
OOBM IPv6 Prefix	--
OOBM Gateway	192.168.10.1
Status	● Connected
Connections	--
External IP Address	66.129.240.1
Tunnel Termination	● Start
Radius Proxy	● Start

Statistics

Insights	Mist Edge Insights
Inactive Upstream VLAN	--

OOBM IP Address
IPv4 IPv6
● Unmanaged ○ DHCP
○ Static ○ Disabled

VLAN Interfaces

VLAN ID	IP/Netmask

Search × Add VLAN

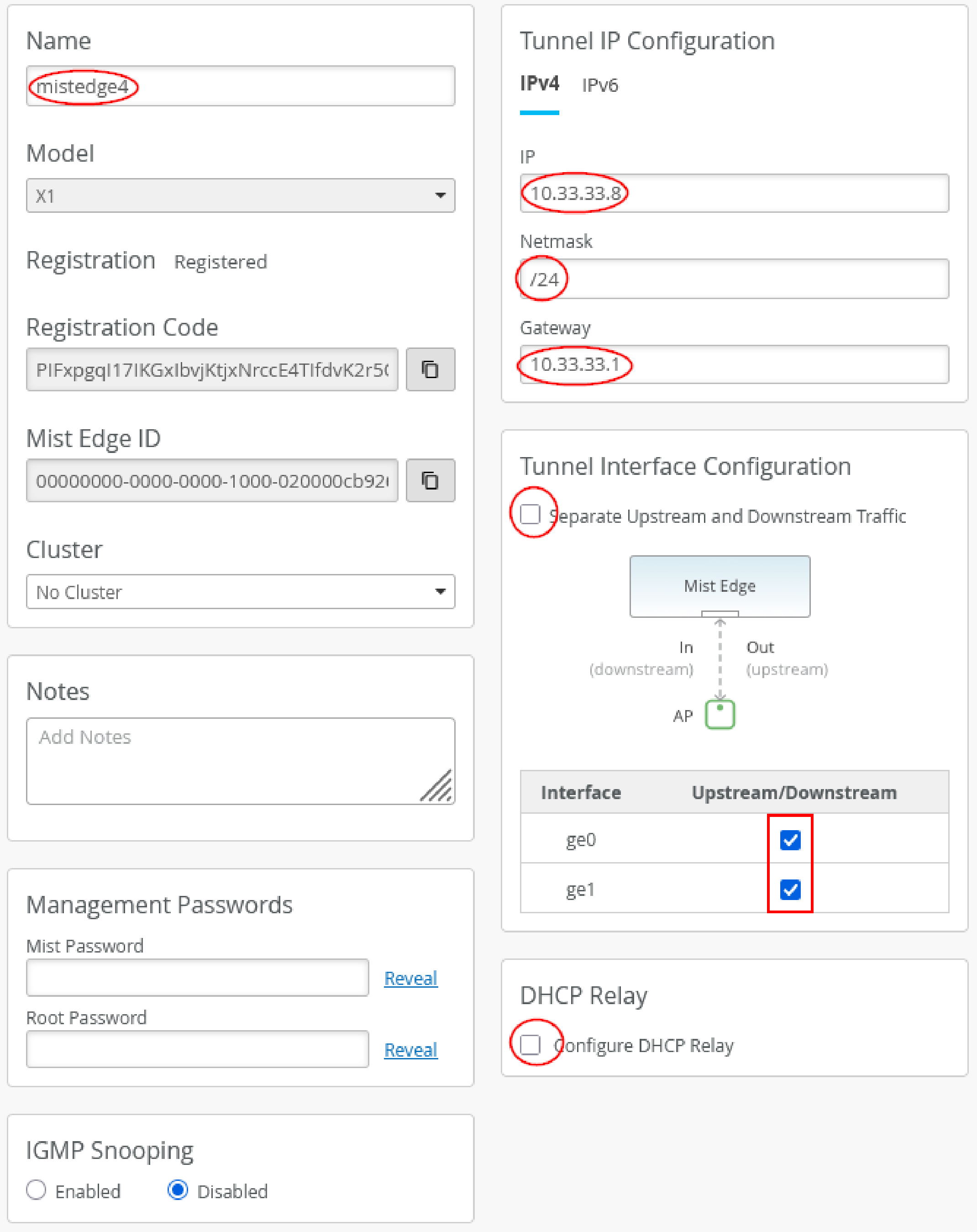
Upstream Resource Monitoring
Device settings overrides URM config from Cluster/ Site settings, old configuration can be removed from API
○ Enabled ● Disabled

Next, we configure the fourth Juniper Mist Edge as follows:

- Name=`mistedge4`

- IP=`10.11.11.8` (remember the AP VLAN should be different from the VLAN used for Juniper Mist Edges) (remember our APs are in the same VLAN1033 but their DHCP lease range starts at 10.33.33.10)
- Netmask=`/24`
- Gateway=`10.11.11.1`
- Separate Upstream and Downstream Traffic=`Unchecked`
- Interface ge0=`Checked`
- Interface ge1=`Checked`
- DHCP Relay=`Unchecked`

Figure 126: Mist-Edge Four Configuration



Name

mistedge4

Model

X1

Registration Registered

Registration Code

PIFxpgqI17IKGxIbvjKtjxNrccE4TIfdvK2r5(

Mist Edge ID

00000000-0000-0000-1000-020000cb92(

Cluster

No Cluster

Notes

Add Notes

Management Passwords

Mist Password

Reveal

Root Password

Reveal

IGMP Snooping

○ Enabled ● Disabled

Tunnel IP Configuration

IPv4 IPv6

IP

10.33.33.8

Netmask

/24

Gateway

10.33.33.1

Tunnel Interface Configuration

□ Separate Upstream and Downstream Traffic

Interface	Upstream/Downstream
ge0	☑
ge1	☑

DHCP Relay

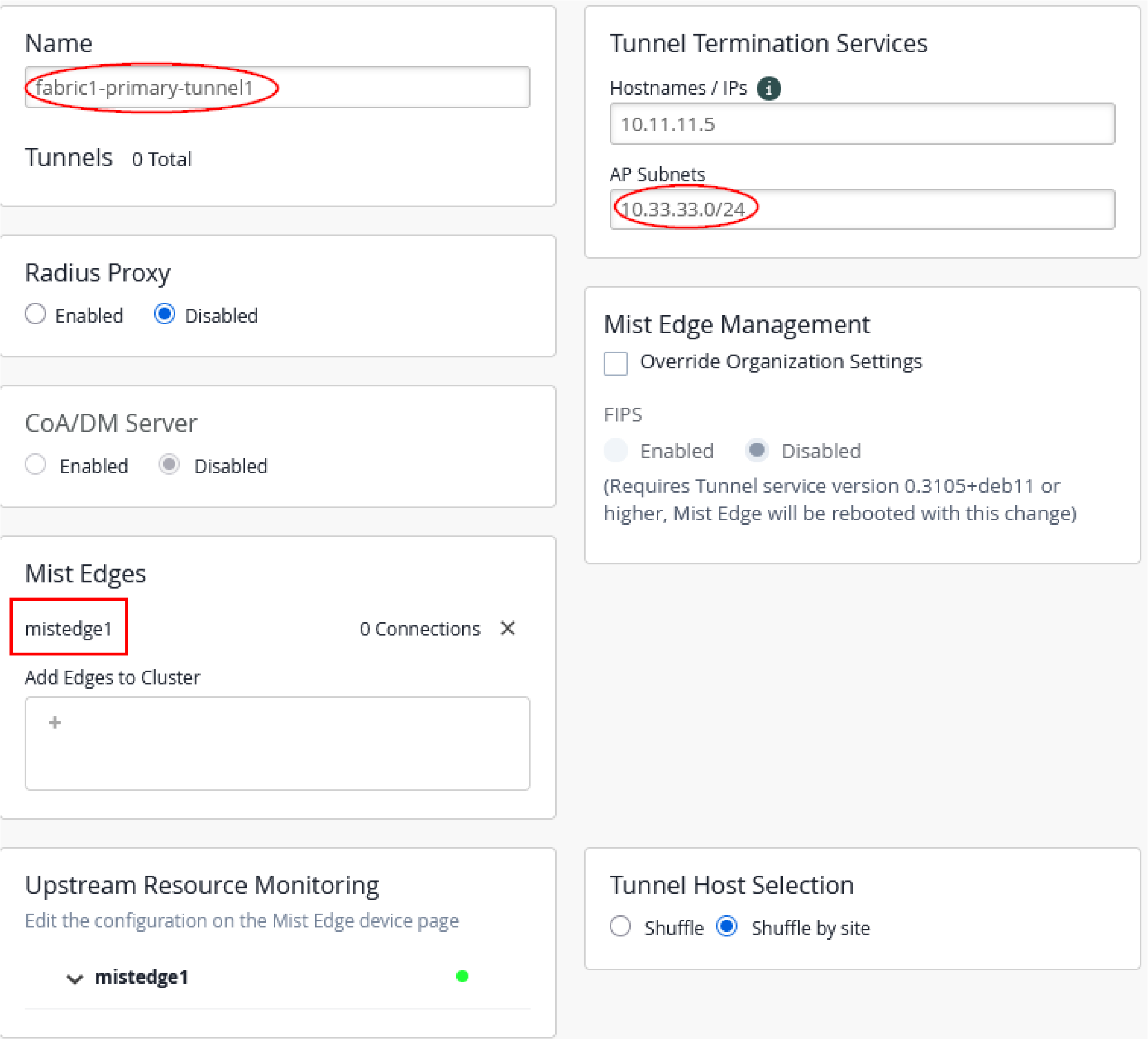
□ Configure DHCP Relay

NOTE: If you still have the configuration from the previous lab, first remove the tunnel assignments from the APs. Next, delete the tunnels themselves, followed by the site cluster configuration. Do not modify the existing Juniper Mist Edges, as their configurations can be reused without changes.

For the first Mist Edge cluster, configure the following:

- Name=`fabric1-primary-tunnel1`
- Mist Edges=`mistedge1`
- Hostnames / IPs=`10.11.11.5` (happens automatically when assigning Mist Edge)
- AP Subnets=`10.33.33.0/24` (This is an optional configuration. With it enabled, the Juniper Mist Edge will only accept tunnels if the AP's source IP address falls within the range assigned to VLAN 1033)
- Tunnel Host Selection=`Shuffle by Site` (This setting is required to configure all APs within the same site (fabric) to use the same Mist Edge)

Figure 127: First Mist-Edge Pair Primary Cluster



Name

fabric1-primary-tunnel1

Tunnels 0 Total

Radius Proxy

○ Enabled ● Disabled

CoA/DM Server

○ Enabled ● Disabled

Mist Edges

mistedge1 0 Connections ✕

Add Edges to Cluster

+

Upstream Resource Monitoring

Edit the configuration on the Mist Edge device page

mistedge1 ●

Tunnel Termination Services

Hostnames / IPs

10.11.11.5

AP Subnets

10.33.33.0/24

Mist Edge Management

□ Override Organization Settings

FIPS

○ Enabled ● Disabled

(Requires Tunnel service version 0.3105+deb11 or higher, Mist Edge will be rebooted with this change)

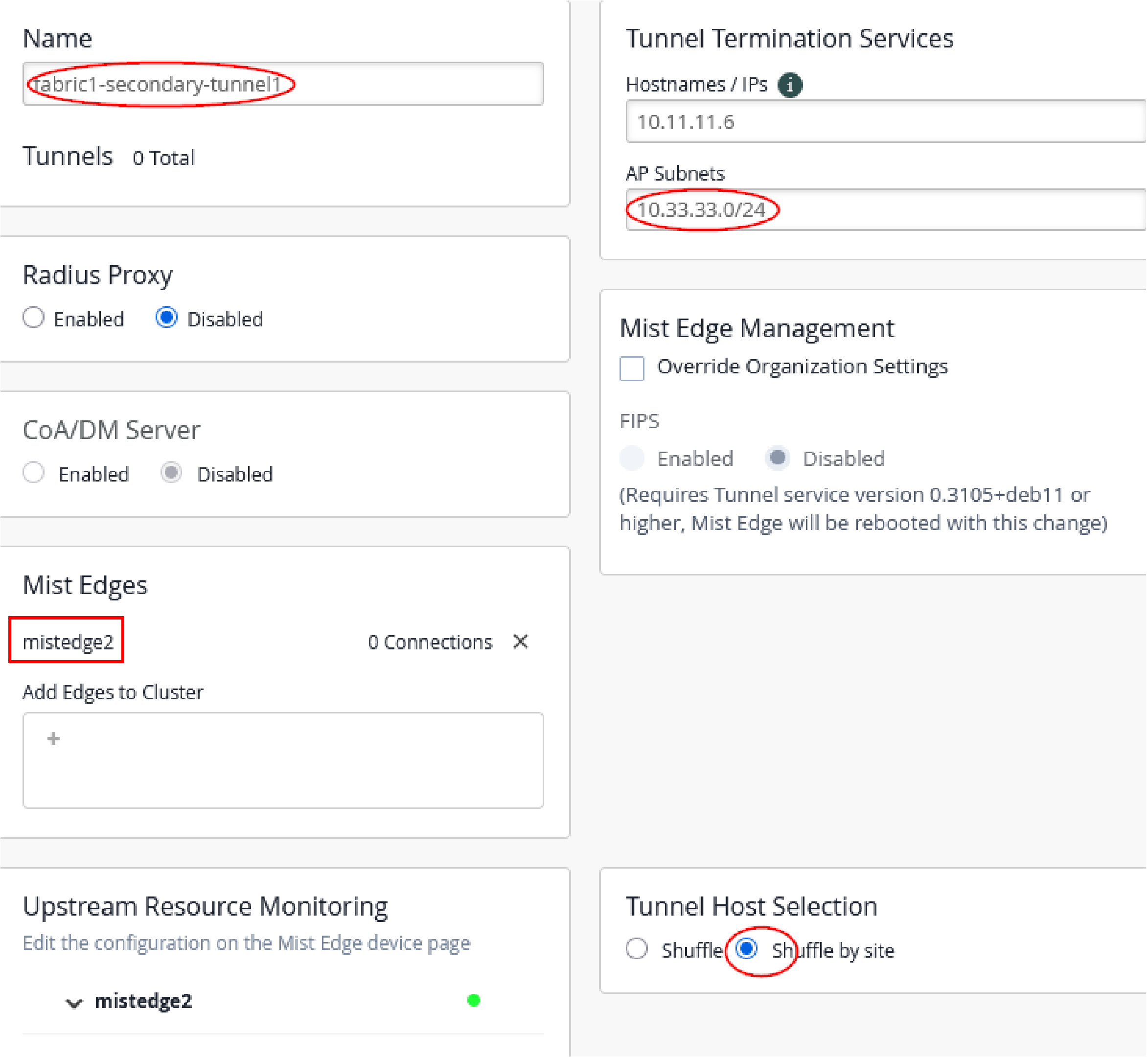
Tunnel Host Selection

○ Shuffle ● Shuffle by site

For the second Mist-Edge cluster configure:

- Name=`fabric1-secondary-tunnel1`
- Mist Edges=`mistedge2`
- Hostnames / IPs=`10.11.11.6` (happens automatic when assigning Mist Edge)
- AP Subnets=`10.33.33.0/24` this is an OPTIONAL configuration and Mist-Edge would then only accept Tunnels when the Source IP-Address of the AP is in range on vlan1033.
- Tunnel Host Selection=`Shuffle by Site` MANDATORY to tell all AP's in same site (fabric) use the same Mist-Edge.

Figure 128: First Mist-Edge Pair Secondary Cluster



Name

fabric1-secondary-tunnel1

Tunnels 0 Total

Radius Proxy

○ Enabled ● Disabled

CoA/DM Server

○ Enabled ● Disabled

Mist Edges

mistedge2 0 Connections ×

Add Edges to Cluster

+

Upstream Resource Monitoring

Edit the configuration on the Mist Edge device page

mistedge2 ●

Tunnel Termination Services

Hostnames / IPs

10.11.11.6

AP Subnets

10.33.33.0/24

Mist Edge Management

□ Override Organization Settings

FIPS

○ Enabled ● Disabled

(Requires Tunnel service version 0.3105+deb11 or higher, Mist Edge will be rebooted with this change)

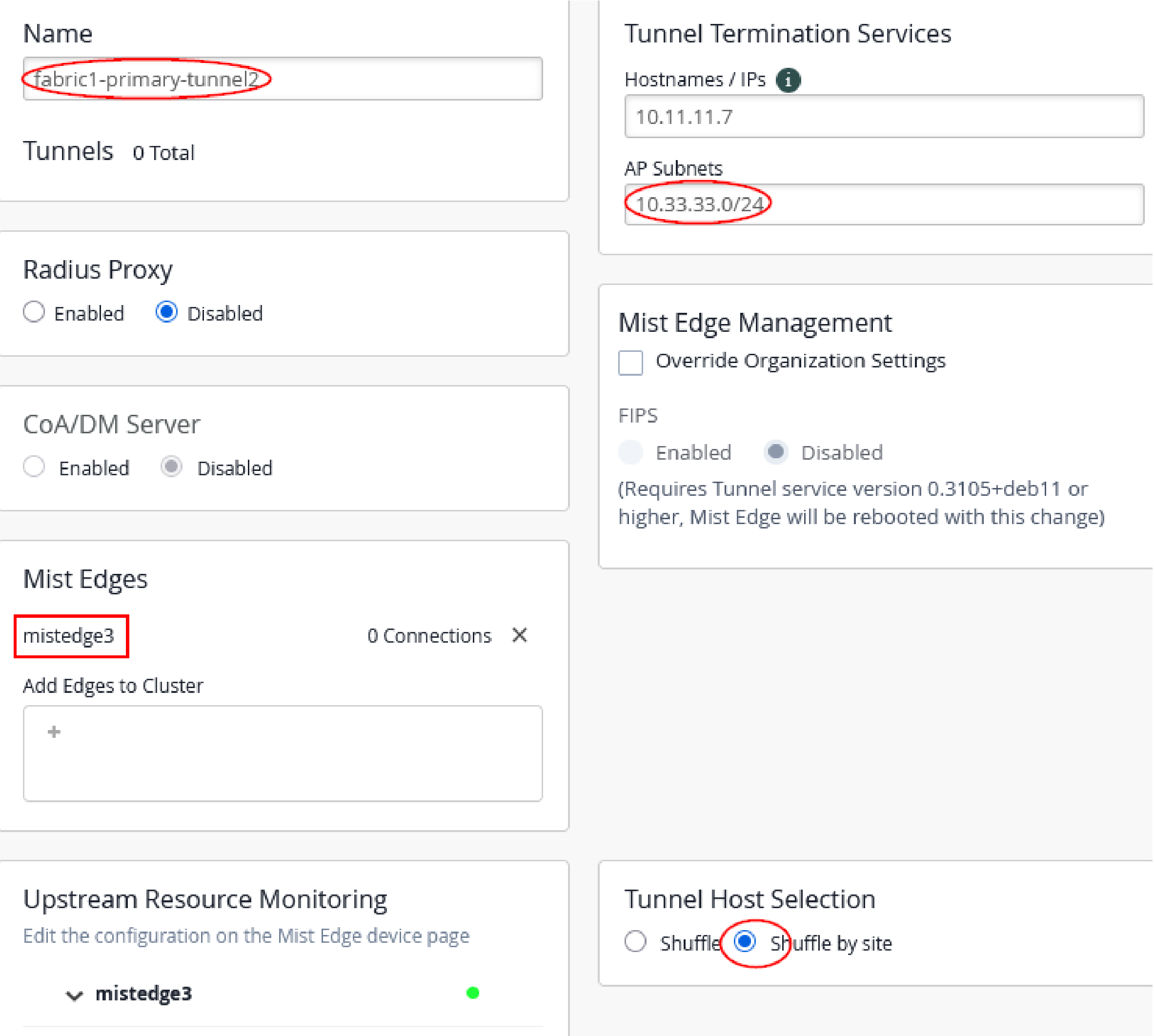
Tunnel Host Selection

○ Shuffle ● Shuffle by site

For the third Juniper Mist Edge cluster configure the following:

- Name=`fabric1-primary-tunnel2`
- Mist Edges=`mistedge3`
- Hostnames / IPs=`10.11.11.7` (happens automatically when assigning the Juniper Mist Edge)
- AP Subnets=`10.33.33.0/24` (This configuration is optional. With it enabled, the Juniper Mist Edge will only accept tunnels if the AP's source IP address falls within the range assigned to VLAN 1033)
- Tunnel Host Selection=`Shuffle by Site` (This configuration is required and ensures that all APs within the same site or fabric use the same Juniper Mist Edge)

Figure 129: Second Mist-Edge Pair Primary Cluster



Name

fabric1-primary-tunnel2

Tunnels 0 Total

Radius Proxy

○ Enabled ● Disabled

CoA/DM Server

○ Enabled ● Disabled

Mist Edges

mistedge3 0 Connections ✕

Add Edges to Cluster

+

Upstream Resource Monitoring

Edit the configuration on the Mist Edge device page

mistedge3 ●

Tunnel Termination Services

Hostnames / IPs

10.11.11.7

AP Subnets

10.33.33.0/24

Mist Edge Management

□ Override Organization Settings

FIPS

○ Enabled ● Disabled

(Requires Tunnel service version 0.3105+deb11 or higher, Mist Edge will be rebooted with this change)

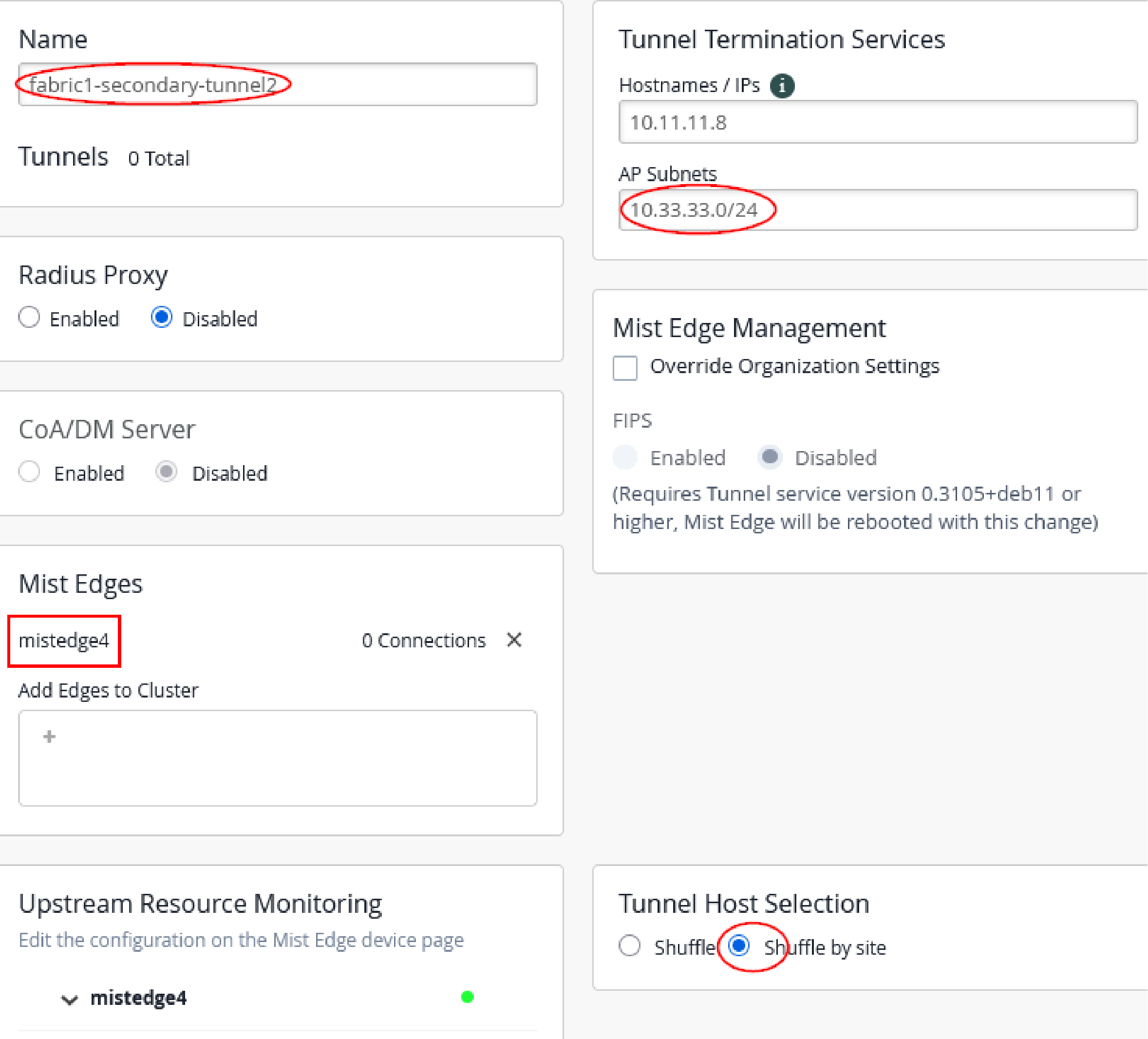
Tunnel Host Selection

○ Shuffle ● Shuffle by site

For the fourth Juniper Mist Edge cluster configure the following:

- Name=`fabric1-secondary-tunnel2`
- Mist Edges=`mistedge4`
- Hostnames / IPs=`10.11.11.8` (happens automatically when assigning the Juniper Mist Edge)
- AP Subnets=`10.33.33.0/24` (This configuration is optional. With it enabled, the Juniper Mist Edge will only accept tunnels if the AP's source IP address falls within the range assigned to VLAN 1033)
- Tunnel Host Selection=`Shuffle by Site` (This configuration is required and ensures that all APs within the same site or fabric use the same Juniper Mist Edge)

Figure 130: Second Mist-Edge Pair Secondary Cluster



Name

fabric1-secondary-tunnel2

Tunnels 0 Total

Radius Proxy

○ Enabled ● Disabled

CoA/DM Server

○ Enabled ● Disabled

Mist Edges

mistedge4 0 Connections ✕

Add Edges to Cluster

Upstream Resource Monitoring

Edit the configuration on the Mist Edge device page

mistedge4 ●

Tunnel Termination Services

Hostnames / IPs

10.11.11.8

AP Subnets

10.33.33.0/24

Mist Edge Management

□ Override Organization Settings

FIPS

○ Enabled ● Disabled

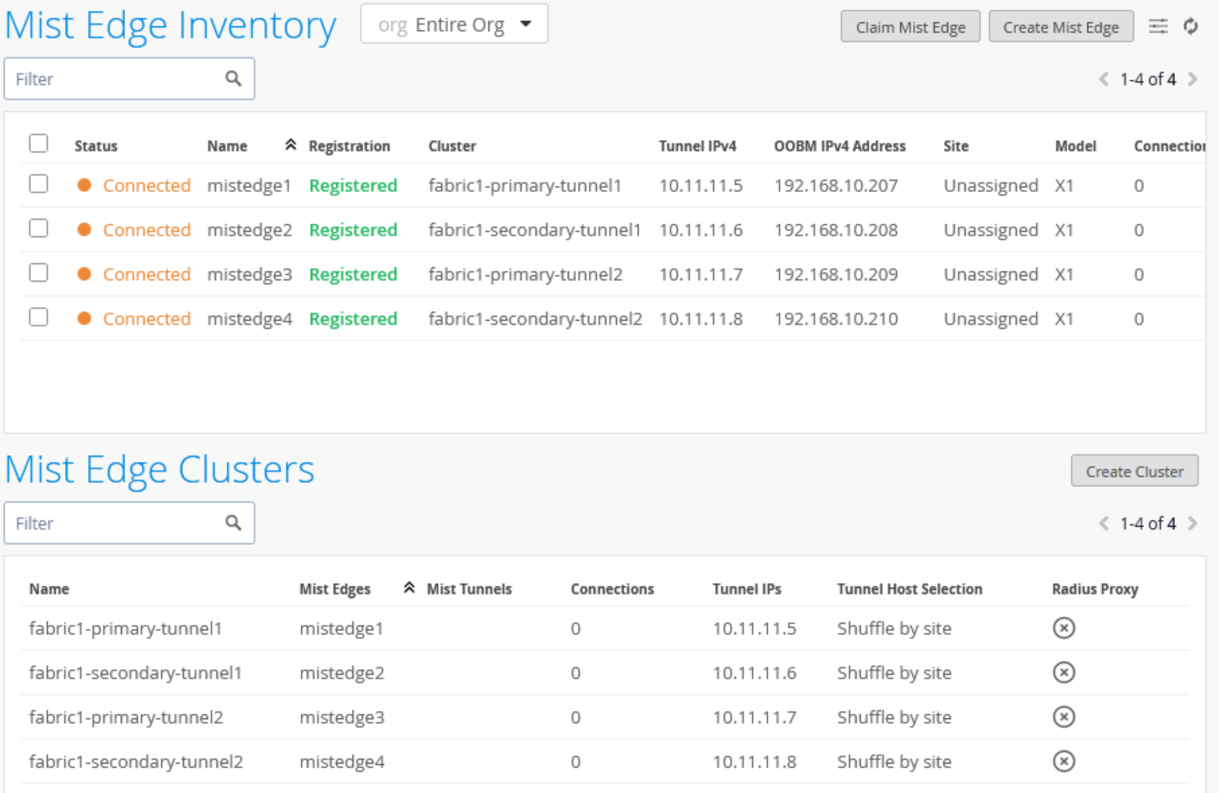
(Requires Tunnel service version 0.3105+deb11 or higher, Mist Edge will be rebooted with this change)

Tunnel Host Selection

○ Shuffle ● Shuffle by site

After making your changes, the configuration should look like the image shown below:

Figure 131: New Mist-Edges and Clusters Added



□	Status	Name	Registration	Cluster	Tunnel IPv4	OOBM IPv4 Address	Site	Model	Connectio
□	● Connected	mistedge1	Registered	fabric1-primary-tunnel1	10.11.11.5	192.168.10.207	Unassigned	X1	0
□	● Connected	mistedge2	Registered	fabric1-secondary-tunnel1	10.11.11.6	192.168.10.208	Unassigned	X1	0
□	● Connected	mistedge3	Registered	fabric1-primary-tunnel2	10.11.11.7	192.168.10.209	Unassigned	X1	0
□	● Connected	mistedge4	Registered	fabric1-secondary-tunnel2	10.11.11.8	192.168.10.210	Unassigned	X1	0

Name	Mist Edges	Mist Tunnels	Connections	Tunnel IPs	Tunnel Host Selection	Radius Proxy
fabric1-primary-tunnel1	mistedge1		0	10.11.11.5	Shuffle by site	⊗
fabric1-secondary-tunnel1	mistedge2		0	10.11.11.6	Shuffle by site	⊗
fabric1-primary-tunnel2	mistedge3		0	10.11.11.7	Shuffle by site	⊗
fabric1-secondary-tunnel2	mistedge4		0	10.11.11.8	Shuffle by site	⊗

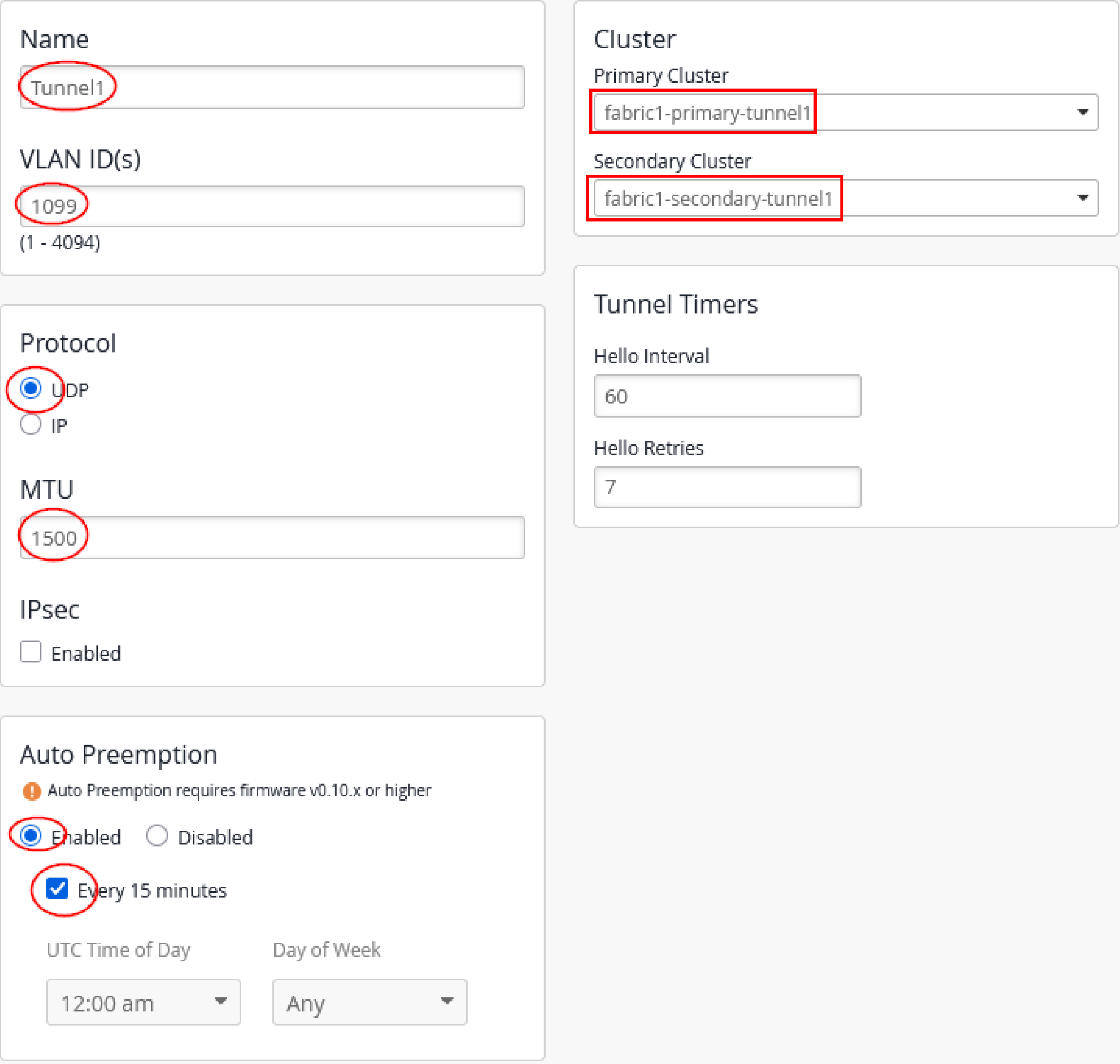
NOTE: We are keeping the shuffle-by-site option here even if it is technically not a hard requirement as we have always only one Mist Edge per primary/secondary cluster. Using the shuffle-by-site option should be always used in all EVPN Fabric designs as good practice.

Create the first Overlay-Tunnel from Access Point to Mist-Edge (in our case in vlan1033) configuration:

- Name=`Tunnel1`
- VLAN ID=`1099`
- Primary Cluster=`fabric1-primary-tunnel1`
- Secondary Cluster=`fabric1-secondary-tunnel1`
- Protocol=`UDP`
- MTU=`1500`
- IPsec=`Unchecked`

- Auto Preemption=`Enabled` (If an AP accidentally roams to the second Juniper Mist Edge while the primary Edge is still active and managing the other APs, the AP will automatically roam back to the primary)
  - Every 15 minutes=`Checked` (roaming back should happen as fast as possible, hence this configuration)

Figure 132: First Tunnel Using First Mist Edge Cluster Pair



Name

Tunnel1

VLAN ID(s)

1099

(1 - 4094)

Protocol

● UDP

○ IP

MTU

1500

IPsec

□ Enabled

Auto Preemption

Auto Preemption requires firmware v0.10.x or higher

● Enabled ○ Disabled

☑ Every 15 minutes

UTC Time of Day: 12:00 am

Day of Week: Any

Cluster

Primary Cluster: fabric1-primary-tunnel1

Secondary Cluster: fabric1-secondary-tunnel1

Tunnel Timers

Hello Interval: 60

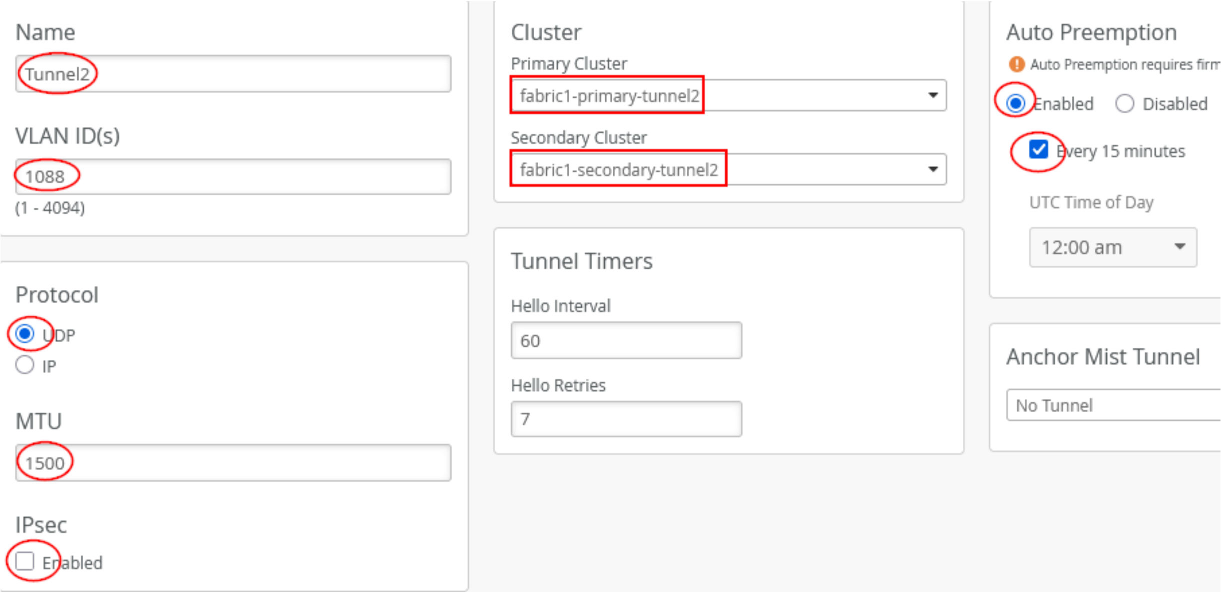
Hello Retries: 7

Next, add the configuration for the second overlay tunnel from AP to Juniper Mist Edge (in our case, in vlan1033):

- Name=`Tunnel2`
- VLAN ID=`1088`

- Primary Cluster=`fabric1-primary-tunnel2`
- Secondary Cluster=`fabric1-secondary-tunnel2`
- Protocol=`UDP`
- MTU=`1500`
- IPsec=`Unchecked`
- Auto Preemption=`Enabled` (If an AP accidentally roams to the second Juniper Mist Edge while the primary Edge is still active and managing the other APs, the AP will automatically roam back to the primary)
  - Every 15 minutes=`Checked` (roaming back should happen as fast as possible, hence this configuration)

Figure 133: Second Tunnel Using Second Mist-Edge Cluster Pair



Name
Tunnel2

VLAN ID(s)
1088
(1 - 4094)

Protocol
● UDP
○ IP

MTU
1500

IPsec
□ Enabled

Cluster
Primary Cluster
fabric1-primary-tunnel2
Secondary Cluster
fabric1-secondary-tunnel2

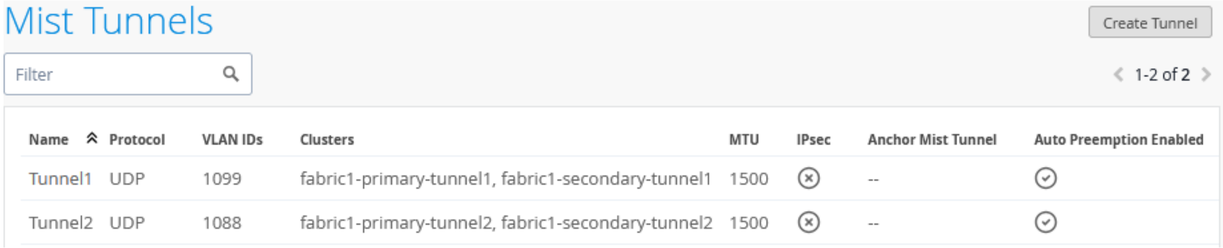
Tunnel Timers
Hello Interval
60
Hello Retries
7

Auto Preemption
Auto Preemption requires firm
● Enabled ○ Disabled
☑ Every 15 minutes
UTC Time of Day
12:00 am

Anchor Mist Tunnel
No Tunnel

Your changes should look like the image shown below:

Figure 134: Two Tunnels with Distributed VLANs and Mist-Edge Clusters

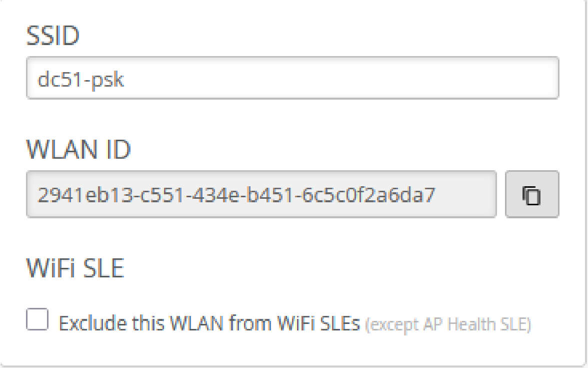


Mist Tunnels

Name	Protocol	VLAN IDs	Clusters	MTU	IPsec	Anchor Mist Tunnel	Auto Preemption Enabled
Tunnel1	UDP	1099	fabric1-primary-tunnel1, fabric1-secondary-tunnel1	1500	⊗	--	✓
Tunnel2	UDP	1088	fabric1-primary-tunnel2, fabric1-secondary-tunnel2	1500	⊗	--	✓

Check the configuration of the first SSID:

Figure 135: First SSID



SSID

dc51-psk

WLAN ID

2941eb13-c551-434e-b451-6c5c0f2a6da7

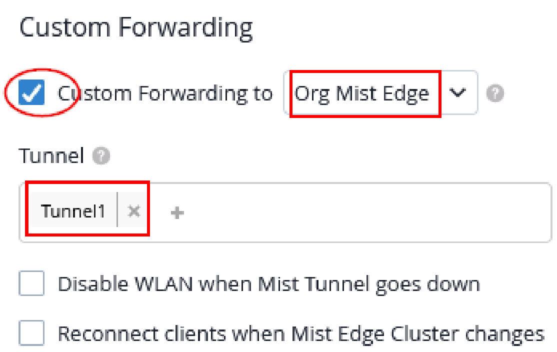
WiFi SLE

☐ Exclude this WLAN from WiFi SLEs (except AP Health SLE)

As we used VLAN 1099 for Tunnel1 and need to create the Custom Forwarding now configuring:

- Custom Forwarding=`Checked/Enabled`
  - To=`Org Mist Edge`
- Tunnel=`Tunnel1`

Figure 136: First SSID Uses First Tunnel



Custom Forwarding

☑ Custom Forwarding to Org Mist Edge

Tunnel

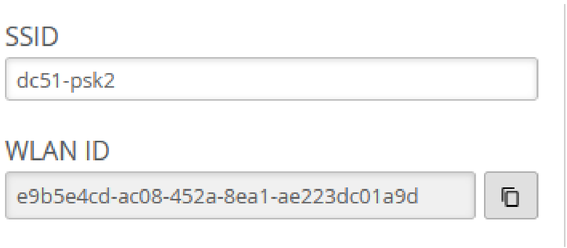
Tunnel1

□ Disable WLAN when Mist Tunnel goes down

□ Reconnect clients when Mist Edge Cluster changes

Check the configuration of the second SSID:

Figure 137: Second SSID



SSID

dc51-psk2

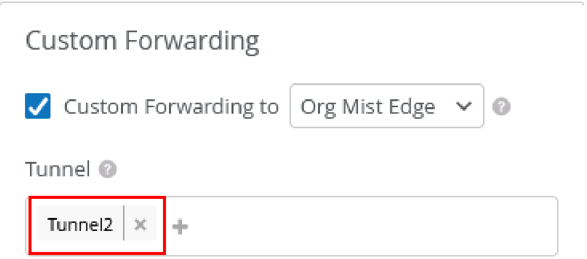
WLAN ID

e9b5e4cd-ac08-452a-8ea1-ae223dc01a9d

As we used VLAN 1088 for Tunnel2 and need to create the Custom Forwarding now configuring:

- Custom Forwarding=`Checked/Enabled`
  - To=`Org Mist Edge`
- Tunnel=`Tunnel2`

Figure 138: Second SSID Uses Second Tunnel



Custom Forwarding

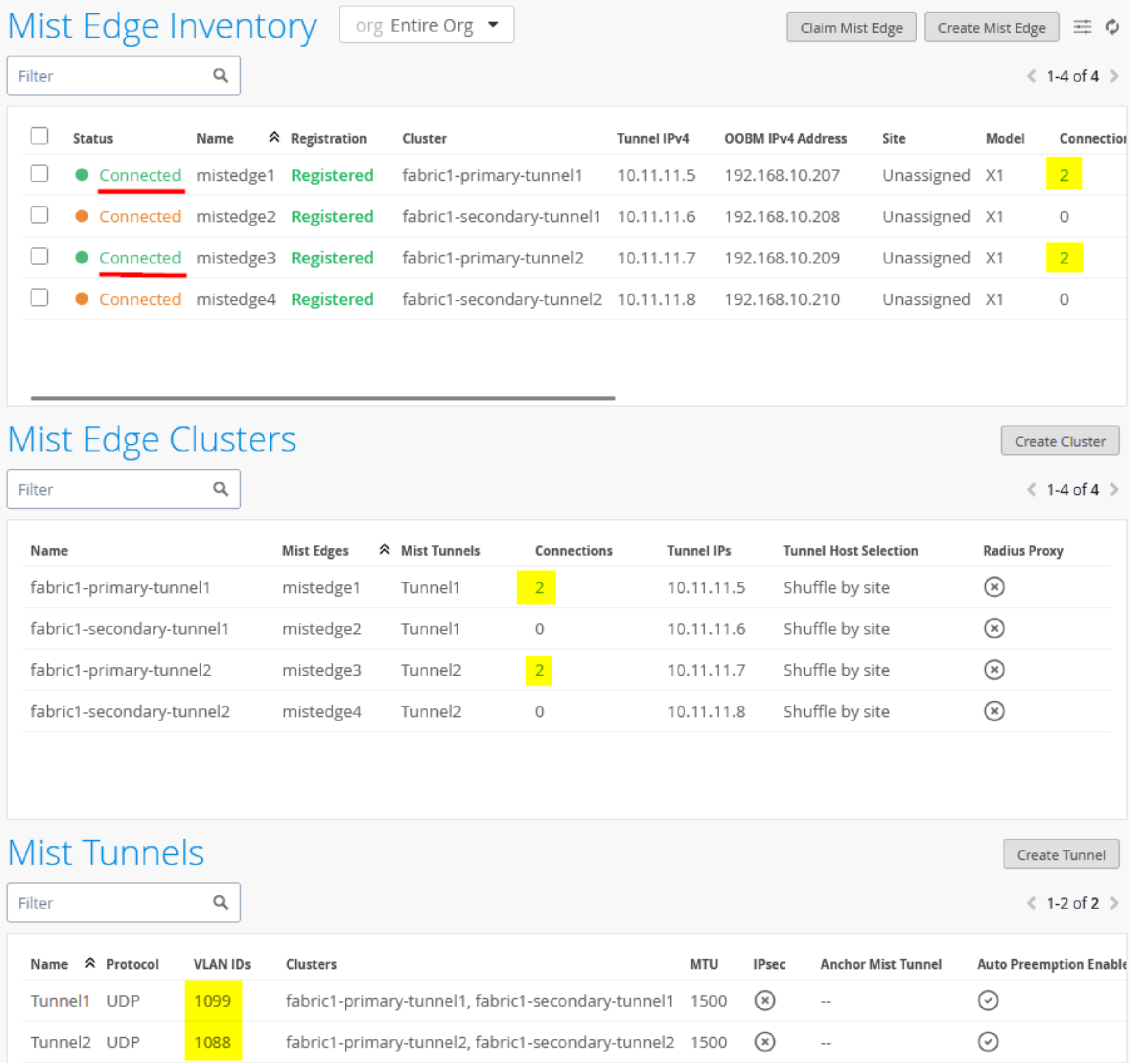
☑ Custom Forwarding to Org Mist Edge

Tunnel

Tunnel2

After making this adjustment, each AP, with both SSIDs configured, will establish two tunnels to the designated primary cluster as shown below:

Figure 139: Mist-Edge Tunnel Scale Distribution



Mist Edge Inventory

□	Status	Name	Registration	Cluster	Tunnel IPv4	OOBM IPv4 Address	Site	Model	Connectio
□	● Connected	mistedge1	Registered	fabric1-primary-tunnel1	10.11.11.5	192.168.10.207	Unassigned	X1	2
□	● Connected	mistedge2	Registered	fabric1-secondary-tunnel1	10.11.11.6	192.168.10.208	Unassigned	X1	0
□	● Connected	mistedge3	Registered	fabric1-primary-tunnel2	10.11.11.7	192.168.10.209	Unassigned	X1	2
□	● Connected	mistedge4	Registered	fabric1-secondary-tunnel2	10.11.11.8	192.168.10.210	Unassigned	X1	0

Mist Edge Clusters

Name	Mist Edges	Mist Tunnels	Connections	Tunnel IPs	Tunnel Host Selection	Radius Proxy
fabric1-primary-tunnel1	mistedge1	Tunnel1	2	10.11.11.5	Shuffle by site	⊗
fabric1-secondary-tunnel1	mistedge2	Tunnel1	0	10.11.11.6	Shuffle by site	⊗
fabric1-primary-tunnel2	mistedge3	Tunnel2	2	10.11.11.7	Shuffle by site	⊗
fabric1-secondary-tunnel2	mistedge4	Tunnel2	0	10.11.11.8	Shuffle by site	⊗

Mist Tunnels

Name	Protocol	VLAN IDs	Clusters	MTU	IPsec	Anchor Mist Tunnel	Auto Preemption Enable
Tunnel1	UDP	1099	fabric1-primary-tunnel1, fabric1-secondary-tunnel1	1500	⊗	--	✓
Tunnel2	UDP	1088	fabric1-primary-tunnel2, fabric1-secondary-tunnel2	1500	⊗	--	✓

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners. Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice. Copyright © 2026 Juniper Networks, Inc. All rights reserved.