

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

Published
2024-04-05

Table of Contents

About this Document	1
Overview	1
Benefits of Campus Fabric IP Clos	2
Technical Overview	4
Campus Fabric IP Clos Deployment Types	13
Juniper Mist Wired Assurance	16
Campus Fabric IP Clos High-Level Architecture	17
Campus Fabric IP Clos Components	18
Juniper Mist Wired Assurance	19
Juniper Mist Wired Assurance Switches	20
Create the Campus Fabric	29
Verification	51
EVPN Insights	65
Summary	67
Additional Information	68

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

Juniper Networks Validated Designs provide customers with a comprehensive, end-to-end blueprint for deploying Juniper solutions in their network. These designs are created by Juniper's expert engineers and tested to ensure they meet the customer's requirements. Using a validated design, customers can reduce the risk of costly mistakes, save time and money, and ensure that their 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 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 Mist Access Points integration.

Overview

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 (EVPN-VXLAN with 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. 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 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.

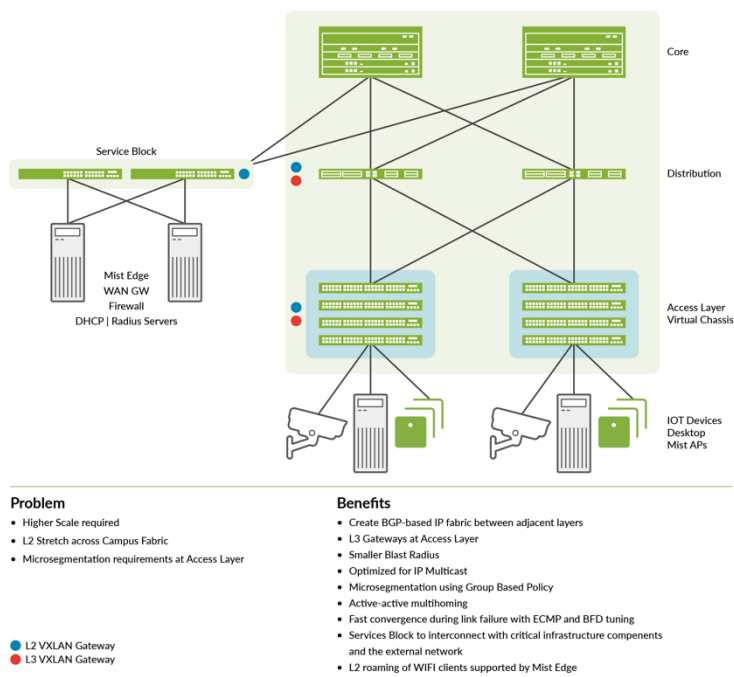
With an EVPN-VXLAN campus architecture, you can easily add core, distribution, and access layer devices as your business grows without a need for redesigning your network. As EVPN-VXLAN is vendor-agnostic, you can use the existing access layer infrastructure and gradually migrate to access layer switches. This supports EVPN-VXLAN capabilities once the core and distribution part of the network is deployed. Connectivity with legacy switches that do not support EVPN VXLAN is accomplished with standards-based ESI-LAG. ESI-LAG uses standards-based Link Aggregation Control Protocol (LACP) to interconnect with legacy switches.

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 multi-fold 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 endpoint (VTEP) that encapsulate and decapsulate the VXLAN traffic. VTEP represents 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



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 group-based policy (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. VXLAN overlay provides L2 extension across campuses without any changes to the underlay network. Juniper uses optimal BGP timers between the adjacent layers of the campus fabric with Bidirectional Forwarding Detection (BFD) that supports fast convergence in 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 Deployment Types | 13](#)

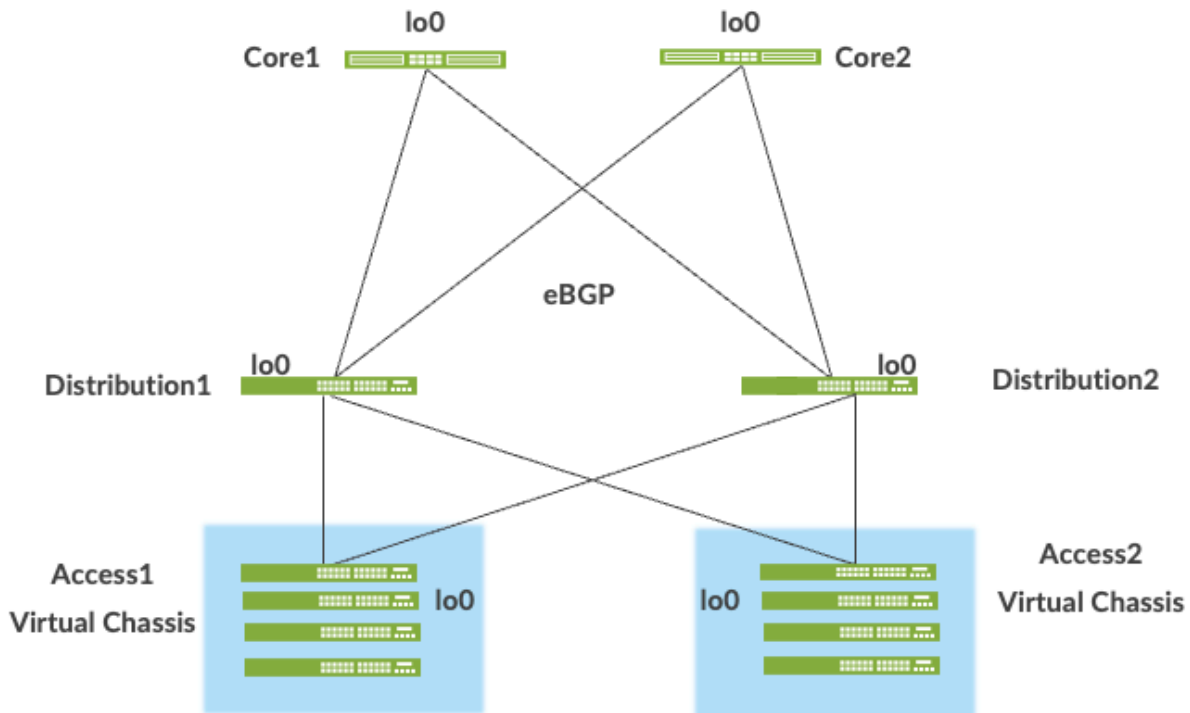
Underlay Network

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

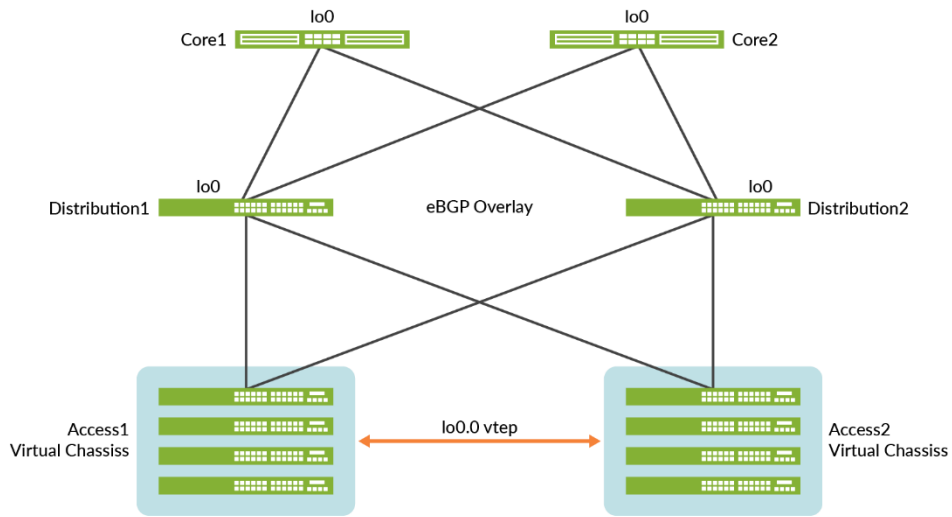


Network overlays enable connectivity and addressing independent of the physical network. Ethernet frames are wrapped in UDP/IP datagrams, which are encapsulated into IP for transport over the underlay. VXLAN enables virtual L2 subnets or VLANs to span underlying physical L3 network.

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 services 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 a L3 gateway. The entity that performs VXLAN encapsulation and decapsulation is called a VXLAN tunnel endpoint (VTEP). Each VTEP is known as the L2 gateway and 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 a 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

Ethernet VPN (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:

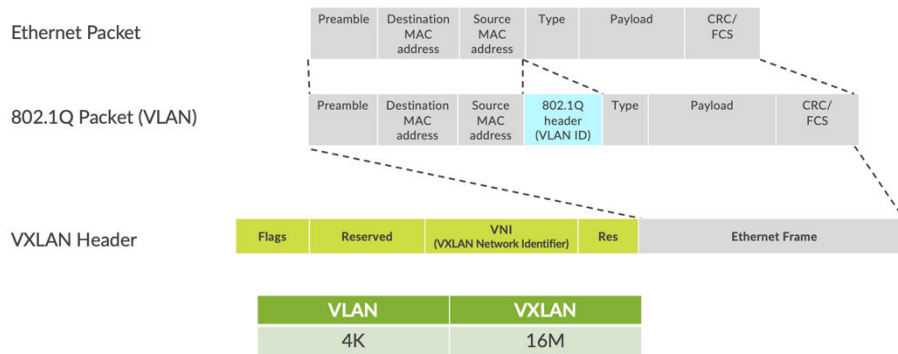
- Minimal flooding—EVPN creates a control plane that shares end-host MAC addresses between VTEPs.
- Multihoming—EVPN supports multihoming for client devices. A control protocol like EVPN that enables synchronization of endpoint addresses between the access switches is needed to support multihoming because 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 broadcast, unknown unicast, and multicast (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. Devices that perform VXLAN encapsulation and decapsulation for the network are referred to as 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



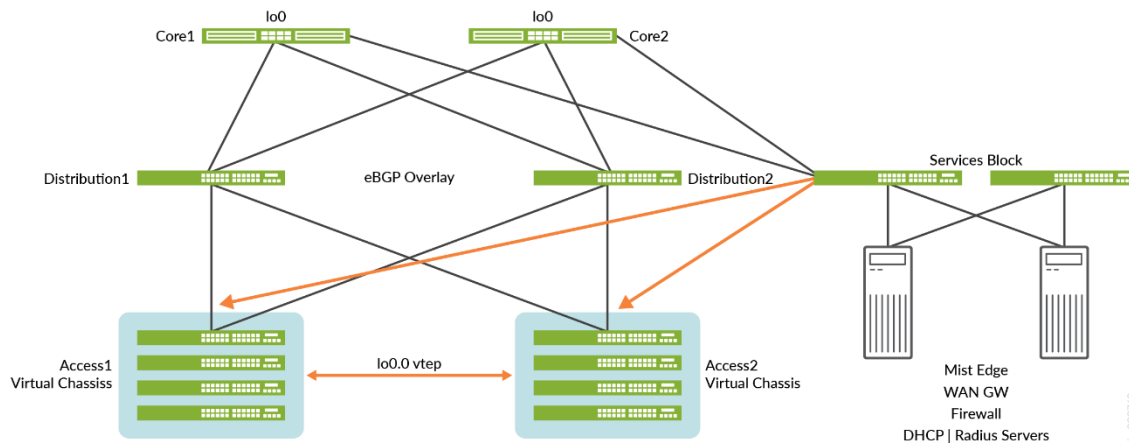
VTEPs are software entities tied to the devices' loopback address that source and terminate VXLAN tunnels. VXLAN tunnels in an IP Clos fabric are provisioned on the following:

- 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.
- Services block devices that interconnect the campus fabric with the outside network.

Overlay Network (Control Plane)

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 Services Block



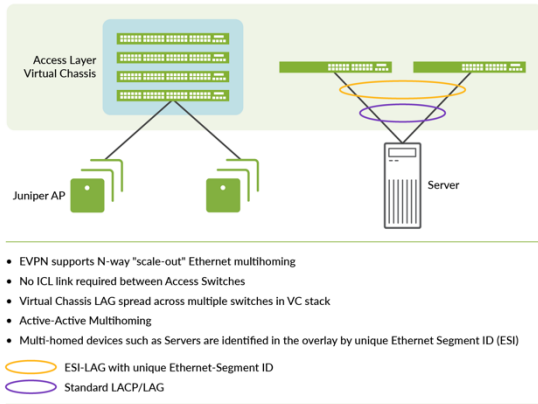
Resiliency and Load Balancing

We support Bi-Directional Forwarding (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. Mist configured BFD minimum intervals of 350ms and 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. ESI-LAG enables link failover in the event of a bad link, supports active-active load balancing, and is automatically assigned by Mist.

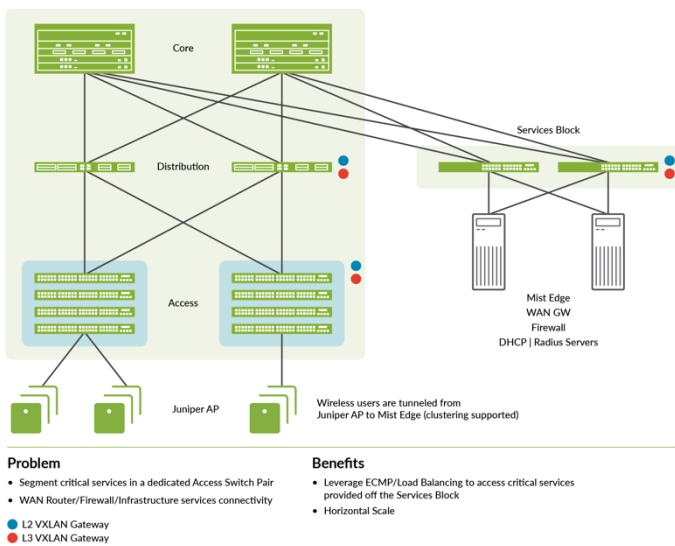
Figure 6: Device Resiliency and Load Balancing



Services Block

You need to position critical infrastructure services off a dedicated access pair of Juniper switches. This can include WAN and firewall connectivity, RADIUS, and DHCP servers for example. If you need to deploy a lean core, the dedicated services 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 services block border capability is supported directly off the core layer or as a dedicated pair of switches.

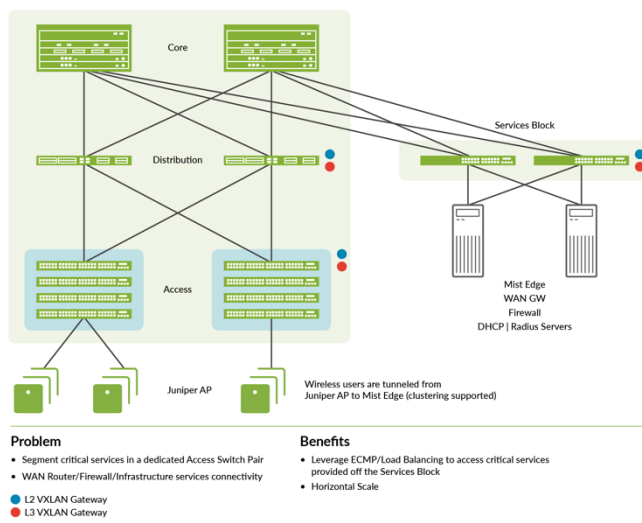
Figure 7: Services Block



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



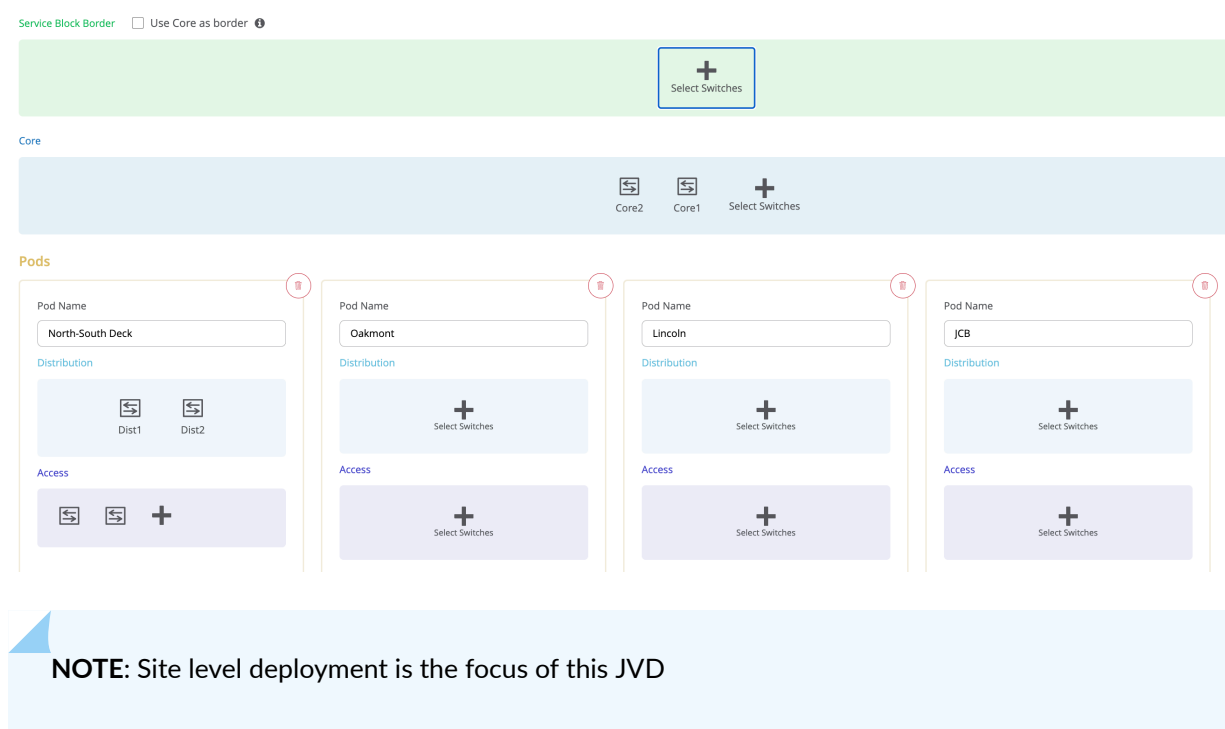
In this example, each access switch or Virtual Chassis is multihomed to two or more distribution switches. Juniper's Virtual Chassis reduces the number of ports required on distribution switches and optimizes the availability of fiber throughout the campus. The Virtual Chassis supports up to 10 member 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.

Campus Fabric Organizational Deployment

Mist campus fabric supports deployments at the Site and Organization level. The Organization-based deployment shown in [Figure 9 on page 12](#), targets enterprises who need to align with a POD structure.

Figure 9: Campus Fabric Node Configuration



Juniper Access Points

In our network, we choose Mist 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. Mist delivers unique capabilities for both wired and wireless LAN:

- Wired and wireless assurance—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 Mist platform. This JVD uses Mist wired assurance services.
- Marvis—An integrated AI engine that provides rapid wired and wireless troubleshooting, trending analysis, anomaly detection, and proactive problem remediation.

Mist Edge

For large campus networks, 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 Mist platform in this JVD addresses both full-stack deployments and automation. For more details on Mist integration with EX switches, see: [How to Connect Mist Access Points and Juniper EX Series Switches](#).

Campus Fabric IP Clos Deployment Types

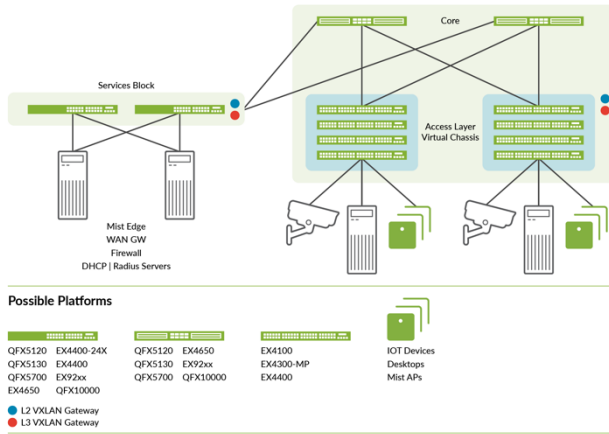
IN THIS SECTION

- [3-Stage IP Clos | 14](#)
- [5-Stage IP Clos | 14](#)
- [Supported Platforms for Campus Fabric IP Clos | 15](#)

Juniper's Wired Assurance supports 3-stage and 5-stage IP Clos deployments. The 3-stage IP Clos is targeted for deployments that do not require a distribution layer and have smaller-scale requirements. This allows for cost-effective EX4400, EX4650, QFX5110, and QFX5120 switching platforms to be deployed at the core layer.

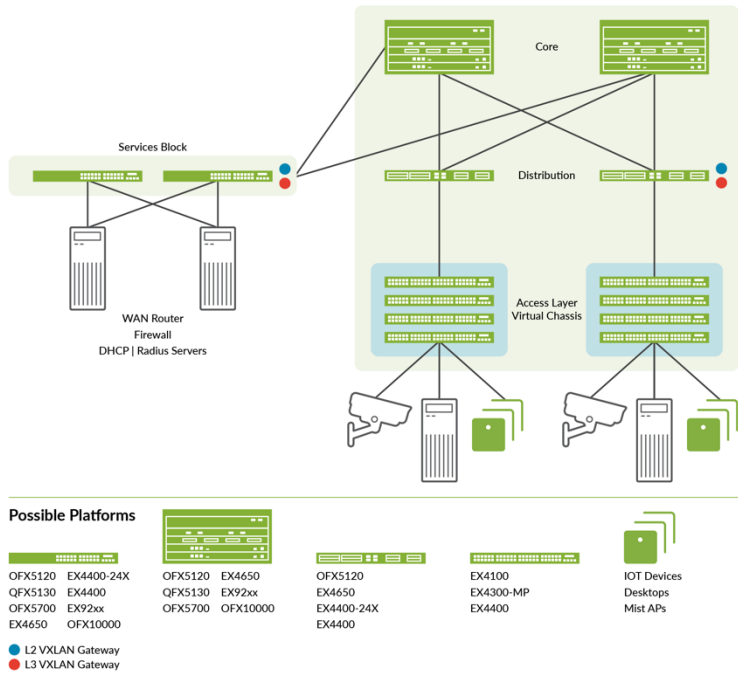
3-Stage IP Clos

Figure 10: 3-Stage IP Clos



5-Stage IP Clos

Figure 11: 5-Stage IP Clos



NOTE: You can deploy the services block in 3-Stage and 5-Stage IP Clos architectures.

Supported Platforms for Campus Fabric IP Clos

Table 1 on page 15 lists the supported platforms for Campus Fabric IP Clos deployment.

Table 1: Supported Platforms for Campus Fabric IP Clos Deployment

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

Table 1: Supported Platforms for Campus Fabric IP Clos Deployment (Continued)

Supported Platforms	
Campus Fabric IP Clos Deployment	Supported Platforms
Services block	EX4400-24X
	EX4650
	QFX5120
	QFX5130
	QFX5700
	EX92xx

NOTE: A hardware limitation on the EX4300-MP Switches does not allow it to be used for VXLAN Group Based Policies (GBP/SGT). Consider the Juniper EX4100/4400 Series 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 EX-Series switches provide streaming telemetry that enables insights for switch health metrics, anomaly detection, and Juniper Mist AI capabilities.

Mist's AI engine and virtual network assistant, Marvis, further simplify 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 Information Technology (IT) operations from reactive troubleshooting to proactive remediation.

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

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

- User Interface (UI) or API-driven configuration at scale.
- Service Level Expectations (SLE) 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.
- 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>

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. In a campus fabric deployment, the use of EVPN VXLAN supports native traffic isolation using software-based routing instances called Virtual Routing and Forwarding (VRFs) for macrosegmentation purposes.

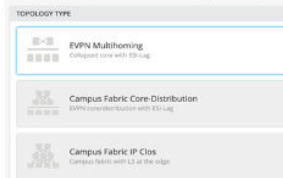
The Mist UI workflow makes it easy to create campus fabrics.

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

Choose EVPN Topology

Choose the topology you want to construct and configure related options



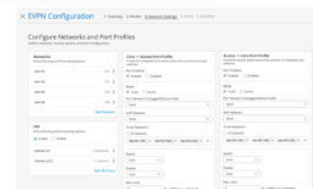
Apply the intent

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



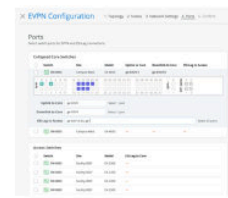
Define Networks of Interest

- Configure the user networks



Define Physical Connections

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



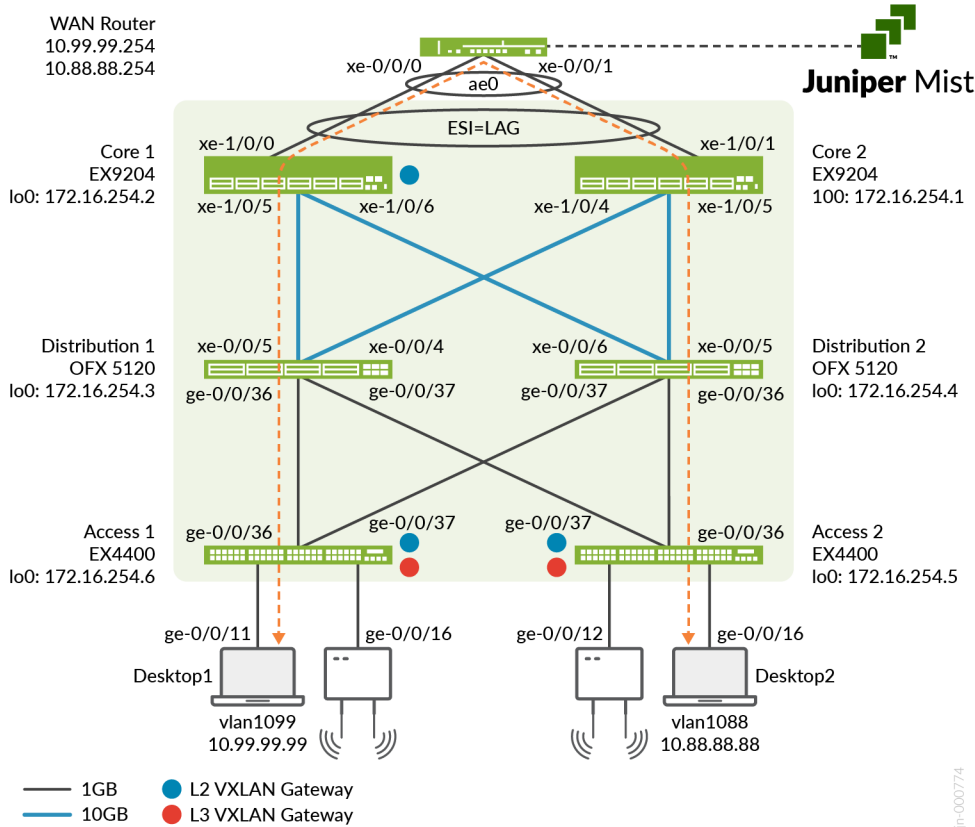
Campus Fabric IP Clos Components

This configuration example uses the following devices:

- Two EX9204 Switches as core devices, software version: Junos OS Release 22.4R2-Sx or later
- Two QFX5120 Switches as distribution devices, software version: Junos OS Release 22.4R2-Sx or later
- Two access layer EX4400 Switches, software version: Junos OS Release 22.4R2-Sx or later
- One MX80 WAN router, software version: Junos OS Release 21.2R3-Sx or later
- Juniper Access Points
- Two Linux desktops that act as wired clients

NOTE: Juniper's recommended software version for Campus Fabric IP Clos is available under the EVPN/VXLAN ERB section at: https://supportportal.juniper.net/s/article/Junos-Software-Versions-Suggested-Releases-to-Consider-and-Evaluate?language=en_US

Figure 12: Campus Fabric IP Clos Topology



Juniper Mist Wired Assurance

IN THIS SECTION

- [Juniper Switch Onboarding to the Mist Cloud | 20](#)

Juniper Mist Wired Assurance, through the Mist UI, can be used to centrally manage all Juniper switches. Wired Assurance gives you full visibility on the devices that comprise your network’s access layer. The Juniper Mist 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, power over ethernet (PoE) compliance, switch-AP affinity, and virtual LAN (VLAN) insights.

Juniper Switch Onboarding to the Mist Cloud

<https://www.juniper.net/documentation/us/en/quick-start/hardware/cloud-ready-switches/topics/topic-map/step-1-begin.html>

Wired Assurance, through the Mist UI, 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 all layers of the campus fabric.
- Assignment of unique BGP AS numbers per device participating in the underlay and overlay.
- The creation of Virtual Routing and Forwarding (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) assigned to the access layer.
- IP addressing of each lo0.0 loopback.
- 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.juniper.net/documentation/us/en/software/mist/mist-wired/index.html>.

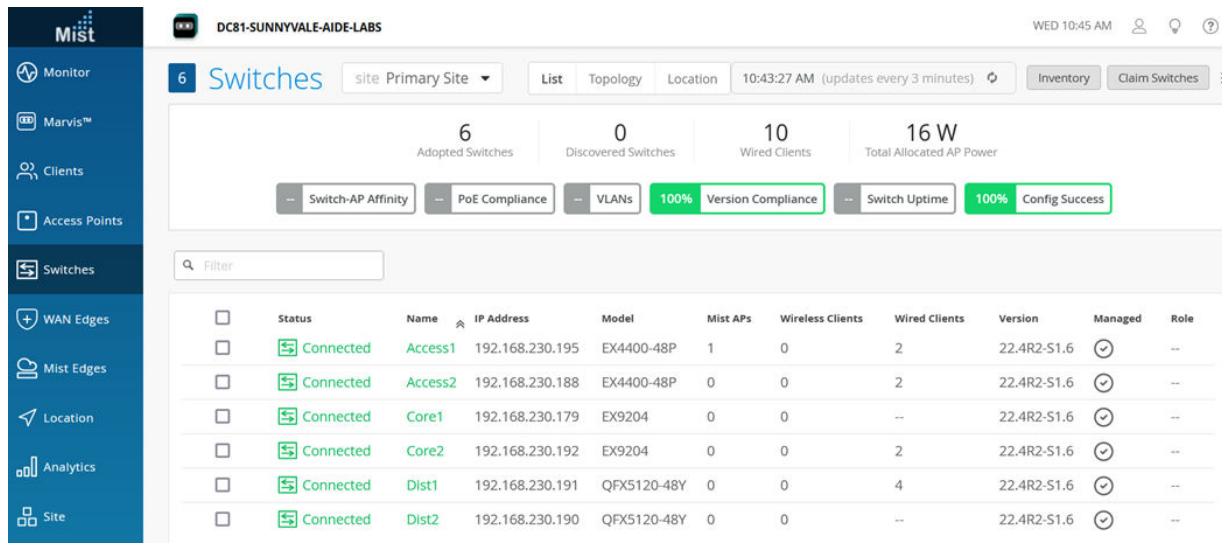
Juniper Mist Wired Assurance Switches

IN THIS SECTION

- [Overview | 21](#)
- [Templates | 21](#)
- [Topology | 29](#)

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 respective layers in the fabric to facilitate building and operating the fabric. See [Figure 13 on page 21](#).

Figure 13: Switches Assigned to the Primary Site



Overview

Use this JVD to deploy a single campus fabric with an L3 IP-based underlay network that uses EVPN as the control plane protocol and VXLAN as the data plane protocol in the overlay network.

Mist Wired Assurance configures eBGP on the directly connected links to exchange loopback routes, and eBGP between the core and distribution devices and distribution and access devices in the overlay to share reachability information about endpoints in the fabric.

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 (Organization, Site, and Switch) provides both scale and granularity.

Templates and the hierarchical model mean 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 Organization levels that apply to the same device, the narrower settings (in this case, Site) 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. Individual switches can also have their own unique configurations.

You can include individual command line interface (CLI) commands at any level of the hierarchy. Mist appends these commands to all the switches in that group on an “AND” basis. If the additional CLI commands apply to the pre-existing configuration on the device, the original configuration is overwritten with the configuration sent by Mist. If the added commands apply to a configuration that did not exist on the device previously, then the new configuration is appended to the existing device configuration during the update.

NOTE: If you run CLI commands for items not native to the Mist GUI, this configuration data is applied last. These commands overwrite existing configuration data within the same stanza. You can access the CLI command option from the Switch Template or individual switch configuration.

Figure 14: Additional CLI Commands Field

CLI CONFIGURATION
⌵ ⌵

Additional CLI Commands i

Under **Organization > Switch Templates**, we use the following template:

Switch Templates		
TEMPLATE	SITES	SWITCHES
campus-fabric	1	6

We provide a copy of the template configuration in JSON format below for import into your system for verification.

```
{
  "ntp_servers": [],
  "dns_servers": [
    "8.8.8.8",
    "9.9.9.9"
  ],
  "dns_suffix": [],
  "additional_config_cmds": [],
  "networks": {
    "vlan1099": {
      "vlan_id": 1099,
      "subnet": "10.99.99.0/24"
    },
    "vlan1088": {
      "vlan_id": 1088,
      "subnet": "10.88.88.0/24"
    },
    "vlan1033": {
      "vlan_id": 1033,
      "subnet": "10.33.33.0/24"
    }
  },
  "port_usages": {
    "myaccess": {
      "mode": "trunk",
      "disabled": false,
      "port_network": "vlan1033",
      "voip_network": null,
      "stp_edge": false,
      "port_auth": null,
      "all_networks": false,
      "networks": [
        "vlan1033",
        "vlan1088",
        "vlan1099"
      ],
      "speed": "auto",
      "duplex": "auto",
      "mac_limit": 0,
    }
  }
}
```

```
"persist_mac": false,
"poe_disabled": false,
"enable_qos": false,
"storm_control": {},
"mtu": 9018,
"description": ""
},
"myesilag": {
  "mode": "trunk",
  "disabled": false,
  "port_network": null,
  "voip_network": null,
  "stp_edge": false,
  "port_auth": null,
  "all_networks": true,
  "networks": [],
  "speed": "auto",
  "duplex": "auto",
  "mac_limit": 0,
  "persist_mac": false,
  "poe_disabled": false,
  "enable_qos": false,
  "storm_control": {},
  "mtu": 9014,
  "description": ""
},
"dynamic": {
  "mode": "dynamic",
  "reset_default_when": "link_down",
  "rules": []
},
"vlan1099": {
  "mode": "access",
  "disabled": false,
  "port_network": "vlan1099",
  "voip_network": null,
  "stp_edge": false,
  "all_networks": false,
  "networks": null,
  "port_auth": null,
  "speed": "auto",
  "duplex": "auto",
  "mac_limit": 0,
```

```
"persist_mac": false,
"poe_disabled": false,
"enable_qos": false,
"storm_control": {},
"mtu": 9014,
"description": "Corp-IT",
"disable_autoneg": false,
"mac_auth_protocol": null,
"enable_mac_auth": null,
"mac_auth_only": null,
"guest_network": null,
"bypass_auth_when_server_down": null
},
"vlan1088": {
  "mode": "access",
  "disabled": false,
  "port_network": "vlan1088",
  "voip_network": null,
  "stp_edge": false,
  "all_networks": false,
  "networks": null,
  "port_auth": null,
  "speed": "auto",
  "duplex": "auto",
  "mac_limit": 0,
  "persist_mac": false,
  "poe_disabled": false,
  "enable_qos": false,
  "storm_control": {},
  "mtu": 9014,
  "description": "Developers",
  "disable_autoneg": false,
  "mac_auth_protocol": null,
  "enable_mac_auth": null,
  "mac_auth_only": null,
  "guest_network": null,
  "bypass_auth_when_server_down": null
},
"vlan1033": {
  "mode": "access",
  "disabled": false,
  "port_network": "vlan1033",
  "voip_network": null,
```

```

    "stp_edge": false,
    "all_networks": false,
    "networks": null,
    "port_auth": null,
    "speed": "auto",
    "duplex": "auto",
    "mac_limit": 0,
    "persist_mac": false,
    "poe_disabled": false,
    "enable_qos": false,
    "storm_control": {},
    "mtu": 9014,
    "description": "Guest-WiFi",
    "disable_autoneg": false,
    "mac_auth_protocol": null,
    "enable_mac_auth": null,
    "mac_auth_only": null,
    "guest_network": null,
    "bypass_auth_when_server_down": null
  }
},
"switch_matching": {
  "enable": true,
  "rules": [
    {
      "name": "core",
      "match_model": "EX9204",
      "port_config": {},
      "additional_config_cmds": [
        ""
      ],
      "ip_config": {
        "type": "dhcp",
        "network": "default"
      },
      "oob_ip_config": {
        "type": "dhcp",
        "use_mgmt_vrf": false
      }
    }
  ],
  {
    "name": "distribution",
    "port_config": {},

```

```

    "additional_config_cmds": [
      ""
    ],
    "ip_config": {
      "type": "dhcp",
      "network": "default"
    },
    "oob_ip_config": {
      "type": "dhcp",
      "use_mgmt_vrf": false
    },
    "match_model[0:7]": "QFX5120"
  },
  {
    "name": "access",
    "port_config": {
      "ge-0/0/16": {
        "usage": "myaccess",
        "dynamic_usage": null,
        "critical": false,
        "description": "",
        "no_local_overwrite": true
      }
    },
    "additional_config_cmds": [
      ""
    ],
    "ip_config": {
      "type": "dhcp",
      "network": "default"
    },
    "oob_ip_config": {
      "type": "dhcp",
      "use_mgmt_vrf": false
    },
    "match_model[0:6]": "EX4400"
  }
]
},
"switch_mgmt": {
  "config_revert_timer": 10,
  "root_password": "juniper123",
  "protect_re": {

```

```
    "enabled": false
  },
  "tacacs": {
    "enabled": false
  }
},
"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
},
"acl_policies": [],
"mist_nac": {
  "enabled": true,
  "network": null
},
"name": "campus-fabric"
}
```

Topology

Wired Assurance provides the template for LAN and loopback IP addressing for each 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 adjacent devices within the Campus Fabric IP Clos.

The WAN router can be provisioned using the Mist UI but is separate from the campus fabric workflow. The WAN router has a southbound link aggregation group (LAG) configured to connect to the ESI-LAG on the core switches. WAN routers can be standalone or built as high-availability clusters.

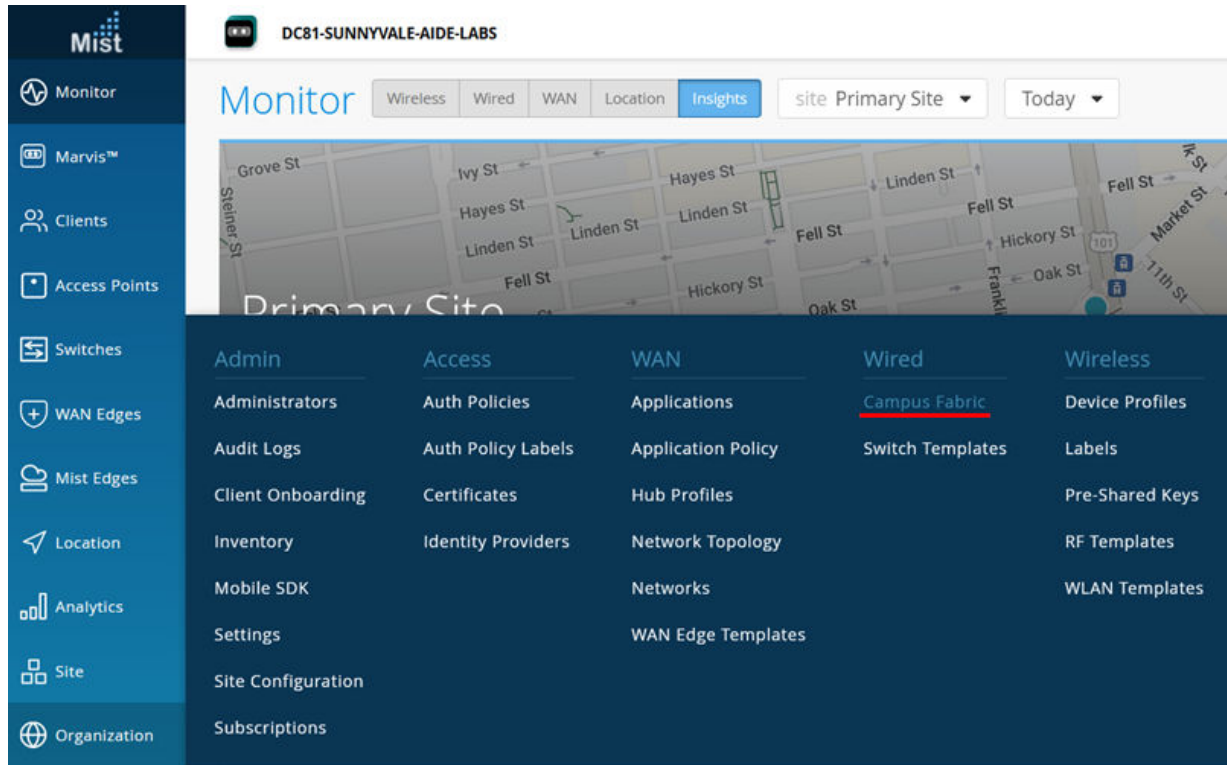
Create the Campus Fabric

IN THIS SECTION

- [Choose the Campus Fabric Topology | 31](#)
- [Topology Settings | 32](#)
- [Select Campus Fabric Nodes | 34](#)
- [Configure Networks | 35](#)
- [VRF | 36](#)
- [Networks | 36](#)
- [Other IP Configuration | 39](#)
- [Configure Campus Fabric Ports | 43](#)
- [Core Switches | 43](#)
- [Distribution Switches | 45](#)
- [Access Switches | 46](#)
- [Campus Fabric Configuration Confirmation | 47](#)
- [Apply VLANs to Access Ports | 49](#)

Navigate to **Organization** > **Campus Fabric** in the Mist UI.

Figure 15: Campus Fabric Location in Mist GUI



Mist provides the option of deploying a campus fabric at the Organization or Site level noted on the upper left-hand Campus Fabric menu as shown in [Figure 16 on page 30](#). For example, if you are building a campus-wide architecture with multiple buildings, and each building housing distribution and access switches, you can consider building an organization-level campus fabric. An organization-level campus fabric ties each of the sites together forming a holistic campus fabric. Otherwise, the site built with a single set of core, distribution, and access switches is sufficient.

Figure 16: Organization Level Campus Fabric Configuration Window

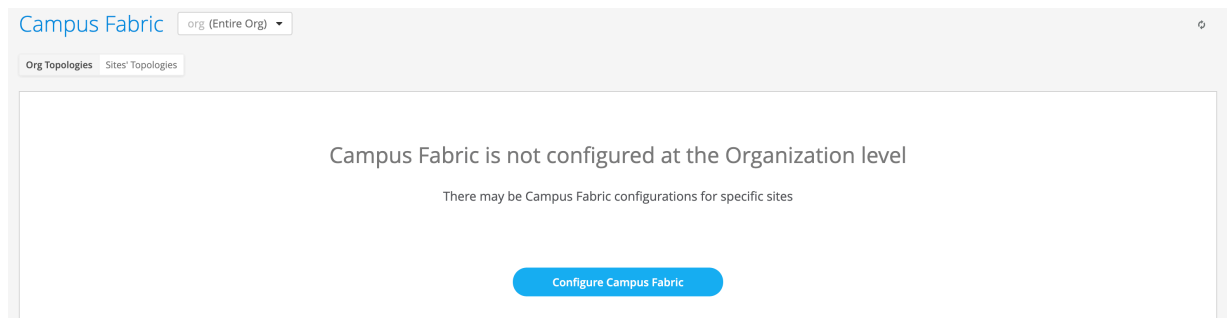
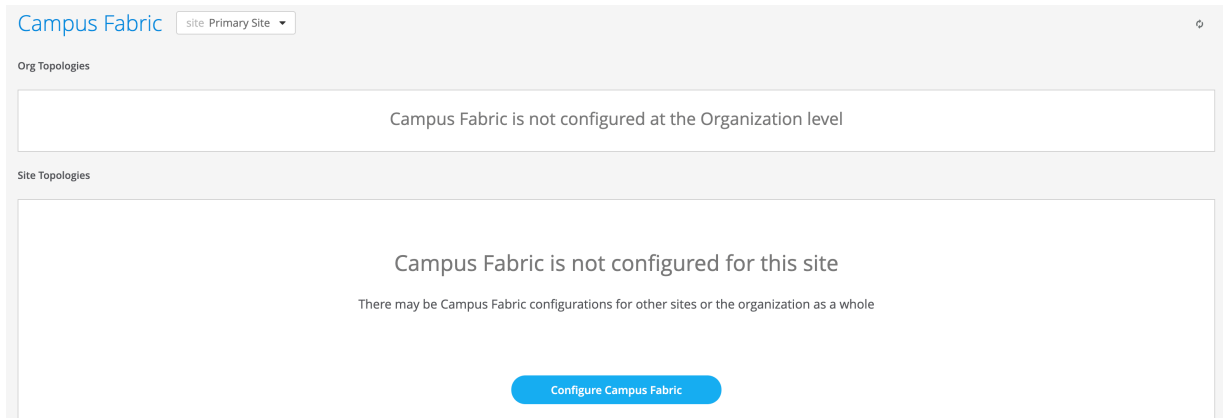


Figure 17: Site Level Campus Fabric Configuration Window



NOTE: Site-level campus fabric deployment is the focus of this JVD.

Choose the Campus Fabric Topology

Select the Campus Fabric IP Clos option as shown in [Figure 18 on page 32](#).

Figure 18: Campus Fabric Configuration Options

The screenshot shows the 'Campus Fabric Configuration' interface. At the top, there is a breadcrumb trail: '1. Topology', '2. Nodes', '3. Network Settings', '4. Ports', and '5. Confirm'. Below this, the main heading is 'Choose Campus Fabric Topology' with a sub-instruction: 'Choose the topology you want to construct and configure related options'. Under the heading 'TOPOLOGY TYPE', three options are listed: 'EVPN Multihoming' (Collapsed core with ESI-Lag), 'Campus Fabric Core-Distribution' (EVPN core/distribution with ESI-Lag), and 'Campus Fabric IP Clos' (Campus fabric with L3 at the edge). The 'Campus Fabric IP Clos' option is selected and highlighted with a blue border. Below the topology selection, there are two panels: 'CONFIGURATION' and 'TOPOLOGY SETTINGS'. The 'CONFIGURATION' panel has a 'Topology Name' field containing 'Campus Fabric IPClos'. The 'TOPOLOGY SETTINGS' panel contains three fields: 'BGP Local AS' with value '65001', 'Subnet' with value '10.255.240.0/20', 'Auto Router ID Subnet' with value '172.16.254.0/23', and 'Loopback per-VRF subnet' with value '172.16.192.0/19'. Each of the three settings fields has a small information icon to its right.

In the Topology Name field, provide a name following company standards.

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 Mist to ensure the AS numbers are never advertised outside of the fabric.
- **Subnet:** Represents the pool of IP addresses used for P2P links between devices. You can use whatever range suits your deployment. Mist breaks this subnet into /31 subnet addressing per link. This number can be modified to suit the specific deployment scale. For example, a /24 subnet 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 /32 assigned from this pool. You can use whatever range suits your deployment. VXLAN tunneling 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.
- **Loopback per-VRF-subnet:** Represents a second pool of loopback IP addresses which are associated each 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 that is external to the fabric. This is the case for anycast fabrics like ERB and IP Clos. If those L3-VRF use a dedicated loopback IP address per VRF and switch it is easy to send back returning answers to an originating VRF or switch.

NOTE: In previous versions of this document, you did not have the default configuration fields for Auto Router ID, Subnet, and loopback per-VRF subnet. Instead, you had a field for loopback prefix definition like shown in [Figure 19 on page 33](#) and then you had to assign the loopback addresses for each fabric node manually. This now changed towards automatic loopback address assignments via the configuration of the Prefix Pool.

Figure 19: Topology Settings

TOPOLOGY SETTINGS	
BGP Local AS	<input type="text" value="65001"/>
	<small>(2-byte or 4-byte)</small>
Loopback prefix	<input type="text" value="/24"/> ⓘ
Subnet	<input type="text" value="10.255.240.0/20"/> ⓘ
	<small>(xxx.xxx.xxx.xxx/xx)</small>

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

Select Campus Fabric Nodes

Select devices to participate at each layer of the Campus Fabric IP Clos. We recommend that you validate each device's presence in the Site switch inventory before the creation of the campus fabric.

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

The Services Block Router is where the campus fabric interconnects external devices such as firewalls, routers, or other critical devices. For example, DHCP and RADIUS servers. Devices to which external services connect to the campus fabric are known as border leaf devices. 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.

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 ⓘ

Core

+
Select Switches

Distribution

Access

Filter

<input type="checkbox"/>	Name	MAC Address	Serial	Router ID	Model
<input type="checkbox"/>	Dist2	d8:53:9a:64:b5:c0	XH3121410874	--	QFX5120-48Y
<input type="checkbox"/>	Dist1	d8:53:9a:64:6f:c0	XH3121410895	--	QFX5120-48Y
<input checked="" type="checkbox"/>	Core2	f4:b5:2f:f3:f4:00	JN122EFFFFFC	--	EX9204
<input checked="" type="checkbox"/>	Core1	f4:b5:2f:f4:04:00	JN122EFFFFFC	--	EX9204
<input type="checkbox"/>	Access2	00:cc:34:f3:cf:00	ZD4422030024	--	EX4400-48P
<input type="checkbox"/>	Access1	00:cc:34:f4:72:00	ZD4422070133	--	EX4400-48P

Select 2 Cancel

NOTE: Placing the Services Block Router on a dedicated pair of switches (or single switch) alleviates the encapsulation and de-encapsulation 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.

Once all layers have selected the appropriate devices, you must provide an underlay loopback IP address for each device. This loopback is associated with a logical construct called a VTEP and is used as the

source address of the VXLAN tunnel. Campus Fabric IP Clos has VTEPs for VXLAN tunneling on the access switches and the core switches when you enable the Core Border option.

When you define an Auto Router ID Subnet prefix, the underlay loopback IP address and router ID assignment happen automatically. There is no need to manually assign them. You may still see warnings as shown in [Figure 20 on page 35](#) about an unassigned router ID. You can ignore those since the automatic assignments happen at a later phase.

Figure 20: Select Campus Fabric Switch 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 ⓘ

Core

Core1 Core2 Select Switches

Distribution

Dist1 Dist2 Select Switches

Access ^{*required}

Access1 Access2 Select Switches

Core1

MAC Address f4:b5:2f:f4:04:00

Model EX9204

Status connected

Site Primary Site

Router ID

NOTE: If the Auto Router ID Subnet Field is not configured (empty), you can use the previous mode of operation and manually assign the underlay loopback IP addresses as router IDs on each device that needs one. Make sure that all IP addresses are in the same subnet as required by Mist Cloud Fabric configuration.

Configure Networks

Enter network information such as VLANs and VRF (routing instances for traffic isolation purposes) options. VLANs are mapped to virtual network identifiers (VNIs) and can optionally be mapped to VRFs to provide customers a way to logically separate traffic patterns such as IoT devices from Corp IT.

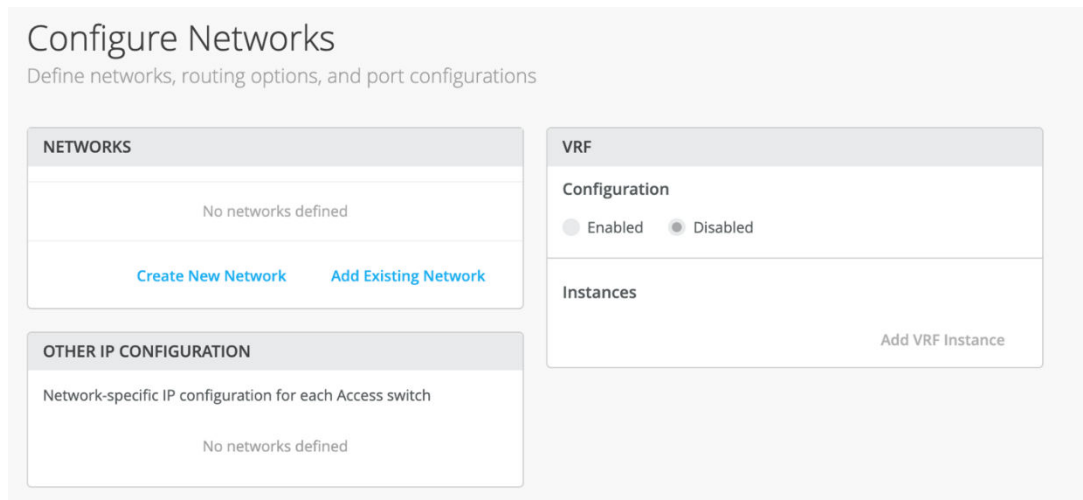
VRF

In campus fabric deployment, the use of EVPN VXLAN supports native traffic isolation using routing instances, commonly called VRFs, for macro-segmentation purposes.

For more information on routing instances, see <https://www.juniper.net/documentation/us/en/software/junos/routing-overview/topics/concept/routing-instances-overview.html>.

VLANs can be placed into a common VRF. Here, all VLANs within each VRF have full connectivity to each other and other external networking resources. A common use case includes most enterprise domains that isolate guest wireless traffic to save Internet connectivity.

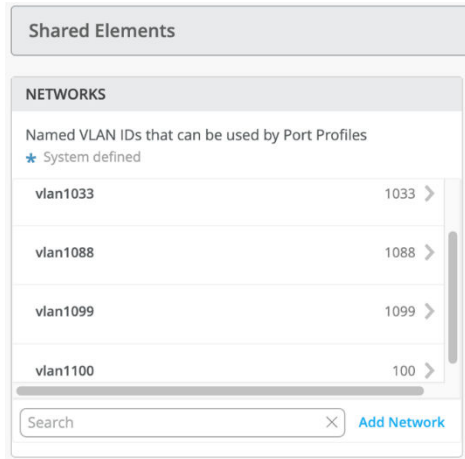
By default, the campus fabric provides complete isolation between VRFs, thus forcing inter-VRF communications to traverse a firewall or other security device. This aligns with most enterprise security use cases and compliance and is represented in this document.



Networks

VLANs can be created or imported under this section including the IP subnet and Default Gateway for each VLAN.

The **Shared Elements** section of the campus fabric template includes the Networks section mentioned above where VLANs are created. This can be found under the **Organization > Switch Templates** section, then choose the appropriate template:



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

NETWORKS

Add Existing Network ✓ ✕

Available Networks

<input type="checkbox"/> Name	VLAN ID
<input type="checkbox"/> vlan1033	1033
<input type="checkbox"/> vlan1088	1088
<input type="checkbox"/> vlan1099	1099

Import from Template

Template

DC81-IP-Clos:3 Networks ✓

<input checked="" type="checkbox"/> Name	VLAN ID
<input checked="" type="checkbox"/> vlan1033	1033
<input checked="" type="checkbox"/> vlan1088	1088
<input checked="" type="checkbox"/> vlan1099	1099

Search ✕

You can edit existing networks, manually add new networks, or import from an existing template:

The screenshot shows a dialog box titled "Edit Network" with a close button (X) and a checkmark. The dialog contains three input fields:

- Name:** A text field containing "vlan1099".
- VLAN ID:** A text field containing "1099". Below the field is the text "(1 - 4094 or {{siteVar}})".
- Subnet:** A text field containing "10.99.99.0/24". To the right of the field is an information icon (i).

Other IP Configuration

Mist Wired Assurance provides automatic IP addressing of integrated routing and bridging (IRB) interfaces for each of the VLANs. Then, **Port Profiles** and **Port Configuration** associate the VLAN with specific ports. In this case, we selected Campus Fabric IP Clos routed at Edge at the beginning of the campus fabric build.

CONFIGURATION

Topology Name

Campus Fabric IPClos

Topology Sub-type

- Routed at Distribution
Centrally-routed and bridged with gateways on the Distribution
- Routed at Edge
Edge-routed and bridged with anycast gateways on the access

This option uses anycast addressing for all devices participating in the L3 subnet. In this case, Access1 and Access2 switches are configured with shared IP addresses for each L3 subnet.

For more information on anycast gateways, see <https://www.juniper.net/documentation/us/en/software/junos/evpn-vxlan/topics/concept/evpn-mclag-irb-gateway-anycast-address.html>.

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 >

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 >

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. Here, you build the first corp-it VRF and select the pre-defined VLAN, vlan1099.

VRF

Configuration

Enabled Disabled

Instances

No VRF instances defined

[Add VRF Instance](#)

New VRF Instance ✓ ✕

Name

corp-it

Networks

vlan1099 ✕ +

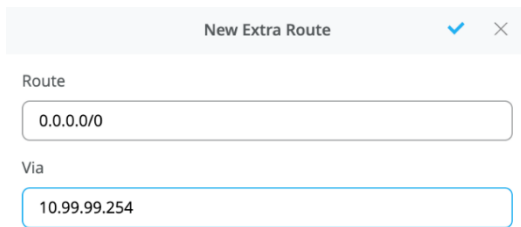
Extra Routes

No extra routes defined

[Add Extra Routes](#)

By default, inter-VRF communications are not supported within the campus fabric. If inter-VRF communications are required, each VRF can include extra routes such as a default route that instructs the campus fabric to use an external router or firewall for further routing capabilities or security inspection. In this example, all traffic is trunked over the ESI-LAG and the Juniper MX router handles inter-VRF routing. See [Figure 12 on page 19](#).

Notice that the MX router participates in the VLANs defined within the campus fabric and is the gateway of last resort for all traffic leaving the subnet. Select the Add Extra Routes option to inform Mist to forward all traffic leaving 10.99.99.0/24 to use the next hop of the Juniper MX router: 10.99.99.254.



The image shows a configuration window titled "New Extra Route" with a blue checkmark and a close button (X). It contains two input fields: "Route" with the value "0.0.0.0/0" and "Via" with the value "10.99.99.254".

Create two additional VRFs:

- developers using vlan 1088 with 0.0.0.0/0 utilizing 10.88.88.254
- guest-wifi using vlan 1033 with 0.0.0.0/0 utilizing 10.33.33.254

Configure Networks

Define networks, routing options, and port configurations

NETWORKS

vlan1033	1033 >
vlan1088	1088 >
vlan1099	1099 >

[Create New Network](#) [Add Existing Network](#)

VRF

Configuration
 Enabled Disabled

Instances

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

[Add VRF Instance](#)

OTHER IP CONFIGURATION

Network-specific IP configuration for each Access switch

Access1	3 Static >
Access2	3 Static >

VRF

Configuration
 Enabled Disabled

Instances

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

[Add VRF Instance](#)

Now that all VLANs are configured and assigned to each VRF, click **Continue** at the upper-right section of the Mist UI to move to the next step.

Configure Campus Fabric Ports

The final step is to select physical ports among core, distribution, and access switches.

Ports
Select switch ports for Fabric connections

Core Switches

Switch	Model	Link to Distribution
Core2	EX9204	0/2 ?

FPC 1 FPC 2

1

SFP+

1	3	5	7	1	3	5	7	1	3	5	7	1	3	5	7	EX9200-32XS
0	2	4	6	0	2	4	6	0	2	4	6	0	2	4	6	

Core1	EX9204	0/2 ?
-------	--------	-------

Distribution Switches

QFX5120-48Y [Edit Ports for all QFX5120-48Y](#)

Switch	Model	Link to Core	Link to Access
Dist2	QFX5120-48Y	0/2 ?	0/2 ?
Dist1	QFX5120-48Y	0/2 ?	0/2 ?

Access Switches

EX4400-48P [Edit Ports for all EX4400-48P](#)

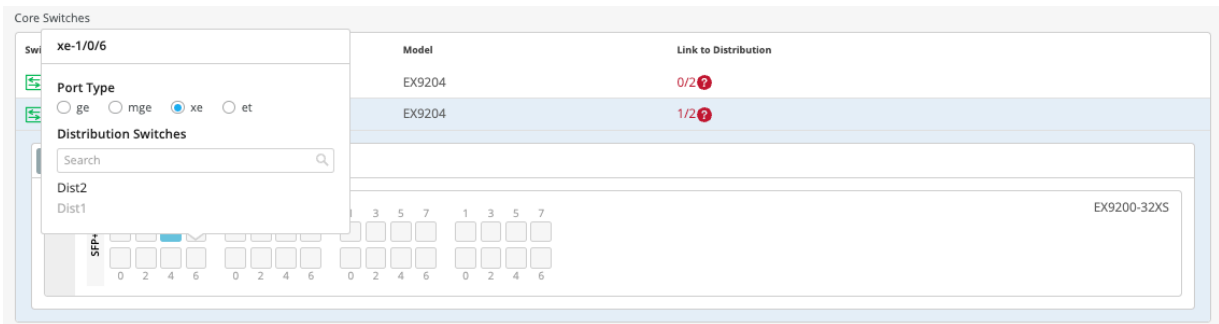
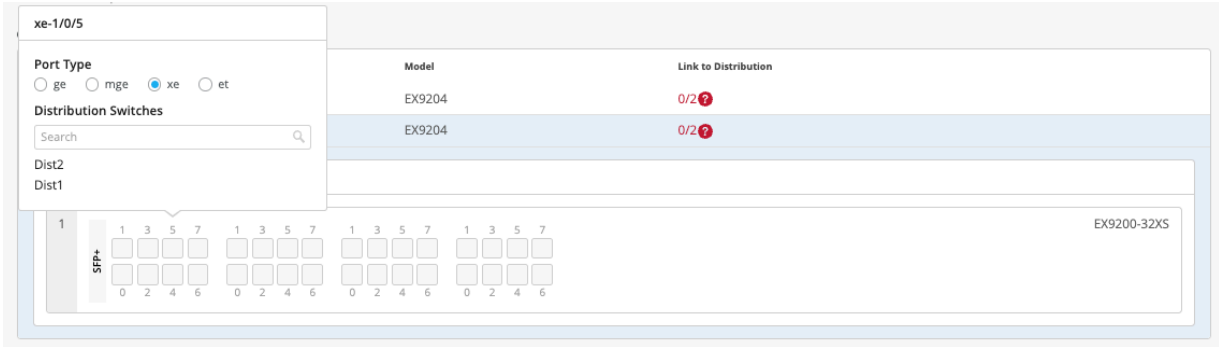
Switch	Link to Distribution
Access2	0/2 ?
Access1	0/2 ?

NOTE: We recommend that you have the output of the “show lldp neighbors” command from each switch. Juniper enables LLDP out of the box and provides additional LLDP attributes when the switch is added to a campus fabric. This output provides a source of truth for which ports should be selected at each layer.

Core Switches

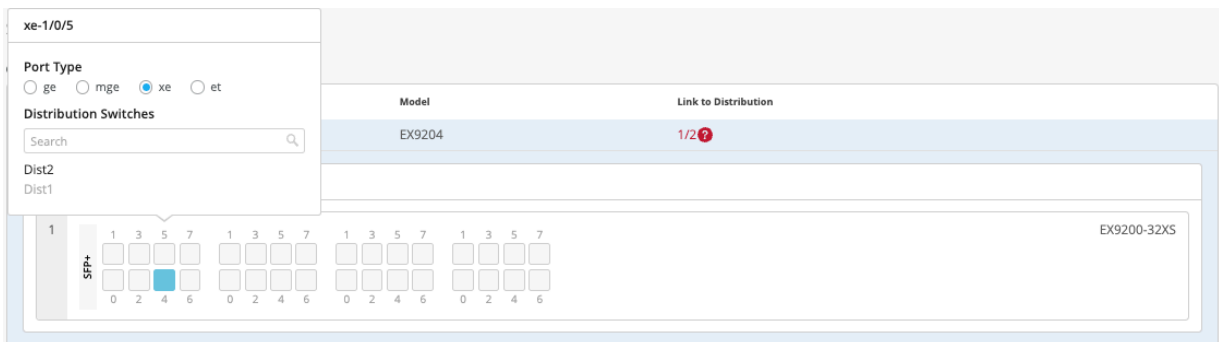
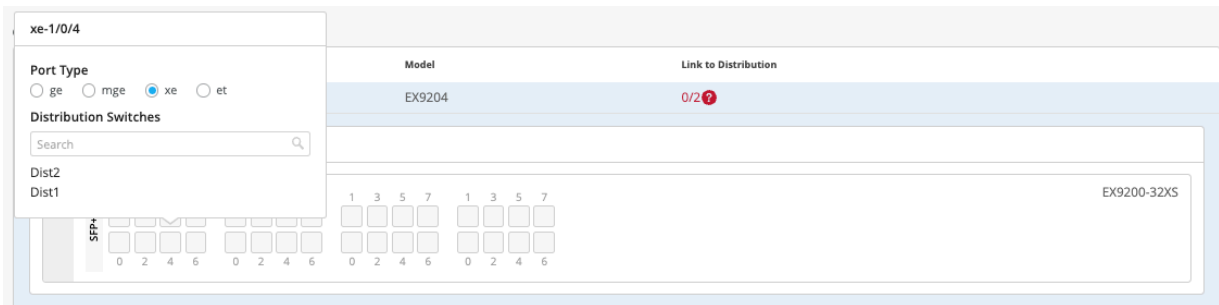
Core1:

Starting with Core1, select xe-1/0/5 and xe-1/0/6 terminating on distribution switches 1 and 2 respectively.



Core2:

On Core2, select xe-1/0/4 and xe-1/0/5 terminating on distribution switches 1 and 2 respectively.



Distribution Switches

When configuring the distribution switches, Dist1 and Dist2, you'll notice two interconnect options exist:

- Link to Core
- Link to Access

Dist1:

Select **Link to Core** and choose xe-0/0/5 and xe-0/0/4 terminating on core switches 1 and 2 respectively.

Distribution Switches
xe-0/0/5

Port Type
 ge mge xe et

Port Connection

Model	Link to Core	Link to Access
QFX5120-48Y	0/2 ?	0/2 ?
QFX5120-48Y	0/2 ?	0/2 ?

SFP28 ports: 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55

QSF28 Uplink ports: 48, 50, 52, 54, 49, 51, 53, 55

Edit Ports for all QFX5120-48Y

Distribution Switches
xe-0/0/4

Port Type
 ge mge xe et

Port Connection

Model	Link to Core	Link to Access
QFX5120-48Y	0/2 ?	0/2 ?
QFX5120-48Y	1/2 ?	0/2 ?

SFP28 ports: 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 49, 51, 53, 55

QSF28 Uplink ports: 48, 50, 52, 54, 49, 51, 53, 55

Edit Ports for all QFX5120-48Y

Select **Link to Access** and choose ge-0/0/36 and ge-0/0/37 terminating on access switches 1 and 2 respectively.

Distribution Switches
QFX5120-48Y

Switch
 Dist2
 Dist1

Model
 QFX5120-48Y
 QFX5120-48Y

ge-0/0/36
 Port Type
 ge mge xe et

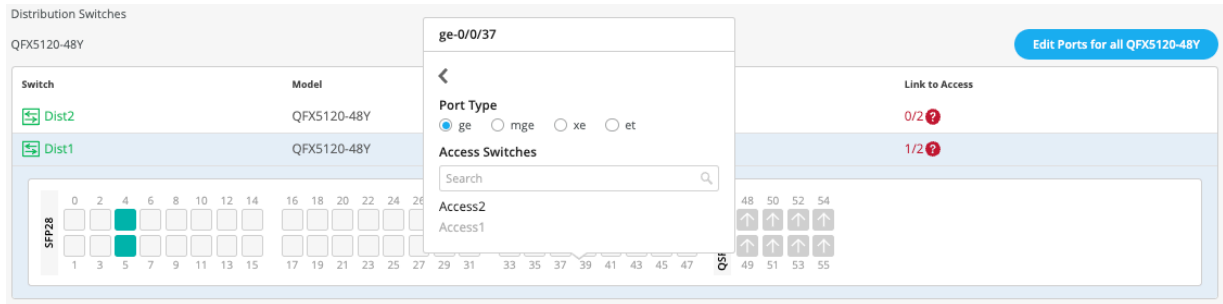
Access Switches
 Search
 Access2
 Access1

Link to Access
 0/2 ?
 0/2 ?

SFP28 ports: 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 49, 51, 53, 55

QSF28 Uplink ports: 48, 50, 52, 54, 49, 51, 53, 55

Edit Ports for all QFX5120-48Y



Next, select the following interconnects of **Dist2**:

- Link to Core:
 - xe-0/0/6 – Core1
 - xe-0/0/5 – Core2
- Link to Access:
 - ge-0/0/36 – Access2
 - ge-0/0/37 – Access1

Access Switches

Finally, select the following interface combinations for Access1 and Access2:

NOTE: QFX 5120-48Y Switch is an example switch that is targeted for the distribution layer in a campus fabric. The device supports blocks of four ports per PHY; Ports0-3, 4-7, and so on. All ports within the same PHY must operate at the same speed.

Access1:

- ge-0/0/36 – Distribution Switch – Dist1
- ge-0/0/37 – Distribution Switch – Dist2

Access2:

- ge-0/0/36 – Distribution Switch – Dist1
- ge-0/0/37 – Distribution Switch – Dist2

Once you have completed selecting all requisite port combinations, click **Continue** at the upper right-hand corner of the Mist UI.

Campus Fabric Configuration Confirmation

This last section provides the ability to confirm each device's configuration as shown in [Figure 21 on page 47](#):

Figure 21: Campus Fabric Configuration Confirmation

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

Core

Distribution

Access

Core1

MAC Address f4:b5:2f:f4:04:00
Model EX9204
Status connected
Site Primary Site
Router ID --

ID	IP Address	Name
1033	--	vlan1033
1088	--	vlan1088
1099	--	vlan1099

Switch	Port Id
Dist1	xe-1/0/5
Dist2	xe-1/0/6

NOTE: Because we have configured the use of Auto Router ID Subnet, the underlay loopback IP addresses may still not be assigned in this page and warnings may appear as shown above. Please ignore this for now as the assignment happens when you apply the configuration for the first time.

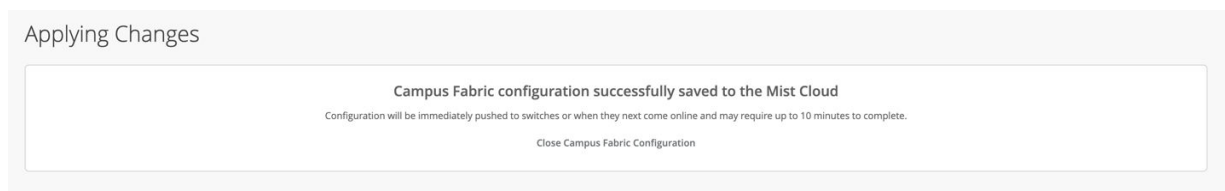
Once you have completed verification, click the **Apply Changes** button at the upper right-hand corner of the Mist UI.

X [Campus Fabric Configuration](#) 1. Topology 2. Nodes 3. Network Settings 4. Ports 5. Confirm ← Back [Apply Changes](#)

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

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

- Mist builds point-to-point interfaces between all 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 at 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 level for device loopback reachability. The primary goal of the eBGP overlay is to support customer traffic using EVPN-VXLAN.
- IP addressing of each L3 gateway IRB assigned to the access layer.
- IP addressing of each underlay lo0.0 loopback. This happens 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.
- Mist creates VXLAN to VLAN mapping using Virtual Network Identifier (VNI) addresses that are automatically assigned.
- VRF creation of corp-it, developers, and guest-wifi instances, each with an associated VLAN.
- VXLAN tunnelling creation between access devices and access-core devices (in support of the northbound MX 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.



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

Campus Fabric site Primary Site Create Campus Fabric

Org Topologies

Campus Fabric is not configured at the Organization level

Site Topologies

Name	Topology ID	Site	Type	Routed At	Date Created
Campus Fabric IP Clos	9ac2078-c2cc-40e5-a701-58954d8711b9	Primary Site	Campus Fabric IP Clos	Access	02:36:47 PM, Mar 15 2023

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.

< Campus Fabrics : Campus Fabric IPClos Edit Configuration Delete Connection Table

Core1

MAC Address f4:b5:2f:f4:04:00

Model EX9204

Status connected

Site Primary Site

Router ID 172.16.254.2

VLANs

ID	IP Address	Name
1033	--	vlan1033
1088	--	vlan1088
1099	--	vlan1099

Connections to Distribution

Switch	RX Bytes	TX Bytes	Link Status
Dist1	121.9 MB	2.8 MB	Up
Dist2	122.2 MB	2.5 MB	Up

Remote Shell Insights Details

Neighbor Information 2:54 PM (Updates Every 3 Minutes)

Status	State	Neighbor	Neighbor AS	Local AS	Uptime	RX Routes	RX Packets	TX Packets	VRF Name	Neighbor Type
Connected	Established	10.255.240.9	65004	65002	1h 36m	5	4	219	218	default Underlay
Connected	Established	10.255.240.7	65003	65002	1h 36m	5	3	220	218	default Underlay
Connected	Established	172.16.254.3	65003	65002	1h 36m	6	3	229	227	default Overlay
Connected	Established	172.16.254.4	65004	65002	1h 36m	6	3	230	225	default Overlay

Connection Table spreadsheet:

Role 1	Switch 1	Mac 1	Model 1	Serial 1	Site 1	Port Role 1	AE 1	Port 1	< --- >	Port 2	AE 2	Port Role 2	Site 2	Serial 2	Model 2	Mac 2	Switch 2	Role 2
distribution	Dist2	d8539a64b5c0	QFX5120-48Y	XH3121410874	Primary Site	uplink		xe-0/0/5	< --- >	xe-1/0/5		downlink	Primary Site	JN122EFFFRFC	EX9204	f4b52ff3f400	Core2	core
distribution	Dist2	d8539a64b5c0	QFX5120-48Y	XH3121410874	Primary Site	uplink		xe-0/0/6	< --- >	xe-1/0/6		downlink	Primary Site	JN122EFFFRFC	EX9204	f4b52ff40400	Core1	core
distribution	Dist2	d8539a64b5c0	QFX5120-48Y	XH3121410874	Primary Site	downlink		ge-0/0/36	< --- >	ge-0/0/36		uplink	Primary Site	ZD4422030024	EX4400-48P	00cc34f3cf00	Access2	access
distribution	Dist2	d8539a64b5c0	QFX5120-48Y	XH3121410874	Primary Site	downlink		ge-0/0/37	< --- >	ge-0/0/37		uplink	Primary Site	ZD4422070133	EX4400-48P	00cc34f47200	Access1	access
distribution	Dist1	d8539a646fc0	QFX5120-48Y	XH3121410895	Primary Site	uplink		xe-0/0/4	< --- >	xe-1/0/4		downlink	Primary Site	JN122EFFFRFC	EX9204	f4b52ff3f400	Core2	core
distribution	Dist1	d8539a646fc0	QFX5120-48Y	XH3121410895	Primary Site	uplink		xe-0/0/5	< --- >	xe-1/0/5		downlink	Primary Site	JN122EFFFRFC	EX9204	f4b52ff40400	Core1	core
distribution	Dist1	d8539a646fc0	QFX5120-48Y	XH3121410895	Primary Site	downlink		ge-0/0/37	< --- >	ge-0/0/37		uplink	Primary Site	ZD4422030024	EX4400-48P	00cc34f3cf00	Access2	access
distribution	Dist1	d8539a646fc0	QFX5120-48Y	XH3121410895	Primary Site	downlink		ge-0/0/36	< --- >	ge-0/0/36		uplink	Primary Site	ZD4422070133	EX4400-48P	00cc34f47200	Access1	access

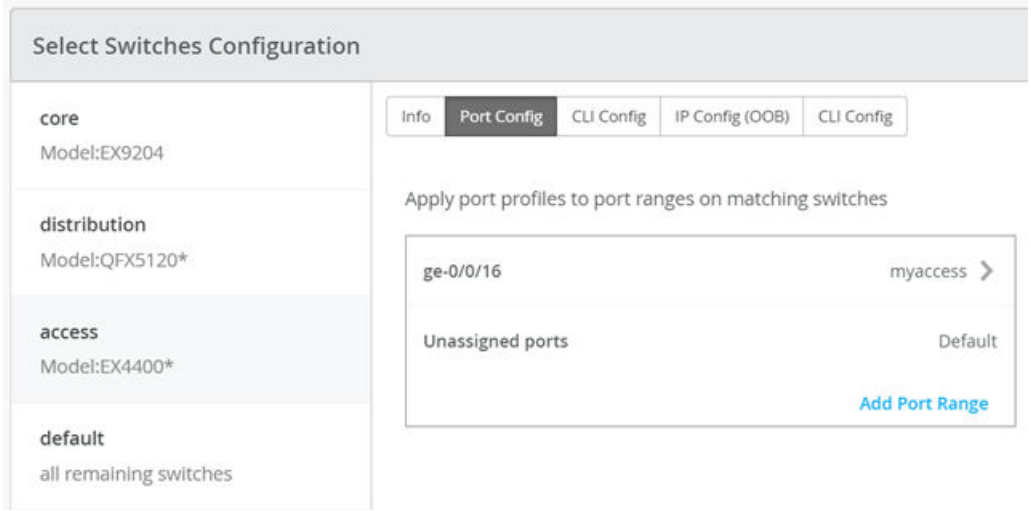
Apply VLANs to Access Ports

As previously discussed, Mist provides the ability to templatzize well-known services such as RADIUS, NTP, DNS, and others that can be used across all devices within a site. These templates can also include

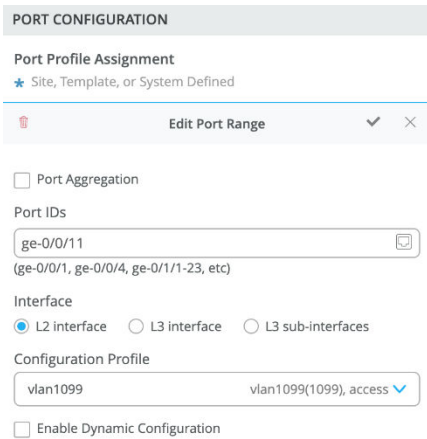
VLANs and port profiles that can be targeted at each device within a site. The last step before verification is to associate VLANs with the requisite ports on each access switch.

In this case, Desktop1 and Desktop2 are associated with different ports on each access switch which requires the configuration to be applied to Access1/2 respectively. See [Figure 12 on page 19](#).

Mist APs connect to the same port on Access1/2 allowing the Switch Template to be customized with this configuration. For example, the following (found under the **Organization > Switch Template**) is customized to associate each switch with its role: Core, Distribution, and Access. Additionally, all access switches (defined by EX4400 Switch in this example) associated the AP port profile named “myaccess” with ge-0/0/16 without the need to configure each switch.



Using Access1 as an example, we apply vlan1099 to port ge-0/0/11 under the Port Configuration section on the Access1 switch. In this example, vlan1099 (corp-it), vlan1088 (developers), and vlan1033 (guest-wifi) are defined in the Switch Template. These VLANs are defined under the **Organization > Switch template** section. Here, vlan1099 is selected under the configuration profile.



The Switch Template definition for vlan1099 is shown in [Figure 22 on page 51](#), representing attributes associated with VLANs such as dot1x authentication, Quality of Service (QoS), and power over Ethernet (PoE). Similarly, you can configure vlan1088 and vlan1033.

Figure 22: Switch Template Definition

Edit Port Profile ✓ ✕

Name
vlan1099

Port Enabled
 Enabled Disabled

Description
Corp-IT

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)

PoE
 Enabled Disabled

STP Edge
 Yes No

QoS
 Enabled Disabled
 Enable MTU

Storm Control
 Enabled Disabled

Persistent (Sticky) MAC Learning

Verification

IN THIS SECTION



BGP Underlay | 53

- Purpose | 54
- Action | 54
- Verification of BGP Peering | 55
- EVPN VXLAN Verification Between Access and Core Switches | 57
- Verification of the EVPN Database on Both Access Switches | 57
- Verification of VXLAN Tunnelling Between Access and Core Devices | 58
- External Campus Fabric Connectivity Through the Border Gateway Core EX9204 Switches | 59

Verification of the Campus Fabric IP Clos deployment. See [Figure 12 on page 19](#).

Currently, there are two desktops to validate the campus fabric. Let's take a quick look to see if Desktop1 can connect internally and externally. A third-party tool such as SecureCRT can be used to validate each desktop's configuration with Desktop1 shown in [Figure 23 on page 52](#).

Figure 23: Configuration Validation in CLI

```

root@desktop1:~# ifconfig vlan1099
vlan1099: 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:fec6:8a58 prefixlen 64 scopeid 0x20<link>
    ether 52:54:00:c6:8a:58 txqueuelen 1000 (Ethernet)
    RX packets 421822 bytes 434750459 (434.7 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 106373 bytes 5238868 (5.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@desktop1:~# ip r
default via 10.99.99.1 dev vlan1099 proto static
10.99.99.0/24 dev vlan1099 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 10.99.99.1 -c2
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=5.91 ms
64 bytes from 10.99.99.1: icmp_seq=2 ttl=64 time=5.69 ms

--- 10.99.99.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 5.690/5.799/5.909/0.109 ms
root@desktop1:~# ping 10.99.99.254 -c2
PING 10.99.99.254 (10.99.99.254) 56(84) bytes of data.

--- 10.99.99.254 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1018ms

```

Validation steps:

- Confirmed local IP address, VLAN and default gateway were configured on Desktop1.
- Can ping default gateway—indicates that we can reach the access switch.
- Ping to WAN router failed (10.99.99.254)—we need to troubleshoot.

Start by validating the campus fabric in the Mist UI, by navigating to **Organization > Campus Fabric**.

Site Topologies			
Name	Topology ID	Site	Date Created
DCB1-IPClos	1f4467cc-bfaf-4439-b91a-89a66d79d74c	Primary Site	05:46:13 PM, Nov 7 2022

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

- 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.

< Campus Fabrics : **Campus Fabric IPClos** Edit Configuration Delete

Core

Distribution

Access

Core1

MAC Address f4:b5:2f:f4:04:00
 Model EX9204
 Status connected
 Site Primary Site
 Router ID 172.16.254.2

VLANs

ID	IP Address	Name
1033	--	vlan1033
1088	--	vlan1088
1099	--	vlan1099

Connections to Distribution

Switch	RX Bytes	TX Bytes	Link Status
> Dist1	123.2 MB	4.1 MB	Up
> Dist2	123.6 MB	3.8 MB	Up

Neighbor Information 4:24 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	10.255.240.9	65004	65002	3h 6m	5	4	415	416	default	Underlay
● Connected	Established	10.255.240.7	65003	65002	3h 6m	5	3	416	417	default	Underlay
● Connected	Established	172.16.254.3	65003	65002	3h 6m	27	10	471	455	default	Overlay
● Connected	Established	172.16.254.4	65004	65002	3h 6m	27	17	479	457	default	Overlay

[Remote Shell](#) [Insights](#) [Details](#)

BGP Underlay

Purpose

Verifying the state of eBGP between adjacent 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.
- Bi-directional forwarding (BFD), to decrease convergence times during failures.
- BGP peering.
- Loopback to VXLAN reachability.

Without requiring verification at each layer, the focus can be on Dist1 and Dist2 and their eBGP relationships with Access1 and Access2, and Core1 and Core2. If both distribution switches have “established” eBGP peering sessions with each adjacent layer, you can move to the next phase of verification.

Due to the automated assignment of loopback IP addresses, for this fabric, we have the following configuration to remember:

Switch Type	Switch Name	Auto assigned Loopback IP
Core	Core1	172.16.254.2
Core	Core2	172.16.254.1
Distribution	Dist1	172.16.254.3
Distribution	Dist2	172.16.254.4
Access	Access1	172.16.254.6
Access	Access2	172.16.254.5

Action

Verify that BGP sessions are established from Dist1 and Dist2 with access and core 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 Mist GUI or using an external application such as SecureCRT or Putty.

Verification of BGP Peering

Dist1:

From **Switch > Utilities**, access Remote Shell via the bottom right of the campus fabric, from the switch view or via Secure Shell (SSH).

```
mist@Dist1> 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 10 8 0 0 0 0
bgp.evpn.0 58 27 0 0 0 0
Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn State|#Activ
e/Received/Accepted/Damped...
10.255.240.2 65001 435 430 0 0 3:14:54 Establ
inet.0: 2/3/3/0
10.255.240.6 65002 437 433 0 0 3:14:48 Establ
inet.0: 2/3/3/0
10.255.240.11 65005 434 430 0 0 3:14:47 Establ
inet.0: 2/2/2/0
10.255.240.13 65006 434 432 0 0 3:14:48 Establ
inet.0: 2/2/2/0
172.16.254.1 65001 477 495 0 0 3:14:48 Establ
bgp.evpn.0: 0/10/10/0
172.16.254.2 65002 474 487 0 0 3:14:38 Establ
bgp.evpn.0: 0/10/10/0
172.16.254.5 65005 477 483 0 0 3:14:39 Establ
bgp.evpn.0: 13/16/16/0
172.16.254.6 65006 469 481 0 0 3:14:40 Establ
bgp.evpn.0: 14/22/22/0
```

From the BGP summary, we can see that the underlay (10.255.240.X) peer relationships are established to indicate 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 underlay loopback addresses. This demonstrates underlay loopback reachability.

We can also see the routes received. Since the time established is roughly equal, this looks good so far.

The campus fabric build illustrates per device real-time BGP peering status shown in [Figure 24 on page 56](#) from Dist1.

Figure 24: BGP Neighbor Information

Neighbor Information 4:33 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	10.255.240.2	65001	65003	3h 15m	3	4	436	431	default	Underlay
● Connected	Established	10.255.240.6	65002	65003	3h 15m	3	5	438	434	default	Underlay
● Connected	Established	10.255.240.11	65005	65003	3h 15m	2	5	435	432	default	Underlay
● Connected	Established	10.255.240.13	65006	65003	3h 15m	2	5	435	433	default	Underlay
● Connected	Established	172.16.254.1	65001	65003	3h 15m	10	27	478	496	default	Overlay
● Connected	Established	172.16.254.2	65002	65003	3h 15m	10	27	475	488	default	Overlay
● Connected	Established	172.16.254.5	65005	65003	3h 15m	16	14	478	484	default	Overlay
● Connected	Established	172.16.254.6	65006	65003	3h 15m	22	13	471	483	default	Overlay

If BGP is not established, go back and validate the underlay links and addressing. Also, ensure that the loopback addresses are correct. Loopback addresses must be pingable from other loopback addresses. For example, Dist1 can reach Core1 and Access1's loopback addresses once the underlay eBGP peering sessions are established.

```
mist@Dist1> ping 172.16.254.2 count 1
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=9.654 ms

--- 172.16.254.2 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 9.654/9.654/9.654/0.000 ms

(master:0)
mist@Dist1> ping 172.16.254.6 count 1
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=7.892 ms

--- 172.16.254.6 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 7.892/7.892/7.892/0.000 ms
```

NOTE: eBGP sessions are established between adjacent layers in the Campus Fabric IP Clos.

Let's verify the routes are established to the core and other devices across multiple paths. For example, Access1 and Access2 should leverage both paths through Dist1 and Dist2 to access Core1 and Core2's loopback addresses as well as each other's loopback addresses.

Access1: Loopback reachability to Core1 through Dist1 and Dist2

```
mist@Access1> show route forwarding-table destination 172.16.254.2
Routing table: default.inet
Internet:
Destination      Type RtRef Next hop          Type Index  NhRef Netif
172.16.254.2/32  user   0      10.255.240.12      ucst  1695    6 ge-0/0/36.0
                  10.255.240.16      ucst  1696    6 ge-0/0/37.0
```

Access1: Loopback reachability with Core2 through Dist1 and Dist2

```
mist@Access1> show route forwarding-table destination 172.16.254.1
Routing table: default.inet
Internet:
Destination      Type RtRef Next hop          Type Index  NhRef Netif
172.16.254.1/32  user  0      10.255.240.12      ucst  1695   6 ge-0/0/36.0
                  10.255.240.16      ucst  1696   6 ge-0/0/37.0
```

Access1: Loopback reachability with Access2 through Dist1 and Dist2

```
mist@Access1> show route forwarding-table destination 172.16.254.5
Routing table: default.inet
Internet:
Destination      Type RtRef Next hop          Type Index  NhRef Netif
172.16.254.5/32  user  0      10.255.240.12      ucst  1695   6 ge-0/0/36.0
                  10.255.240.16      ucst  1696   6 ge-0/0/37.0
```

This can be repeated for Access 2 and so forth to verify ECMP load balancing.

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

EVPN VXLAN Verification Between Access and Core Switches

Since the desktop can ping its default gateway, we can assume the Ethernet switching tables are correctly populated, and that VLANs and interface mode are correct. If pinging the default gateway failed, then troubleshoot underlay connectivity.

Verification of the EVPN Database on Both Access Switches

```
mist@Access1> show evpn database
Instance: default-switch
VLAN  DomainId  MAC address      Active source      Timestamp          IP address
10001 00:cc:34:f3:cf:00  irb.0             Sep 06 11:15:21
10001 52:54:00:83:56:f2  ge-0/0/14.0       Sep 06 14:59:55
10001 ac:1f:6b:02:b0:c9  ge-0/0/14.0       Sep 06 14:57:40
11033 00:00:5e:e4:31:57  irb.1033          Sep 06 13:08:18  10.33.33.1
11033 5c:5b:35:2e:53:61  172.16.254.5      Sep 06 13:08:28
11033 5c:5b:35:af:29:d5  ge-0/0/16.0       Sep 06 13:08:21
11088 00:00:5e:e4:31:57  irb.1088          Sep 06 13:08:18  10.88.88.1
11088 52:54:00:7b:b4:52  172.16.254.5      Sep 06 14:18:00  10.88.88.88
11099 00:00:5e:e4:31:57  irb.1099          Sep 06 13:08:18  10.99.99.1
11099 52:54:00:c6:8a:58  ge-0/0/11.0       Sep 06 14:18:00  10.99.99.99

{master:0}
mist@Access1> show evpn database | match 52:54:00:c6:8a:58
11099 52:54:00:c6:8a:58  ge-0/0/11.0       Sep 06 14:18:00  10.99.99.99
```

You can view the entire database or search by MAC address.

```
mist@Access2> show evpn database
Instance: default-switch
VLAN DomainId MAC address Active source Timestamp IP address
10001 00:cc:34:f3:cf:00 irb.0 Sep 06 11:15:22
10001 52:54:00:83:56:f2 172.16.254.6 Sep 06 11:15:32
11033 00:00:5e:e4:31:57 irb.1033 Sep 06 13:08:20 10.33.33.1
11033 5c:5b:35:2e:53:61 ge-0/0/16.0 Sep 06 13:08:28
11033 5c:5b:35:af:29:d5 172.16.254.6 Sep 06 13:08:21
11088 00:00:5e:e4:31:57 irb.1088 Sep 06 13:08:20 10.88.88.1
11088 52:54:00:7b:b4:52 ge-0/0/12.0 Sep 06 14:18:00 10.88.88.88
11099 00:00:5e:e4:31:57 irb.1099 Sep 06 13:08:20 10.99.99.1
11099 52:54:00:c6:8a:58 172.16.254.6 Sep 06 14:18:01 10.99.99.99

{master:0}
mist@Access2> show evpn database | match 52:54:00:c6:8a:58
11099 52:54:00:c6:8a:58 172.16.254.6 Sep 06 14:18:01 10.99.99.99
```

Both access switches have identical EVPN databases, which is expected. Notice the entries for Desktop1 (10.99.99.99) and Desktop2 (10.88.88.88) present in each access switch. These entries are learned locally or through the campus fabric as shown in the Active Source column of the output.

10.99.99.99 is associated with irb.1099 and we see VNI of 11099. Let's just double-check VLAN-VNI mapping on the access and core switches.

Access

```
mist@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
```

Core

```
mist@Core1> show configuration | display set | match 1099
mist@Core1> █
```

Verification of VXLAN Tunnelling Between Access and Core Devices

Access1:

```
mist@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> 0 172.16.254.6 lo0.0 0
RVTEP-IP L2-RTT IFL-Idx Interface NH-Id RVTEP-Mode ELP-IP Flags
172.16.254.5 default-switch 558 vtep.32769 1697 RNVE
```

Access 2:

```
mist@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>
RVTEP-IP                L2-RTT
172.16.254.6            default-switch      609  vtep.32769  1739  RNVE
```

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

NOTE: Neither access switch displays Core1 or Core2 as remote tunnel destinations. The reason is that we have not yet configured the uplinks to the WAN router on top of the fabric. Hence no core switch knows about VLAN1099.

External Campus Fabric Connectivity Through the Border Gateway Core EX9204 Switches

There are several options available for attaching a WAN router to a campus fabric:

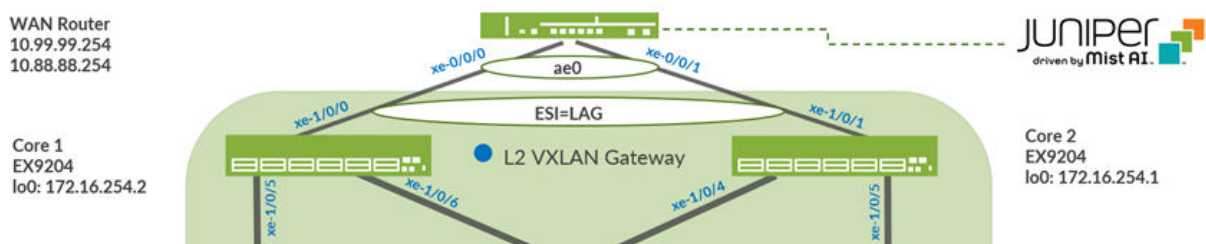
- Using a L2 forwarding:
 - With this method, the fabric uplinks are configured as ESI-LAG and contain one or more tagged VLANs (one for each VRF) to communicate with the WAN router.
 - You must configure the IP address of the WAN router interface manually as the next-hop IP address for default forwarding on each fabric VRF as already done above.
 - The WAN router must understand standard IEEE 802.3ad LAG with active LACP.
 - If you have more than one WAN router attached for redundancy, then we recommend that you configure VRRP between them for the interface IP addresses towards the Fabric.
- Using a L3 forwarding:
 - Configure the fabric uplinks as L3 peer-to-peer IP links.
 - You must establish a peer-to-peer link, per fabric VRF, with the WAN router.
 - Usually, there are multiple peer-to-peer links on a single physical uplink. The peer-to-peer links are further segmented using tagged VLANs to provide isolation on the uplinks.
 - There is no need to manually configure the next hops for each VRF inside the fabric since it is assumed that the propagation of the default gateways will be obtained for the WAN router via a routing protocol.

- Between the Fabric and the WAN router, a routing protocol must be established to exchange routes. Campus fabric supports external BGP and OSPF as routing protocols towards the WAN router.

For simplicity, we have chosen to utilize the L2 exit via ESI-LAG in this JVD. However, this is not the recommended method for large fabrics since IP Clos is intended to grow. For IP Clos fabrics in production environments, you should consider using BGP as the route exchange protocol as well as a redundant pair of WAN routers.

Remember that you chose to leverage the BGP capability of the EX9204 switches during the Campus Fabric IP Clos deployment, as shown in [Figure 25 on page 60](#).

Figure 25: L2 ESI-LAG Supporting Active-Active Load Balancing



Mist enables the EX9204 Switch to translate between VXLAN traffic within the campus fabric and standard Ethernet switching for external connectivity, in this case, an MX router. Let's verify the Ethernet Segment Identifier (ESI) status on the core switches.

```
root@Core1> show lacp statistics interfaces
warning: lacp subsystem not running - not needed by configuration.
```

We must configure the ESI-LAG since Mist does not configure this automatically. Add a port profile on core switches interfaces facing the WAN router.

The following represents an existing port profile applied to each MX router facing an EX9204 switch port. (On the Core2 switch the Port is xe-1/0/1).

PORT CONFIGURATION

Port Profile Assignment
★ Site, Template, or System Defined

New Port Range ✓ ✕

Port Aggregation
 Disable LACP

AE Index (0 - 255)

ESI-LAG

Allow switch port operator to modify port profile
 Yes No

Port IDs
 ✕
(ge-0/0/1, ge-0/0/4, ge-0/1/1-23, etc)

Interface
 L2 interface L3 interface L3 sub-interfaces

Configuration Profile
 trunk ▼

Enable Dynamic Configuration
 Enable "Up/Down Port" Alert Type i
Manage Alert Types in [Alerts Page](#)

Save the configuration and then verify the changes on the core switch.

```
mist@Core1> show configuration interfaces ae0 | display set | display inheritance
set interfaces ae0 mtu 9014
set interfaces ae0 esi 00:11:00:00:00:01:00:01:02:00
set interfaces ae0 esi all-active
set interfaces ae0 aggregated-ether-options lACP active
set interfaces ae0 aggregated-ether-options lACP periodic fast
set interfaces ae0 aggregated-ether-options lACP system-id 00:00:00:31:57:00
set interfaces ae0 aggregated-ether-options lACP admin-key 0
set interfaces ae0 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae0 unit 0 family ethernet-switching vLAN members all

mist@Core1> show evpn database
Instance: evpn_vs
VLAN  DomainId  MAC address      Active source      Timestamp          IP address
10001  52:54:00:83:56:f2  172.16.254.6      Sep 06 16:54:57
11033  5c:5b:35:2e:53:61  172.16.254.5      Sep 06 16:54:57
11033  5c:5b:35:af:29:d5  172.16.254.6      Sep 06 16:54:57
11033  f0:1c:2d:c8:e8:f0  00:11:00:00:00:01:00:01:02:00  Sep 06 16:55:39  10.33.33.254
11088  52:54:00:7b:b4:52  172.16.254.5      Sep 06 16:54:57  10.88.88.88
11088  f0:1c:2d:c8:e8:f0  00:11:00:00:00:01:00:01:02:00  Sep 06 16:55:39  10.88.88.254
11099  52:54:00:c6:8a:58  172.16.254.6      Sep 06 16:54:57  10.99.99.99
11099  f0:1c:2d:c8:e8:f0  00:11:00:00:00:01:00:01:02:00  Sep 06 16:55:39  10.99.99.254
```

Note that LACP is up, and this infers there is an existing configuration on the MX router.

```

root@Core1> show lacp statistics interfaces
Aggregated interface: ae0
LACP Statistics:      LACP Rx      LACP Tx      Unknown Rx      Illegal Rx
xe-1/0/0              2165         2166         0               0

root@Core1> show lacp interfaces
Aggregated interface: ae0
LACP state:      Role   Exp  Def  Dist  Col  Syn  Aggr  Timeout  Activity
xe-1/0/0        Actor  No   No   Yes  Yes  Yes  Yes   Fast    Active
xe-1/0/0        Partner No   No   Yes  Yes  Yes  Yes   Fast    Active
LACP protocol:  Receive State  Transmit State      Mux State
xe-1/0/0        Current      Fast periodic      Collecting distributing

root@Core1>

```

Verify if Desktop1's MAC address is advertised via BGP:

```

mist@Access1> show route advertising-protocol bgp 172.16.254.3 evpn-mac-address 52:54:00:c6:8a:58 table bgp.evpn.0
-----
bgp.evpn.0: 27 destinations, 40 routes (27 active, 0 holddown, 0 hidden)
Prefix      Nexthop      MED      Lclpref      AS path
* 2:172.16.254.6:1::11099::52:54:00:c6:8a:58/304 MAC/IP
*      Self                      I
* 2:172.16.254.6:1::11099::52:54:00:c6:8a:58::10.99.99.99/304 MAC/IP
*      Self                      I

```

Verify if the route is received on the core.

```

mist@Core1> show route receive-protocol bgp 172.16.254.3 evpn-mac-address 52:54:00:c6:8a:58 table bgp.evpn.0
-----
bgp.evpn.0: 52 destinations, 92 routes (52 active, 0 holddown, 0 hidden)
Prefix      Nexthop      MED      Lclpref      AS path
2:172.16.254.6:1::11099::52:54:00:c6:8a:58/304 MAC/IP
      172.16.254.6                      65003 65006 I
2:172.16.254.6:1::11099::52:54:00:c6:8a:58::10.99.99.99/304 MAC/IP
      172.16.254.6                      65003 65006 I

```

Let's check to see if the core has Desktop1 MAC address.

```

mist@Core1> show evpn database | match 52:54:00:c6:8a:58
11099      52:54:00:c6:8a:58 172.16.254.6      Sep 06 16:54:57 10.99.99.99

```

Verify the MAC address mapped to the correct VTEP interface. This is on the core. You can also verify on an access switch.


```

mist@Core1> show route forwarding-table family ethernet-switching extensive destination 52:54:00:c6:8a:58
Routing table: evpn_vs.evpn-vxlan [Index 9]
Bridging domain: vlan1099.evpn-vxlan [Index 5]
VPLS:
Enabled protocols: Bridging, ACKed by all peers, EVPN VXLAN,

Destination: 52:54:00:c6:8a:58/48
Learn VLAN: 0           Route type: user
Route reference: 0      Route interface-index: 349
Multicast RPF nh index: 0
P2mpidx: 0
IFL generation: 167    Epoch: 0
Sequence Number: 0    Learn Mask: 0x40000000000000000000000000000000000000000000000000
L2 Flags: control_dyn
Flags: sent to PFE
Nexthop:
Next-hop type: composite      Index: 635      Reference: 8

```

```

mist@Core1> show route forwarding-table family ethernet-switching

```

```

Routing table: evpn_vs.evpn-vxlan
VPLS:
Destination      Type RtRef Next hop            Type Index   NhRef Netif
default          perm  0         dscd             533   1
vtep.32769       intf  0         comp             634   8
vtep.32770       intf  0         comp             635   8
vtep.32771       intf  0         comp             651   7
ae0.0            intf  0         ucst             570   8 ae0.0

```

```

Routing table: evpn_vs.evpn-vxlan
Bridging domain: vlan1099.evpn-vxlan
VPLS:
Enabled protocols: Bridging, ACKed by all peers, EVPN VXLAN,
Destination      Type RtRef Next hop            Type Index   NhRef Netif
52:54:00:c6:8a:58/48 user  0         comp             635   8
0x3000d/51       user  0         comp             650   2
f0:1c:2d:c8:e8:f0/48 user  0         ucst             570   8 ae0.0
0x3000e/51       user  0         comp             649   2
0x30008/51       user  0         comp             639   2

```

Finally, the VTEP interface is up and passing traffic.

```

mist@Core1> show interfaces vtep.32770
Logical interface vtep.32770 (Index 349) (SNMP ifIndex 578)
Flags: Up SNMP-Traps Encapsulation: ENET2
VXLAN Endpoint Type: Remote, VXLAN Endpoint Address: 172.16.254.6, L2 Routing Instance: evpn_vs, L3 Routing Instance: default

Input packets : 528
Output packets: 51
Protocol eth-switch, MTU: Unlimited
Flags: Trunk-Mode

```

Then, confirm the EVPN database now has the ESI entry. Back to Desktop1 to see if it can cross the fabric.

```

root@desktop1:~#
root@desktop1:~# ping 1.1 -c 2
PING 1.1 (1.0.0.1) 56(84) bytes of data.
64 bytes from 1.0.0.1: icmp_seq=1 ttl=52 time=2.11 ms
64 bytes from 1.0.0.1: icmp_seq=2 ttl=52 time=3.00 ms

--- 1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 2.110/2.553/2.997/0.443 ms

```

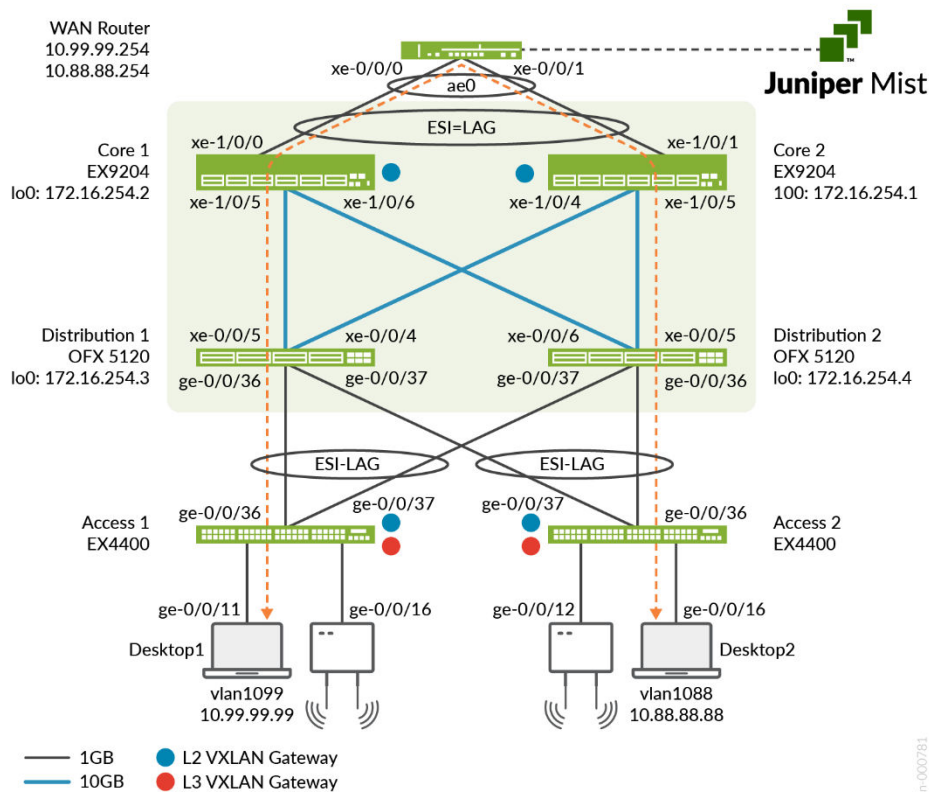
Last step is to verify Desktop1 can ping Desktop2.

```

root@desktop1:~# ping 10.88.88.88 -c 2
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=62 time=4.68 ms
64 bytes from 10.88.88.88: icmp_seq=2 ttl=62 time=0.590 ms

--- 10.88.88.88 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.590/2.635/4.681/2.045 ms
root@desktop1:~#

```



Meaning: Connectivity within the campus fabric and externally is verified. Desktops communicate with each other through the campus fabric, each in an isolated VRF, then forwarded to the MX router through the dual homing ESI-LAG on both Core1 and Core2 for routing between VRFs or routing instances. Internet connectivity was also verified from each desktop.

EVPN Insights

Mist Wired Assurance provides you with real-time status related to the health of the Campus Fabric IP Clos deployment using telemetry such as BGP neighbor status and TX/RX port statistics. The following screens are taken from the Campus Fabric IP Clos build by accessing **Organization > Wired > Campus Fabric** in the Mist Portal.

Neighbor Information 10:46 AM (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	10.255.240.9	65004	65002	21h 31m	5	4	2819	2845	default	Underlay
Connected	Established	10.255.240.7	65003	65002	21h 31m	5	3	2820	2845	default	Underlay
Connected	Established	172.16.254.4	65004	65002	21h 31m	39	34	5870	4222	default	Overlay
Connected	Established	172.16.254.3	65003	65002	21h 31m	39	29	5971	4205	default	Overlay

ID	IP Address	Name
1033	--	vlan1033
1088	--	vlan1088
1099	--	vlan1099

Switch	RX Bytes	TX Bytes	Link Status
> Dist1	143 MB	133.9 MB	Up
> Dist2	143.9 MB	152.9 MB	Up

← Campus Fabrics : Campus Fabric IPClos Edit Configuration Delete Connection

Dist1

MAC Address d8:53:9a:64:6f:c0
 Model QFX5120-48Y
 Status connected
 Site Primary Site
 Router ID 172.16.254.3

VLANs

ID	IP Address	Name
1033	--	vlan1033
1088	--	vlan1088
1099	--	vlan1099

Connections to Core

Switch	RX Bytes	TX Bytes	Link Status
Core2	278.5 MB	242 MB	Up
Core1	140.9 MB	240.8 MB	Up

Connections to Access

Switch	RX Bytes	TX Bytes	Link Status
Access2	98.5 MB	172.7 MB	Up
Access1	109.7 MB	318.3 MB	Up

[Remote Shell](#) [Insights](#) [Details](#)

Neighbor Information 10:54 AM (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	172.16.254.1	65001	65003	21h 37m	30	39	4418	5837	default	Overlay
Connected	Established	172.16.254.2	65002	65003	21h 36m	29	39	4218	5981	default	Overlay
Connected	Established	172.16.254.5	65005	65003	21h 36m	29	38	5115	5761	default	Overlay
Connected	Established	172.16.254.6	65006	65003	21h 37m	37	37	4988	5792	default	Overlay
Connected	Established	10.255.240.2	65001	65003	21h 37m	3	4	2858	2828	default	Underlay

NOTE: The full BGP peering table is not shown.

← Campus Fabrics : Campus Fabric IPClos Edit Configuration Delete Connection

Access1

MAC Address 00:cc:34:f4:72:00
 Model EX4400-48P
 Status connected
 Site Primary Site
 Router ID 172.16.254.6

VLANs

ID	IP Address	Name
1033	10.33.33.1	vlan1033
1088	10.88.88.1	vlan1088
1099	10.99.99.1	vlan1099

Connections to Distribution

Switch	RX Bytes	TX Bytes	Link Status
Dist1	318.1 MB	108.8 MB	Up
Dist2	229.5 MB	111.8 MB	Up

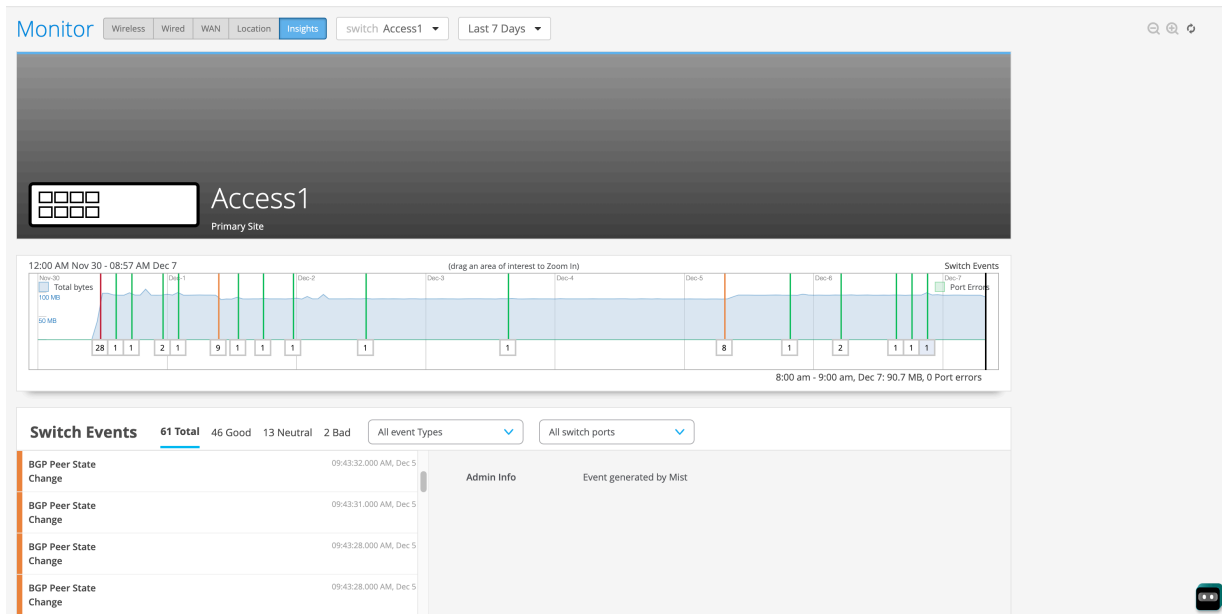
[Remote Shell](#) [Insights](#) [Details](#)

Neighbor Information 11:00 AM (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	10.255.240.16	65004	65006	21h 40m	5	5	2841	2867	default	Underlay
Connected	Established	10.255.240.12	65003	65006	21h 40m	5	2	2839	2863	default	Underlay
Connected	Established	172.16.254.3	65003	65006	21h 40m	37	35	5822	5007	default	Overlay
Connected	Established	172.16.254.4	65004	65006	21h 40m	37	30	5687	5117	default	Overlay

From this view, Mist also provides remote accessibility into each device's console through the remote shell utility as well as rich telemetry through the Switch Insights page. The remote shell utility is demonstrated throughout this document when displaying the real-time operational status of each device during the verification stage.

Switch Insights of the Access1 switch displays historical telemetry including BGP peering status critical to the health of the campus fabric:



Summary

Mist Campus Fabric provides an easy method to build IP Clos to enable EVPN-VXLAN overlay networks. This can be done only via Mist GUI. Steps are added to this document to help you understand the troubleshooting steps if deployment isn't working correctly.

Additional Information

IN THIS SECTION

- Configuration of the Underlay IP Fabric | 68
- Configuration of the EVPN VXLAN Overlay and Virtual Networks | 75
- Configuration of the L2 ESI-LAG Between the Core Switches and MX Router | 84

Configuration of the Underlay IP Fabric

This section displays the configuration output from the Juniper Mist cloud for the IP Fabric underlay on the core, distribution, and access switches using eBGP.

Mist provides the user with the following options (default in parenthesis):

- BGP Local AS (65001)
- Loopback Pool (172.16.254.0/23)
- Subnet (10.255.240.0/20) – point to point interfaces between adjacent layers

Mist enables per-packet load-balancing using ECMP and fast convergence of BGP in the event of a link or node failure using BFD.

Core1 Switch Configuration

1. Interconnects between the two distribution switches.

```
set interfaces xe-1/0/5 description evpn_downlink-to-d8539a646fc0
set interfaces xe-1/0/5 unit 0 family inet address 10.255.240.6/31
set interfaces xe-1/0/6 description evpn_downlink-to-d8539a64b5c0
set interfaces xe-1/0/6 unit 0 family inet address 10.255.240.8/31
```

2. 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
```

3. 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
```

4. BGP underlay network between the two distribution switches.

```
set protocols bgp group evpn_underlay type external
set protocols bgp group evpn_underlay log-updown
set protocols bgp group evpn_underlay import evpn_underlay_import
set protocols bgp group evpn_underlay family inet unicast
set protocols bgp group evpn_underlay authentication-key "xyz"
set protocols bgp group evpn_underlay export evpn_underlay_export
set protocols bgp group evpn_underlay local-as 65002
set protocols bgp group evpn_underlay multipath multiple-as
set protocols bgp group evpn_underlay bfd-liveness-detection minimum-interval 350
set protocols bgp group evpn_underlay bfd-liveness-detection multiplier 3
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
```

Core2 Switch Configuration

1. Interconnects between the two distribution switches.

```
set interfaces xe-1/0/4 description evpn_downlink-to-d8539a646fc0
set interfaces xe-1/0/4 unit 0 family inet address 10.255.240.2/31
set interfaces xe-1/0/5 description evpn_downlink-to-d8539a64b5c0
set interfaces xe-1/0/5 unit 0 family inet address 10.255.240.4/31
```

2. Loopback interface address, 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
```

3. 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
```

4. BGP underlay network between the two distribution switches.

```
set protocols bgp group evpn_underlay type external
set protocols bgp group evpn_underlay log-updown
set protocols bgp group evpn_underlay import evpn_underlay_import
set protocols bgp group evpn_underlay family inet unicast
set protocols bgp group evpn_underlay authentication-key "xyz"
set protocols bgp group evpn_underlay export evpn_underlay_export
set protocols bgp group evpn_underlay local-as 65001
set protocols bgp group evpn_underlay multipath multiple-as
set protocols bgp group evpn_underlay bfd-liveness-detection minimum-interval 350
set protocols bgp group evpn_underlay bfd-liveness-detection multiplier 3
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
```

Dist1 Switch Configuration

1. Interconnects between the two core switches and the two access switches.

```
Core Interfaces:
set interfaces xe-0/0/4 description evpn_uplink-to-f4b52ff3f400
set interfaces xe-0/0/4 unit 0 family inet address 10.255.240.3/31
set interfaces xe-0/0/5 description evpn_uplink-to-f4b52ff40400
set interfaces xe-0/0/5 unit 0 family inet address 10.255.240.7/31
```


Access Interfaces:

```
set interfaces ge-0/0/36 description evpn_downlink-to-00cc34f47200
set interfaces ge-0/0/36 unit 0 family inet address 10.255.240.12/31
set interfaces ge-0/0/37 description evpn_downlink-to-00cc34f3cf00
set interfaces ge-0/0/37 unit 0 family inet address 10.255.240.10/31
```

2. Loopback interface address, 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
```

3. 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
```

4. BGP underlay network between the two core switches and two access switches.

```
set protocols bgp group evpn_underlay type external
set protocols bgp group evpn_underlay log-updown
set protocols bgp group evpn_underlay import evpn_underlay_import
set protocols bgp group evpn_underlay family inet unicast
set protocols bgp group evpn_underlay authentication-key "xyz"
set protocols bgp group evpn_underlay export evpn_underlay_export
set protocols bgp group evpn_underlay local-as 65003
set protocols bgp group evpn_underlay multipath multiple-as
set protocols bgp group evpn_underlay bfd-liveness-detection minimum-interval 350
set protocols bgp group evpn_underlay bfd-liveness-detection multiplier 3
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
```

Dist2 Switch Configuration

1. Interconnects between the two core switches and the two access switches.

Core Interfaces:

```
set interfaces xe-0/0/5 description evpn_uplink-to-f4b52ff3f400
set interfaces xe-0/0/5 unit 0 family inet address 10.255.240.5/31
set interfaces xe-0/0/6 description evpn_uplink-to-f4b52ff40400
set interfaces xe-0/0/6 unit 0 family inet address 10.255.240.9/31
```

Access Interfaces:

```
set interfaces ge-0/0/36 description evpn_downlink-to-00cc34f3cf00
set interfaces ge-0/0/36 unit 0 family inet address 10.255.240.14/31
set interfaces ge-0/0/37 description evpn_downlink-to-00cc34f47200
set interfaces ge-0/0/37 unit 0 family inet address 10.255.240.16/31
```

2. Loopback interface and router ID.

```
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
```

3. 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
```

4. BGP underlay network between the two core switches and two access switches.

```
set protocols bgp group evpn_underlay type external
set protocols bgp group evpn_underlay log-updown
set protocols bgp group evpn_underlay import evpn_underlay_import
set protocols bgp group evpn_underlay family inet unicast
set protocols bgp group evpn_underlay authentication-key "xyz"
set protocols bgp group evpn_underlay export evpn_underlay_export
set protocols bgp group evpn_underlay local-as 65004
set protocols bgp group evpn_underlay multipath multiple-as
set protocols bgp group evpn_underlay bfd-liveness-detection minimum-interval 350
```

```

set protocols bgp group evpn_underlay bfd-liveness-detection multiplier 3
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

```

Access1 Configuration

1. Interconnects between the two distribution switches.

```

set interfaces ge-0/0/36 description evpn_uplink-to-d8539a646fc0
set interfaces ge-0/0/36 unit 0 family inet address 10.255.240.13/31
set interfaces ge-0/0/37 description evpn_uplink-to-d8539a64b5c0
set interfaces ge-0/0/37 unit 0 family inet address 10.255.240.17/31

```

2. Loopback interface and router ID and AS

```

set groups top interfaces lo0 unit 0 family inet address 192.168.255.31/32
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

```

3. 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

```

4. BGP underlay network between the two distribution switches.

```

set protocols bgp group evpn_underlay type external
set protocols bgp group evpn_underlay log-updown
set protocols bgp group evpn_underlay import evpn_underlay_import
set protocols bgp group evpn_underlay family inet unicast
set protocols bgp group evpn_underlay authentication-key "xyz"
set protocols bgp group evpn_underlay export evpn_underlay_export

```

```

set protocols bgp group evpn_underlay local-as 65006
set protocols bgp group evpn_underlay multipath multiple-as
set protocols bgp group evpn_underlay bfd-liveness-detection minimum-interval 350
set protocols bgp group evpn_underlay bfd-liveness-detection multiplier 3
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 graceful-restart

```

Access2 Configuration

1. Interconnects between the two distribution switches.

```

set interfaces ge-0/0/36 description evpn_uplink-to-d8539a64b5c0
set interfaces ge-0/0/36 unit 0 family inet address 10.255.240.15/31
set interfaces ge-0/0/37 description evpn_uplink-to-d8539a646fc0
set interfaces ge-0/0/37 unit 0 family inet address 10.255.240.11/31

```

2. Loopback interface and router ID and AS

```

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

```

3. 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

```

4. BGP underlay network between the two distribution switches.

```

set protocols bgp group evpn_underlay type external
set protocols bgp group evpn_underlay log-updown
set protocols bgp group evpn_underlay import evpn_underlay_import
set protocols bgp group evpn_underlay family inet unicast
set protocols bgp group evpn_underlay authentication-key "xyz"
set protocols bgp group evpn_underlay export evpn_underlay_export

```

```

set protocols bgp group evpn_underlay local-as 65005
set protocols bgp group evpn_underlay multipath multiple-as
set protocols bgp group evpn_underlay bfd-liveness-detection minimum-interval 350
set protocols bgp group evpn_underlay bfd-liveness-detection multiplier 3
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 graceful-restart

```

Configuration of the EVPN VXLAN Overlay and Virtual Networks

This section displays the Juniper Mist cloud configuration output for the EVPN VXLAN overlay on the core, distribution, and access switches using eBGP.

Mist enables load balancing across the overlay network and fast convergence of BGP in the event of a link or node failure using BFD between adjacent layers.

Mist enables VXLAN tunneling, VLAN to VXLAN mapping, and MP-BGP configuration snippets such as vrf-targets on the access layer switches. The core switches have VXLAN tunneling and VLAN to VXLAN mapping enabled based on the selection of the Core as a Border option.

Core1 Switch Configuration

1. BGP Overlay peering between the two distribution switches.

```

set protocols bgp group evpn_overlay type external
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 local-address 172.16.254.2
set protocols bgp group evpn_overlay log-updown
set protocols bgp group evpn_overlay family evpn signaling loops 2
set protocols bgp group evpn_overlay authentication-key "xyz"
set protocols bgp group evpn_overlay local-as 65002
set protocols bgp group evpn_overlay multipath multiple-as
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 neighbor 172.16.254.3 peer-as 65003
set protocols bgp group evpn_overlay neighbor 172.16.254.4 peer-as 65004

```

2. Switch options that define vrf-targets and the source loopback interface used for VXLAN.

```
set groups top routing-instances evpn_vs vtep-source-interface lo0.0
set groups top routing-instances evpn_vs route-distinguisher 172.16.254.2:1
set groups top routing-instances evpn_vs vrf-target target:65000:1
```

3. VXLAN encapsulation.

```
set groups top routing-instances evpn_vs protocols evpn encapsulation vxlan
set groups top routing-instances evpn_vs protocols evpn default-gateway no-gateway-community
set groups top routing-instances evpn_vs protocols evpn extended-vni-list all
```

4. VRFs that are used for traffic isolation.

```
set groups top routing-instances evpn_vs instance-type virtual-switch
set groups top routing-instances evpn_vs protocols evpn encapsulation vxlan
set groups top routing-instances evpn_vs protocols evpn default-gateway do-not-advertise
set groups top routing-instances evpn_vs protocols evpn extended-vni-list all
set groups top routing-instances evpn_vs protocols rstp interface ae0 disable
set groups top routing-instances evpn_vs protocols rstp bpd-block-on-edge
set groups top routing-instances evpn_vs vtep-source-interface lo0.0
set groups top routing-instances evpn_vs interface ae0.0
set groups top routing-instances evpn_vs route-distinguisher 172.16.254.2:1
set groups top routing-instances evpn_vs vrf-target target:65000:1
```

5. VLAN to VXLAN mapping.

```
set groups top routing-instances evpn_vs vlans vlan1033 vlan-id 1033
set groups top routing-instances evpn_vs vlans vlan1033 vxlan vni 11033
set groups top routing-instances evpn_vs vlans vlan1088 vlan-id 1088
set groups top routing-instances evpn_vs vlans vlan1088 vxlan vni 11088
set groups top routing-instances evpn_vs vlans vlan1099 vlan-id 1099
set groups top routing-instances evpn_vs vlans vlan1099 vxlan vni 11099
```

Core2 Configuration

1. BGP overlay peering between the two distribution switches.

```

set protocols bgp group evpn_overlay type external
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 local-address 172.16.254.1
set protocols bgp group evpn_overlay log-updown
set protocols bgp group evpn_overlay family evpn signaling loops 2
set protocols bgp group evpn_overlay authentication-key "xyz"
set protocols bgp group evpn_overlay local-as 65001
set protocols bgp group evpn_overlay multipath multiple-as
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 neighbor 172.16.254.3 peer-as 65003
set protocols bgp group evpn_overlay neighbor 172.16.254.4 peer-as 65004

```

2. Switch options that define vrf-targets and the source loopback interface used for VXLAN.

```

set groups top routing-instances evpn_vs vtep-source-interface lo0.0
set groups top routing-instances evpn_vs route-distinguisher 172.16.254.1:1
set groups top routing-instances evpn_vs vrf-target target:65000:1

```

3. VXLAN encapsulation.

```

set groups top routing-instances evpn_vs protocols evpn encapsulation vxlan
set groups top routing-instances evpn_vs protocols evpn default-gateway no-gateway-community
set groups top routing-instances evpn_vs protocols evpn extended-vni-list all

```

4. VRFs that are used for traffic isolation.

```

set groups top routing-instances evpn_vs instance-type virtual-switch
set groups top routing-instances evpn_vs protocols evpn encapsulation vxlan
set groups top routing-instances evpn_vs protocols evpn default-gateway do-not-advertise
set groups top routing-instances evpn_vs protocols evpn extended-vni-list all
set groups top routing-instances evpn_vs protocols rstp interface ae0 disable
set groups top routing-instances evpn_vs protocols rstp bpdu-block-on-edge
set groups top routing-instances evpn_vs vtep-source-interface lo0.0
set groups top routing-instances evpn_vs interface ae0.0

```

```
set groups top routing-instances evpn_vs route-distinguisher 172.16.254.1:1
set groups top routing-instances evpn_vs vrf-target target:65000:1
```

5. VLAN to VXLAN mapping.

```
set groups top routing-instances evpn_vs vlans vlan1033 vlan-id 1033
set groups top routing-instances evpn_vs vlans vlan1033 vxlan vni 11033
set groups top routing-instances evpn_vs vlans vlan1088 vlan-id 1088
set groups top routing-instances evpn_vs vlans vlan1088 vxlan vni 11088
set groups top routing-instances evpn_vs vlans vlan1099 vlan-id 1099
set groups top routing-instances evpn_vs vlans vlan1099 vxlan vni 11099
```

Dist1 Switch Configuration

1. BGP overlay peering between the two core switches and the two access switches.

```
set protocols bgp group evpn_overlay type external
set protocols bgp group evpn_overlay multihop ttl 1
set protocols bgp group evpn_overlay multihop no-next-hop-change
set protocols bgp group evpn_overlay local-address 172.16.254.3
set protocols bgp group evpn_overlay log-updown
set protocols bgp group evpn_overlay family evpn signaling loops 2
set protocols bgp group evpn_overlay authentication-key "xyz"
set protocols bgp group evpn_overlay local-as 65003
set protocols bgp group evpn_overlay multipath multiple-as
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 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
```

Dist2 Switch Configuration

1. BGP overlay peering between the two core switches and the two access switches.

```
set protocols bgp group evpn_overlay type external
set protocols bgp group evpn_overlay multihop ttl 1
set protocols bgp group evpn_overlay multihop no-next-hop-change
set protocols bgp group evpn_overlay local-address 172.16.254.4
```



```

set protocols bgp group evpn_overlay log-updown
set protocols bgp group evpn_overlay family evpn signaling loops 2
set protocols bgp group evpn_overlay authentication-key "xyz"
set protocols bgp group evpn_overlay local-as 65004
set protocols bgp group evpn_overlay multipath multiple-as
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 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

```

Access1 Configuration

1. BGP overlay peering between the two distribution switches.

```

set protocols bgp group evpn_overlay type external
set protocols bgp group evpn_overlay multihop ttl 1
set protocols bgp group evpn_overlay multihop no-next-hop-change
set protocols bgp group evpn_overlay local-address 172.16.254.6
set protocols bgp group evpn_overlay log-updown
set protocols bgp group evpn_overlay family evpn signaling loops 2
set protocols bgp group evpn_overlay authentication-key "xyz"
set protocols bgp group evpn_overlay local-as 65006
set protocols bgp group evpn_overlay multipath multiple-as
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 neighbor 172.16.254.3 peer-as 65003
set protocols bgp group evpn_overlay neighbor 172.16.254.4 peer-as 65004

```

2. Switch options that define vrf-targets and the source loopback interface used for VXLAN.

```

set groups top switch-options vtep-source-interface lo0.0
set groups top switch-options route-distinguisher 172.16.254.6:1
set groups top switch-options vrf-target target:65000:1

```

3. VXLAN encapsulation.

```
set groups top protocols evpn encapsulation vxlan
set groups top protocols evpn default-gateway no-gateway-community
set groups top protocols evpn extended-vni-list all
```

4. VRFs that are used for traffic isolation.

```
set groups top routing-instances guest-wifi instance-type vrf
set groups top routing-instances guest-wifi routing-options static route 0.0.0.0/0 next-hop
10.33.33.254
set groups top routing-instances guest-wifi routing-options multipath
set groups top routing-instances guest-wifi routing-options auto-export
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 interface irb.1033
set groups top routing-instances guest-wifi interface lo0.3
set groups top routing-instances guest-wifi route-distinguisher 172.16.254.6:103
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 developers instance-type vrf
set groups top routing-instances developers routing-options static route 0.0.0.0/0 next-hop
10.88.88.254
set groups top routing-instances developers routing-options multipath
set groups top routing-instances developers routing-options auto-export
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 interface irb.1088
set groups top routing-instances developers interface lo0.2
set groups top routing-instances developers route-distinguisher 172.16.254.6:102
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 corp-it instance-type vrf
set groups top routing-instances corp-it routing-options static route 0.0.0.0/0 next-hop
10.99.99.254
```

```

set groups top routing-instances corp-it routing-options multipath
set groups top routing-instances corp-it routing-options auto-export
set groups top routing-instances corp-it protocols evpn ip-prefix-routes advertise direct-
nextthop
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 interface irb.1099
set groups top routing-instances corp-it interface lo0.1
set groups top routing-instances corp-it route-distinguisher 172.16.254.6:101
set groups top routing-instances corp-it vrf-target target:65000:101
set groups top routing-instances corp-it vrf-table-label

```

5. VLAN to VXLAN 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

```

6. L3 IRB interface enablement with anycast addressing.

```

set interfaces irb unit 1033 description vlan1033
set interfaces irb unit 1033 family inet mtu 9000
set interfaces irb unit 1033 family inet address 10.33.33.1/24
set interfaces irb unit 1033 mac 00:00:5e:e4:31:57
set interfaces irb unit 1088 description vlan1088
set interfaces irb unit 1088 family inet mtu 9000
set interfaces irb unit 1088 family inet address 10.88.88.1/24
set interfaces irb unit 1088 mac 00:00:5e:e4:31:57
set interfaces irb unit 1099 description vlan1099
set interfaces irb unit 1099 family inet mtu 9000
set interfaces irb unit 1099 family inet address 10.99.99.1/24
set interfaces irb unit 1099 mac 00:00:5e:e4:31:57

```

Access2 Switch Configuration

1. BGP overlay peering between the two distribution switches.

```

set protocols bgp group evpn_overlay type external
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 local-address 172.16.254.5
set protocols bgp group evpn_overlay log-updown
set protocols bgp group evpn_overlay family evpn signaling loops 2
set protocols bgp group evpn_overlay authentication-key "xyz"
set protocols bgp group evpn_overlay local-as 65005
set protocols bgp group evpn_overlay multipath multiple-as
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 neighbor 172.16.254.3 peer-as 65003
set protocols bgp group evpn_overlay neighbor 172.16.254.4 peer-as 65004

```

2. Switch options that define vrf-targets and the source loopback interface used for VXLAN.

```

set groups top switch-options vtep-source-interface lo0.0
set groups top switch-options route-distinguisher 172.16.254.5:1
set groups top switch-options vrf-target target:65000:1

```

3. VXLAN encapsulation.

```

set groups top protocols evpn encapsulation vxlan
set groups top protocols evpn default-gateway no-gateway-community
set groups top protocols evpn extended-vni-list all

```

4. VRFs that are used for traffic isolation.

```

set groups top routing-instances guest-wifi instance-type vrf
set groups top routing-instances guest-wifi routing-options static route 0.0.0.0/0 next-hop
10.33.33.254
set groups top routing-instances guest-wifi routing-options multipath
set groups top routing-instances guest-wifi routing-options auto-export
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 interface irb.1033
set groups top routing-instances guest-wifi interface lo0.3
set groups top routing-instances guest-wifi route-distinguisher 172.16.254.5:103
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 developers instance-type vrf
set groups top routing-instances developers routing-options static route 0.0.0.0/0 next-hop
10.88.88.254
set groups top routing-instances developers routing-options multipath
set groups top routing-instances developers routing-options auto-export
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 interface irb.1088
set groups top routing-instances developers interface lo0.2
set groups top routing-instances developers route-distinguisher 172.16.254.5:102
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 corp-it instance-type vrf
set groups top routing-instances corp-it routing-options static route 0.0.0.0/0 next-hop
10.99.99.254
set groups top routing-instances corp-it routing-options multipath
set groups top routing-instances corp-it routing-options auto-export
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 interface irb.1099
set groups top routing-instances corp-it interface lo0.1
set groups top routing-instances corp-it route-distinguisher 172.16.254.5:101
set groups top routing-instances corp-it vrf-target target:65000:101
set groups top routing-instances corp-it vrf-table-label

```

5. VLAN to VXLAN 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

```

6. L3 IRB interface enablement with anycast addressing.

```

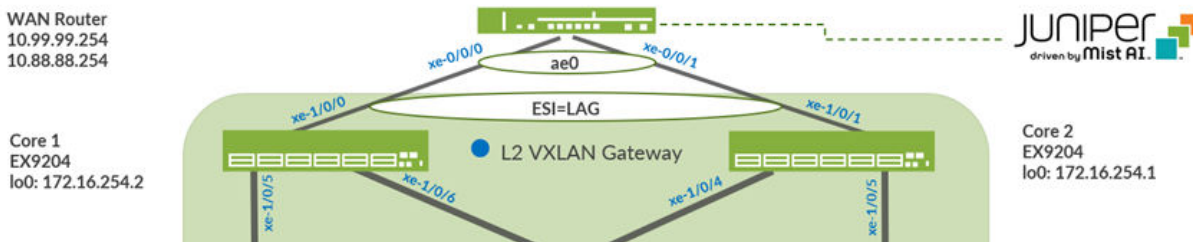
set interfaces irb unit 1033 description vlan1033
set interfaces irb unit 1033 family inet mtu 9000
set interfaces irb unit 1033 family inet address 10.33.33.1/24
set interfaces irb unit 1033 mac 00:00:5e:e4:31:57
set interfaces irb unit 1088 description vlan1088
set interfaces irb unit 1088 family inet mtu 9000
set interfaces irb unit 1088 family inet address 10.88.88.1/24
set interfaces irb unit 1088 mac 00:00:5e:e4:31:57
set interfaces irb unit 1099 description vlan1099
set interfaces irb unit 1099 family inet mtu 9000
set interfaces irb unit 1099 family inet address 10.99.99.1/24
set interfaces irb unit 1099 mac 00:00:5e:e4:31:57

```

Configuration of the L2 ESI-LAG Between the Core Switches and MX Router

This section displays the Juniper Mist Cloud configuration output for the enablement of the L2 ESI Link Aggregation Groups (LAG) between the core switches and MX routers. This Mist profile enables all VLANs on the Ethernet bundle with requisite ESI and LACP configuration options. From the perspective of the MX router, the Ethernet bundle that is configured on the MX router views the ESI-LAG as a single MAC address with the same LACP system id. This enables load hashing between the core and MX router without requiring L2 loop-free detection protocols such as RSTP.

Figure 26: L2 ESI-LAG Supporting Active-Active Load Balancing



Core 1 Switch Configuration

1. Interface association with the newly created Ethernet bundle that includes ESI and LACP configuration.

```

set interfaces xe-1/0/0 hold-time up 120000
set interfaces xe-1/0/0 hold-time down 1
set interfaces xe-1/0/0 ether-options 802.3ad ae0
set interfaces xe-1/0/0 unit 0 family ethernet-switching storm-control default
set groups myesilag interfaces <*> mtu 9014
set groups myesilag interfaces <*> unit 0 family ethernet-switching interface-mode trunk
set groups myesilag interfaces <*> unit 0 family ethernet-switching vlan members all
set interfaces ae0 apply-groups myesilag
set interfaces ae0 esi 00:11:00:00:00:01:00:01:02:00
set interfaces ae0 esi all-active
set interfaces ae0 aggregated-ether-options lACP active
set interfaces ae0 aggregated-ether-options lACP periodic fast
set interfaces ae0 aggregated-ether-options lACP system-id 00:00:00:31:57:00
set interfaces ae0 aggregated-ether-options lACP admin-key 0

```

Core 2 Switch Configuration

1. Interface association with the newly created Ethernet bundle that includes ESI and LACP configuration.

```

set interfaces xe-1/0/1 hold-time up 120000
set interfaces xe-1/0/1 hold-time down 1
set interfaces xe-1/0/1 ether-options 802.3ad ae0
set interfaces xe-1/0/1 unit 0 family ethernet-switching storm-control default
set groups myesilag interfaces <*> mtu 9014
set groups myesilag interfaces <*> unit 0 family ethernet-switching interface-mode trunk
set groups myesilag interfaces <*> unit 0 family ethernet-switching vlan members all

```

```

set interfaces ae0 apply-groups myesilag
set interfaces ae0 esi 00:11:00:00:00:01:00:01:02:00
set interfaces ae0 esi all-active
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp periodic fast
set interfaces ae0 aggregated-ether-options lacp system-id 00:00:00:31:57:00
set interfaces ae0 aggregated-ether-options lacp admin-key 0

```

MX Router Configuration

1. Interface association with newly created Ethernet bundle and LACP configuration.

```

set interfaces xe-0/0/0 gigether-options 802.3ad ae0
set interfaces xe-0/0/1 gigether-options 802.3ad ae0
set chassis aggregated-devices ethernet device-count 1
set interfaces ae0 mtu 9014
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 unit 0 family bridge interface-mode trunk
set interfaces ae0 unit 0 family bridge vlan-id-list 1033
set interfaces ae0 unit 0 family bridge vlan-id-list 1088
set interfaces ae0 unit 0 family bridge vlan-id-list 1099
set interfaces irb unit 1033 family inet address 10.33.33.254/24
set interfaces irb unit 1088 family inet address 10.88.88.254/24
set interfaces irb unit 1099 family inet address 10.99.99.254/24
set bridge-domains vlan1033 vlan-id 1033
set bridge-domains vlan1033 routing-interface irb.1033
set bridge-domains vlan1088 vlan-id 1088
set bridge-domains vlan1088 routing-interface irb.1088
set bridge-domains vlan1099 vlan-id 1099
set bridge-domains vlan1099 routing-interface irb.1099

```

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 © 2024 Juniper Networks, Inc. All rights reserved.