

# Campus Fabric IP Clos Wired Assurance



Juniper Networks, Inc.  
1133 Innovation Way  
Sunnyvale, CA 94089 USA  
[www.juniper.net](http://www.juniper.net)

**Issue Date: 7 December 2022**



---

## Contents

Introduction to Campus Fabric IP Clos Wired Assurance .....	4
Use Case Overview .....	4
Benefits of Campus Fabric: IP Clos.....	4
Technical Overview .....	6
Understanding EVPN .....	6
Underlay Network .....	7
Overlay Network (Data Plane) .....	8
Resiliency and Load Balancing .....	9
Ethernet Segment Identifier (ESI).....	9
Services Block .....	9
Access Layer .....	10
Juniper Access points .....	10
Juniper Mist Wired Assurance .....	10
Campus IP Clos Fabric High Level Architecture .....	11
Campus Fabric IP Clos Components .....	11
Juniper Mist Wired Assurance .....	12
Juniper Mist Wired Assurance Switches Section.....	12
Topology .....	13
Create the Campus Fabric .....	13
Campus Fabric Org Build.....	14
Campus Fabric Site Build.....	14
Choose the campus fabric topology.....	14
Select campus fabric nodes .....	15
Configure Networks.....	16
Other IP Configuration.....	17
Configure campus fabric ports.....	19
Core Switches .....	19
Distribution Switches .....	20
Access Switches .....	21
Campus Fabric Configuration Confirmation .....	21
VERIFICATION .....	25
BGP Underlay.....	25
EVPN VXLAN verification between Access and Core switches.....	27
Verification of the EVPN Database on both access switches.....	27
Verification of VXLAN tunnelling between Access and Core devices .....	28
External Campus Fabric connectivity through the Border GW Core EX9204 switches .....	30
EVPN Insights.....	32
Summary .....	34
Appendix.....	35
Campus Fabric IP Clos Configurations .....	35
eBGP Underlay .....	35
eBGP Overlay in support of EVPN-VXLAN .....	35
VXLAN and EVPN Enablement.....	37
VLAN-VXLAN (VNI) mapping.....	37

---

Routing Policy:.....	38
L3 Interfaces (IRB) .....	38
Routing Instances .....	38
ESI-LAG between Core1/2 and SRX Firewall.....	39

---

## Introduction to Campus Fabric IP Clos Wired Assurance

### About This Network Configuration Example

This network configuration example (NCE) describes 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 as the overlay network, and eBGP in the underlay along with Juniper Mist Access Points integration.

### Use Case Overview

Enterprise networks are undergoing massive transitions to accommodate the growing demand for cloud-ready, scalable, and efficient networks, and the plethora of 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 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.

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/rfc7348>) that is common across campuses and data centers.

The Juniper campus architecture uses a Layer 3 IP-based underlay network and an EVPN-VXLAN overlay network. The simple IP-based Layer 3 network underlay limits the Layer 2 broadcast domain and eliminates the need for Spanning Tree Protocols (STP/RSTP). A flexible overlay network based on a VXLAN tunnels combined with an EVPN control plane efficiently provides Layer 3 or Layer 2 connectivity. This architecture decouples the virtual topology from the physical topology, which improves network flexibility and simplifies network management. Endpoints that require Layer 2 adjacency, such as IoT devices, can be placed anywhere in the network and remain connected to the same logical Layer 2 network.

With an EVPN-VXLAN campus architecture, you can easily add core, distribution, and access layer devices as your business grows without having to redesign the network. EVPN-VXLAN is vendor-agnostic, so you can use the existing access layer infrastructure and gradually migrate to access layer switches that support 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.

### Benefits of Campus Fabric: IP Clos

With increasing number of devices connecting to the network, you will need to scale your campus network rapidly without adding complexity. Many IoT devices have limited networking capabilities and require Layer 2 adjacency across buildings and campuses. Traditionally, this problem was solved by extending VLANs between endpoints using data plane-based flood and learn mechanisms inherent with ethernet switching technologies. The traditional ethernet switching approach is inefficient because it leverages broadcast and multicast technologies to announce MAC addresses. It is also difficult to manage because you need to manually configure 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 BGP or OSPF for route distribution and an overlay based on EVPN-VXLAN. Switches and router participating in EVPN-VXLAN function as VTEPs that encapsulate and decapsulate the VXLAN traffic. 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 by stretching the Layer 2 VXLAN network with routing occurring in the access device instead of the Core or Distribution layers. The IP Clos network encompasses the distribution, core, and access layers of your topology.

## Campus Fabric IP Clos

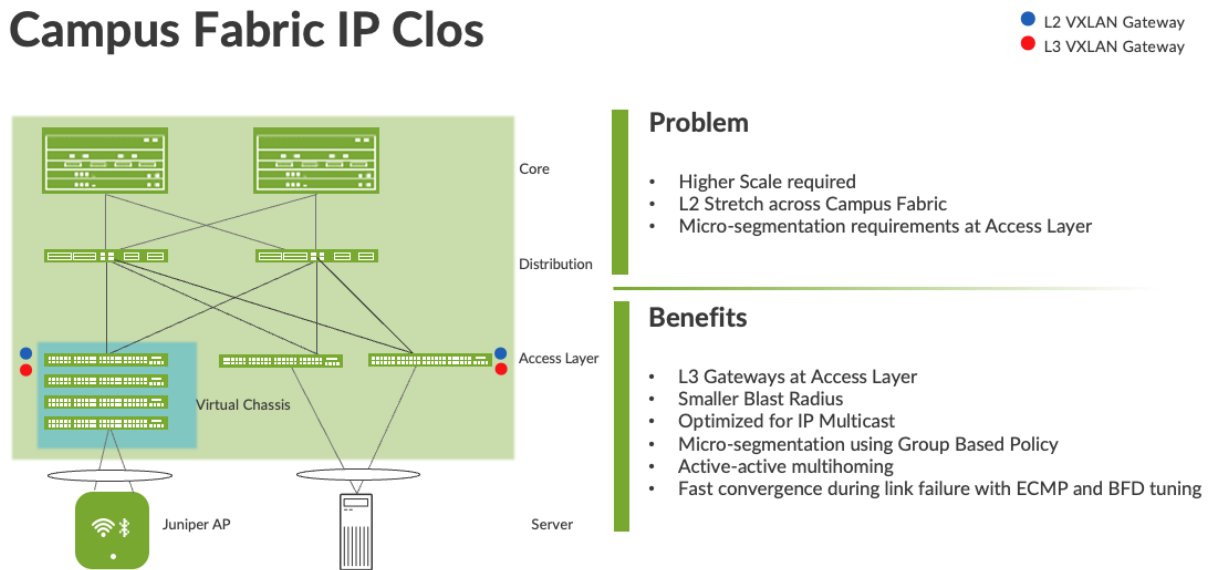


Figure 1 Campus fabric IP Clos

An EVPN-VXLAN fabric solves these issues and provides the following benefits:

- Reduced flooding and learning—Control plane-based Layer 2/Layer 3 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 leaving less bandwidth available for production traffic. The EVPN control plane handles the exchange and learning of MAC addresses through eBGP routing, rather than a Layer-2 forwarding plane.
- Scalability—More efficient control-plane based Layer 2/Layer 3 learning allows the EVPN-VXLAN network to scale up to support hundreds of thousands of endpoints.
- Consistency—A universal EVPN-VXLAN-based architecture across campuses and data-centers enables seamless end-to-end network for endpoints and applications.
- Group Based Policies - With GBT you can enable micro segmentation and macro segmentation with EVPN-VXLAN to minimize Layer 2 flooding, provide traffic isolation, and simplify the network.
- Location-agnostic connectivity—The EVPN-VXLAN campus architecture provides a consistent endpoint experience no matter where the endpoint is located. Some endpoints require Layer 2 reachability, such as legacy building security systems or IoT devices. VXLAN overlay provides Layer 2 extension across campuses without any changes to the underlay network.

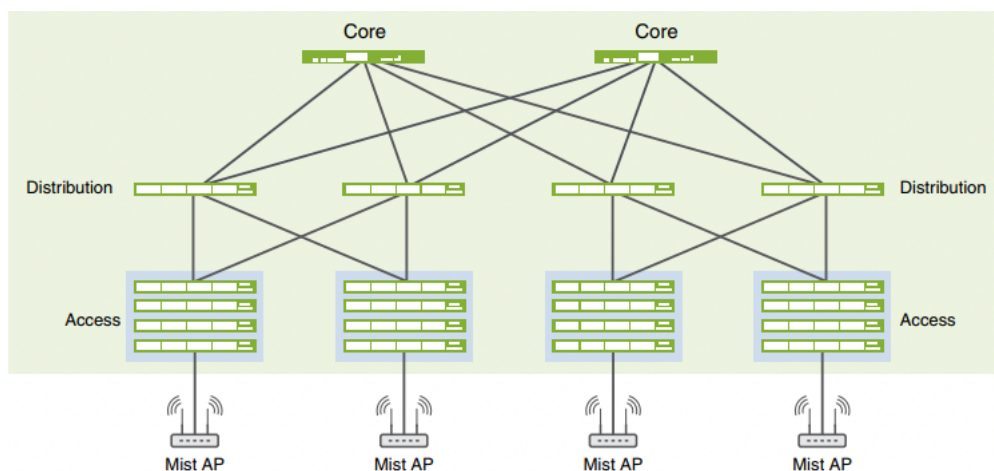


Figure 2 Sample topology

## Technical Overview

Network overlays enable connectivity and addressing independent of the physical network. Ethernet frames are wrapped in IP UDP datagrams that are themselves encapsulated into IP for transport over the underlay. VXLAN enables virtual Layer 2 subnets (or VLANs) to span underlying physical Layer 3 network.

In a VXLAN overlay network, each Layer 2 subnet or segment is uniquely identified by a virtual network identifier (VNI). A VNI segments traffic the same way that a VLAN ID does. 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 Layer-3 gateway. The entity that performs VXLAN encapsulation and decapsulation is called a VXLAN tunnel endpoint (VTEP). Each VXLAN Tunnel Endpoint (VTEP) is known as the Layer 2 Gateway and typically assigned with the device's Loopback address.

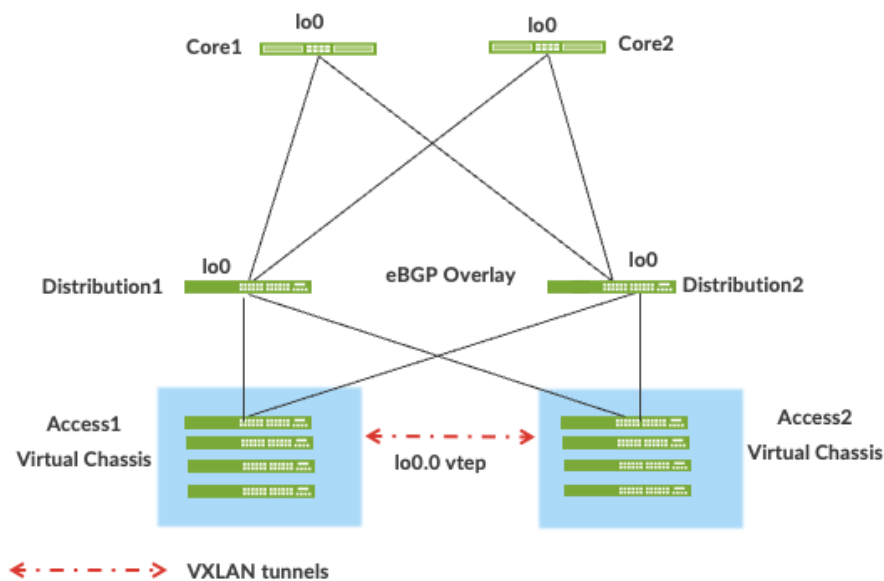


Figure 3. VXLAN VTEP tunnels

VXLAN can be deployed as a tunnelling protocol across a Layer 3 IP Campus Fabric without a control plane protocol. However, the use of VXLAN tunnels alone does not change the flood and learn behaviour of the Ethernet protocol.

The two primary methods for using VXLAN without a control plane protocol—static unicast VXLAN tunnels and VXLAN tunnels that are signalled with a multicast underlay—do not solve the inherent flood and learn problem and are difficult to scale in large multitenant environments. <https://www.rfc-editor.org/rfc/rfc7348>

## Understanding EVPN

Ethernet VPN (EVPN) is an 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 routes. EVPN enables devices acting as VTEPs to exchange reachability information with each other about their endpoints.

---

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 off the multihomed Access layer switches is called ESI-LAG; while the 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.

## Underlay Network

An EVPN-VXLAN fabric architecture makes the network infrastructure simple and consistent across campuses and data centers. All the core and distribution devices must be connected to each other using a Layer 3 infrastructure. Juniper recommends deploying a Clos-based IP fabric to ensure predictable performance and to enable a consistent, scalable architecture.

You could use any Layer 3 routing protocol to exchange loopback addresses between the access, core, and distribution devices. BGP provides benefits like 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 configuration for the underlay and overlay for only the campus fabric. There are options to provision additional BGP speakers to allow customers 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.

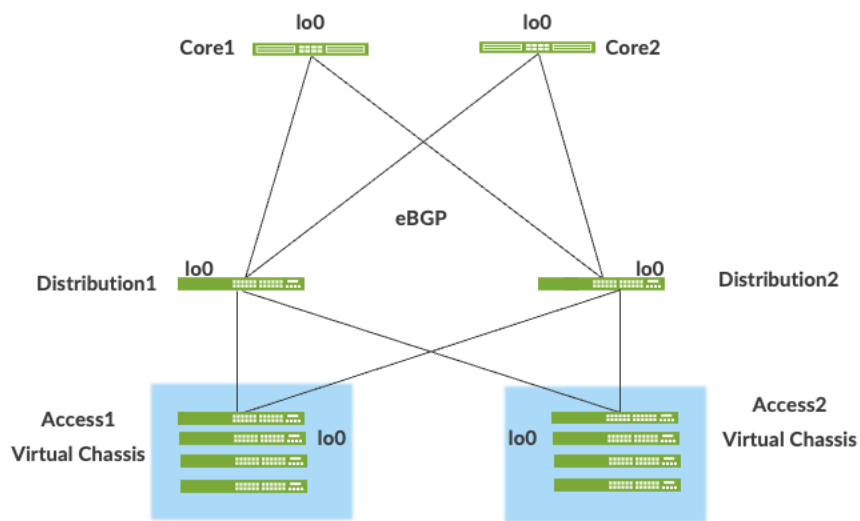


Figure 4. Pt-Pt /31 links between adjacent layers running eBGP

### Overlay Network (Data Plane)

VXLAN is the overlay data plane encapsulation protocol that tunnels Ethernet frames between network endpoints over the Layer 3 IP network. Devices that perform VXLAN encapsulation and decapsulation for the network are referred to as a VXLAN tunnel endpoint (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 packet for transmission to the remote VTEP over the IP fabric.

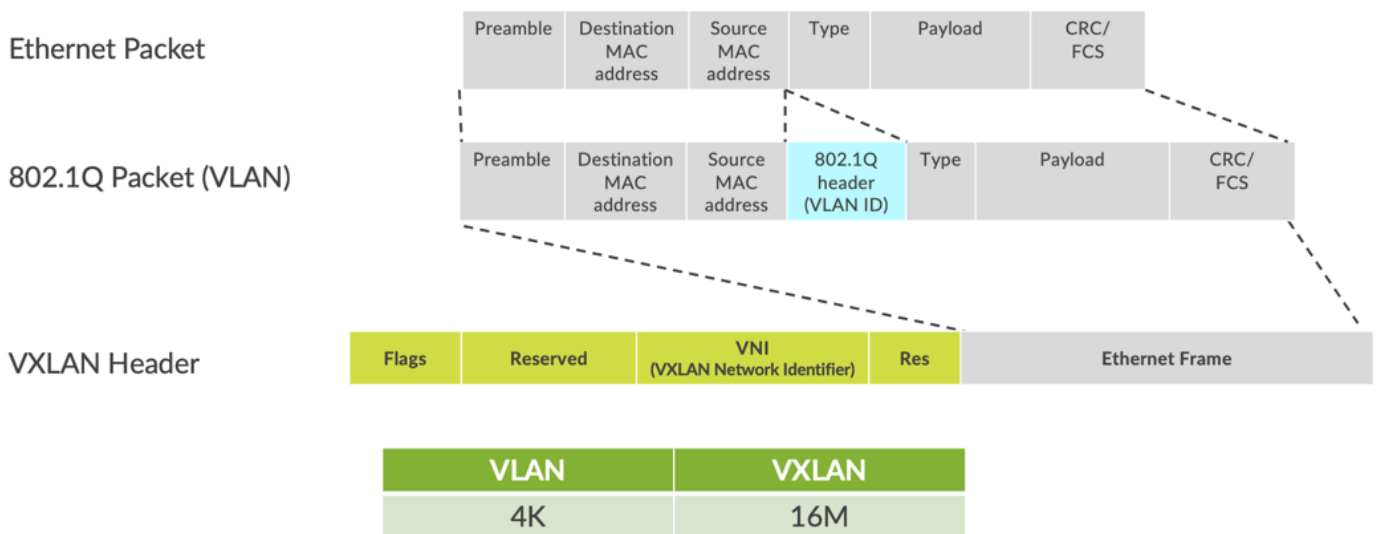


Figure 5 VXlan Header

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, to extend services to external routers or firewalls that exist outside of the Campus Fabric

### Overlay Network (Control Plane)

MP-BGP with EVPN signaling acts as the overlay control plane protocol. Adjacent switches peer using their loopback addresses using next-hops announced by the underlay BGP sessions. The core and distribution devices establish eBGP sessions between each other while the access and distribution establish eBGP sessions with each other. When there is a Layer 2 forwarding table update on any switch participating in campus fabric it will send a BGP update message with the new MAC route to other devices in the fabric. Those devices will then update their local evpn database and routing tables.



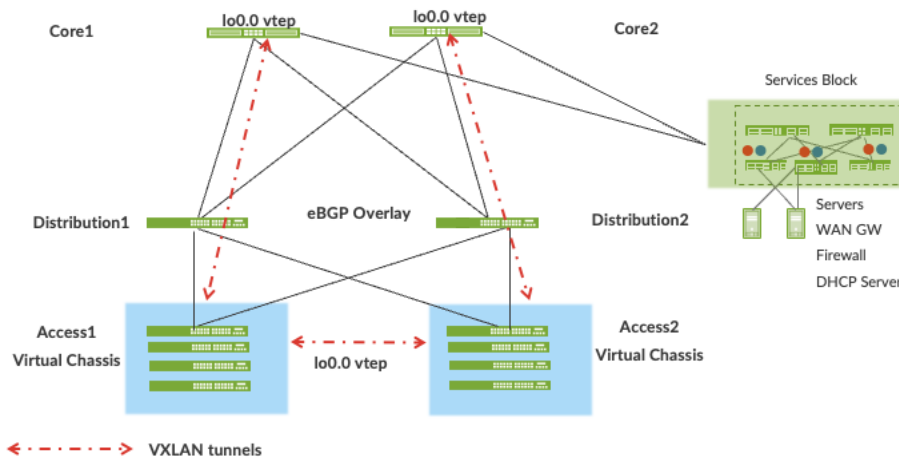


Figure 6

## Resiliency and Load Balancing

Juniper supports BFD, Bi-Directional Forwarding, in conjunction with the underlay and overlay networks. This provides fast convergence in the event of a device or link failure without relying on the routing protocol's timers. Mist configured BFD underlay and over minimum interval of 1000ms and 3000 ms in the underlay and overlay respectively. Load Balancing is supported across all links within the Campus Fabric using ECMP or Equal Cost Multi Pathing enabled at the forwarding plane.

## Ethernet Segment Identifier (ESI)

When switches have aggregated ethernet interfaces or LAG and multihomed to two or more switches an ESI is a 10-octet integer that identifies that segment. This enables link failover in the event of a bad link and is automatically assigned by Mist.

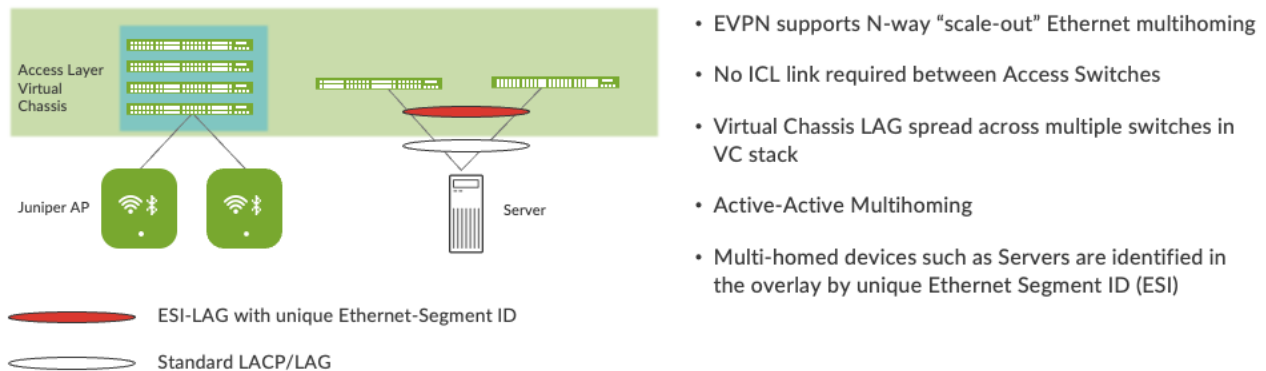


Figure 7. resiliency and load balancing

## Services Block

Customers may wish to position critical infrastructure services off a dedicated Access Pair of Juniper switches which could include WAN and Firewall connectivity. This Services Block Border is supported directly off the Core Layer platforms or as a dedicated pair of switches.

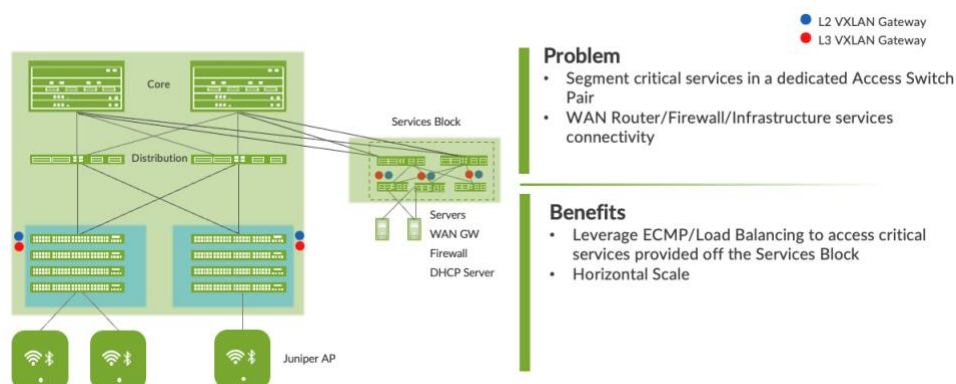
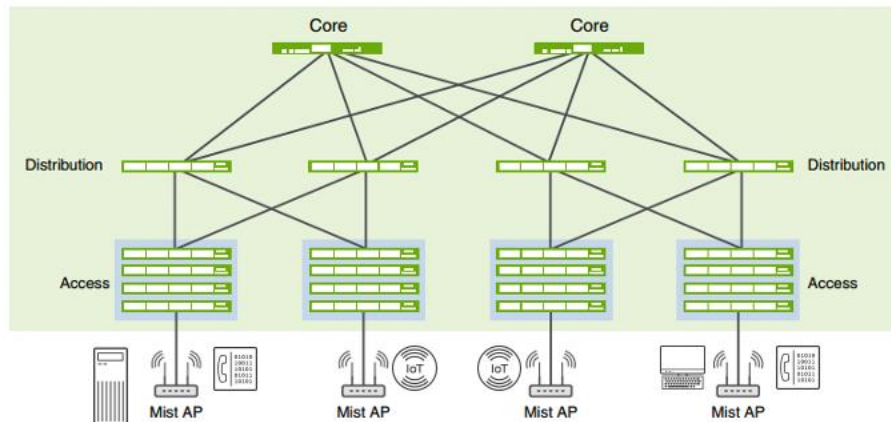


Figure 8. Services Block

## 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. In this IP Clos campus design, the EVPN-VXLAN network extends all the way to the access layer switches.



**Figure 9. End point 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 availability of fiber throughout the campus. The Virtual Chassis is also managed as a single device and supports up to 10 devices (depending on switch model) within a Virtual Chassis. 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.

## Juniper Access points

For this example, we choose Juniper Access points as our preferred access point devices. They are designed from the ground up to meet the stringent networking needs of the modern cloud and smart-device era.

## Juniper Mist Wired Assurance

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, printers, etc. It's about simplification 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 the rich Junos streaming telemetry that enable the insights for switch health metrics and anomaly detection.

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

Mist Cloud services are 100% programmable using open APIs for full automation and/or integration with your IT applications, such as Ticketing Systems, IP Management Systems, etc.

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

- UI 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 wired and wireless troubleshooting, trending analysis, anomaly detection, and proactive problem remediation.

To learn more about Juniper Mist Wired Assurance please access the following datasheet:

<https://www.juniper.net/content/dam/www/assets/datasheets/us/en/cloud-services/juniper-mist-wired-assurance-datasheet.pdf>

# Campus IP Clos Fabric 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 Layer 2 networks across one or more Layer 3 networks. By configuring different routing instances, you can enforce the separation of virtual networks because each routing instance has its own separate routing and switching table.

The 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



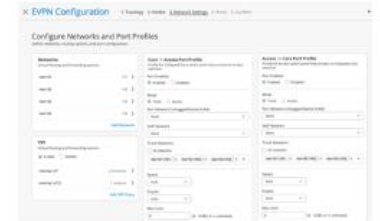
### Apply the intent

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



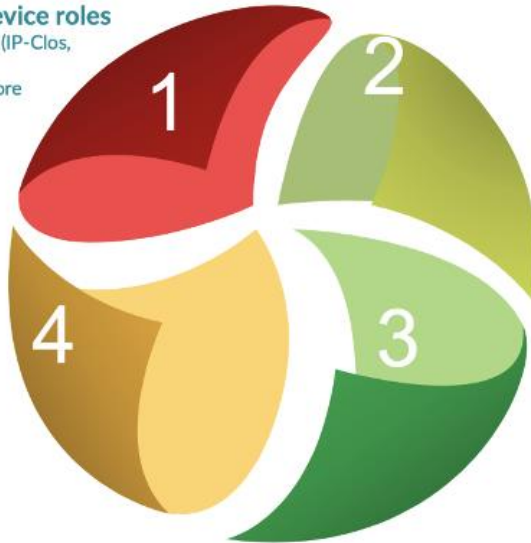
### Define Networks of Interest

- 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 21.4R1.12 or later
- Two QFX5120 switches as distribution devices, Software version: Junos OS Release 21.4R1.12 or later
- Two Access Layer EX4400 switches, Software version: Junos OS Release 22.1R1.10 or later
- One SRX345 wan router, Software version: 20.2R3-S2.5 or later
- Juniper Access Points
- 2 Linux desktops that act as a wired client

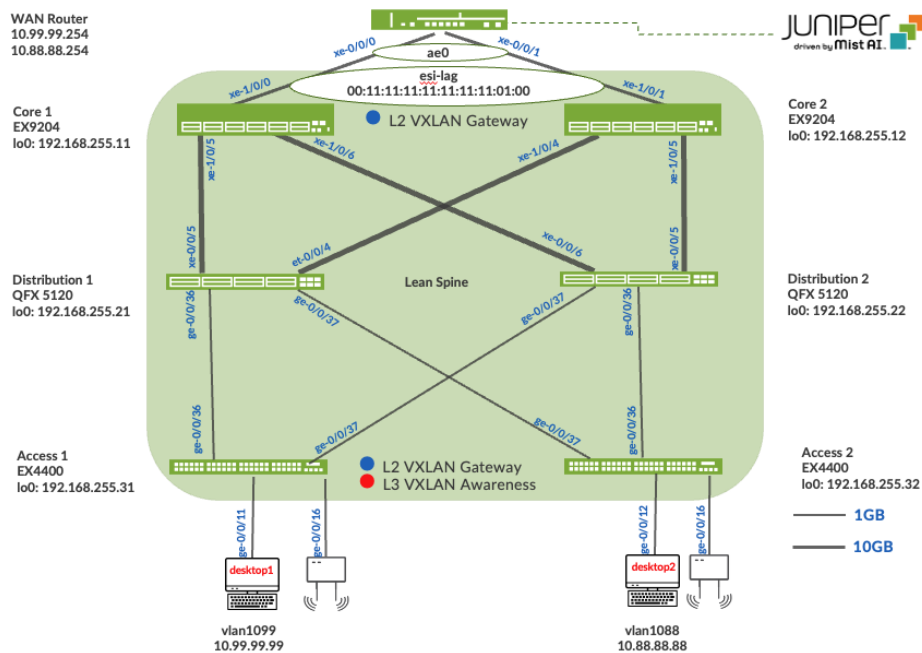


Figure 10. Topology

## Juniper Mist Wired Assurance

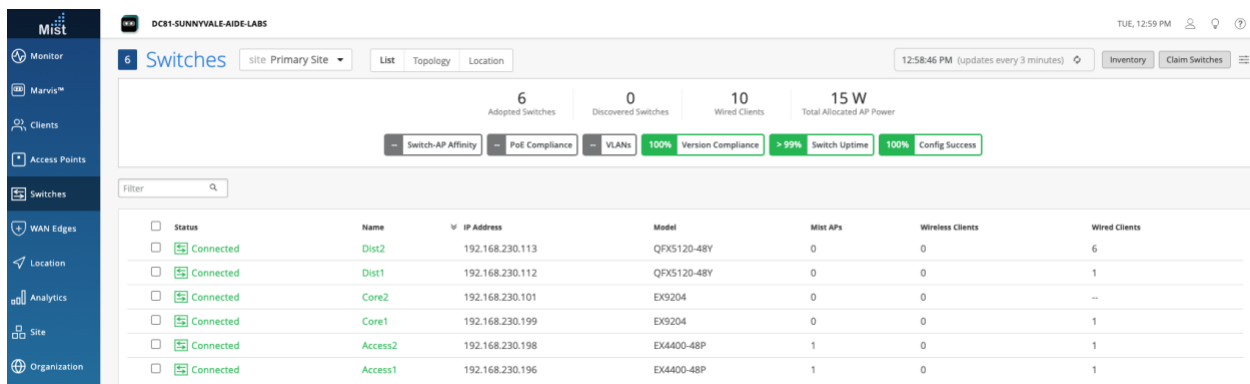
Wired Assurance, through the Mist UI, is used to build a Campus Fabric IP Clos from ground up. This includes the following:

- Assignment of p2p links between all layers of the Campus Fabric
- Assignment of unique BGP AS numbers per device participating in the underlay and overlay.
- Creation of VRF instances to allow the user the ability to logically segment traffic. This also includes the assignment of new or existing VLANs to each representative VRF
- IP addressing of each L3 gateway IRB
- IP addressing of each lo0.0 loopback
- Configuration of routing policies for underlay and overlay connectivity
- Optimized 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, please leverage the following link:  
<https://www.mist.com/documentation/category/wired-assurance/>

## Juniper Mist Wired Assurance Switches Section

The user should validate that each device participating in the Campus Fabric has been adopted or claimed, and assigned to a site. The switches were named for respective layer in the fabric to facilitate building and operating the fabric.



Status	Name	IP Address	Model	Mist APs	Wireless Clients	Wired Clients
Connected	Dist2	192.168.230.113	QFX5120-48Y	0	0	6
Connected	Dist1	192.168.230.112	QFX5120-48Y	0	0	1
Connected	Core2	192.168.230.101	EX9204	0	0	--
Connected	Core1	192.168.230.199	EX9204	0	0	1
Connected	Access2	192.168.230.198	EX4400-48P	1	0	1
Connected	Access1	192.168.230.196	EX4400-48P	1	0	1

## Overview

Use this network configuration example to deploy a single campus fabric with a Layer 3 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 as the underlay routing protocol 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 the ability to use configuration templates and a hierarchical model to group the switches and make bulk updates. Templates provide uniformity and convenience, while the hierarchy (Site and Switch) provides both scale and granularity.

What templates, and the hierarchical model, means in practice is 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 Network and Organizational levels that apply to the same device, the narrower settings (in this case, Network) 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 Network level. Of course, individual switches can also have

their own unique configurations.

You can include individual CLI commands at any level of the hierarchy, which are then appended to all the switches in that group on an “AND” basis– that is, individual CLI settings are appended to the existing configuration (existing setting may be deleted or appended).

Note: If a user utilizes CLI commands for items not native to the Mist UI, this configuration data will be applied last; overwriting existing configuration data within the same stanza.

Under Organization and Switch Templates, we utilize the following template

#### Switch Templates

TEMPLATE	SITES	SWITCHES
campus-fabric	1	6

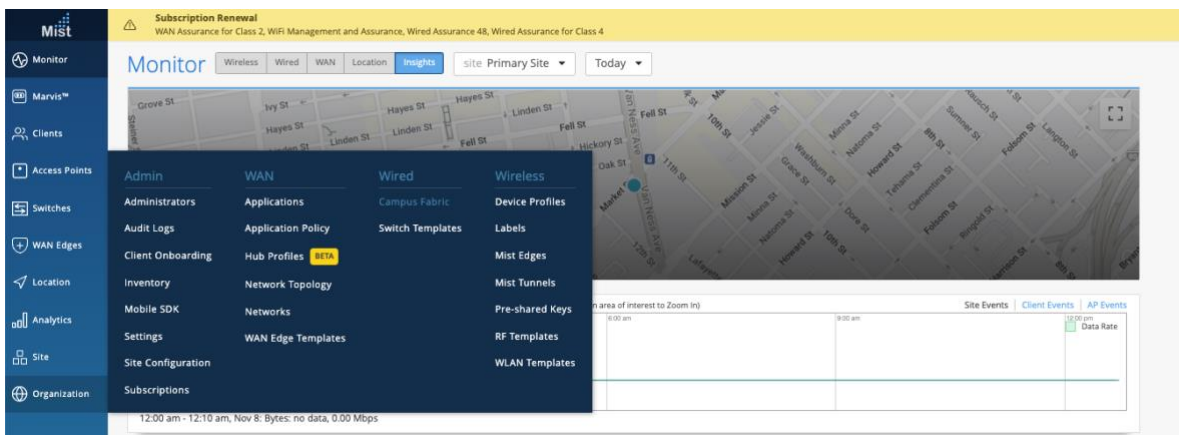
## Topology

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 via Mist UI but is separate from the campus fabric workflow. The WAN router has a southbound lag configured to connect to the ESI-LAG on the core switches. WAN router can standalone or built as an HA cluster.

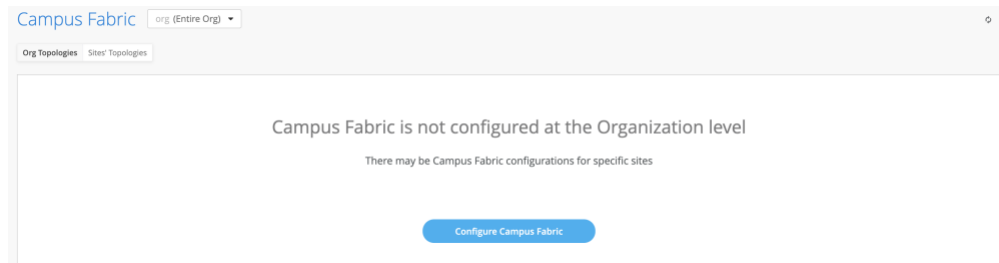
## Create the Campus Fabric

From the Organization option on the leaf hand section of the Mist UI, select Wired Campus Fabric

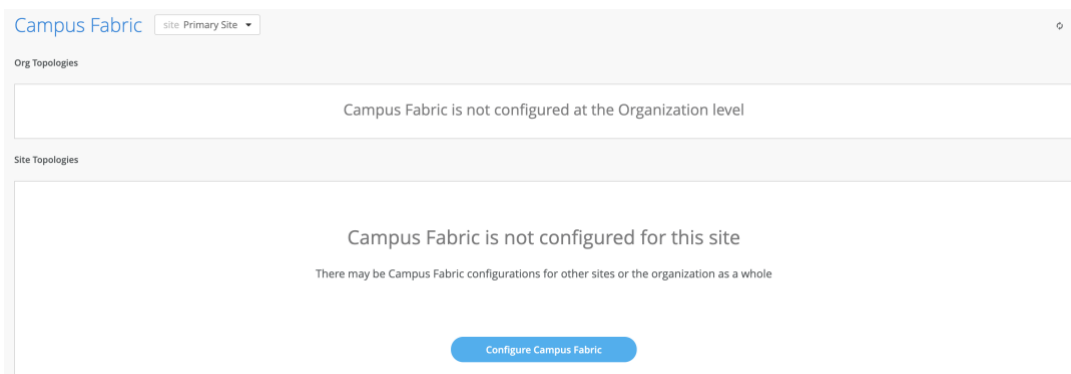


Mist provides the option of deploying a Campus Fabric at the Org or Site level noted on the upper left hand Campus Fabric pull down menu shown below. For example, those who are building a Campus wide architecture with multiple buildings, each building housing distribution and access switches, could consider building an Org level Campus Fabric that ties each of the sites together forming a holistic Campus Fabric. Otherwise, the Site build with a single set of Core, Distribution and Access switches would suffice.

## Campus Fabric Org Build



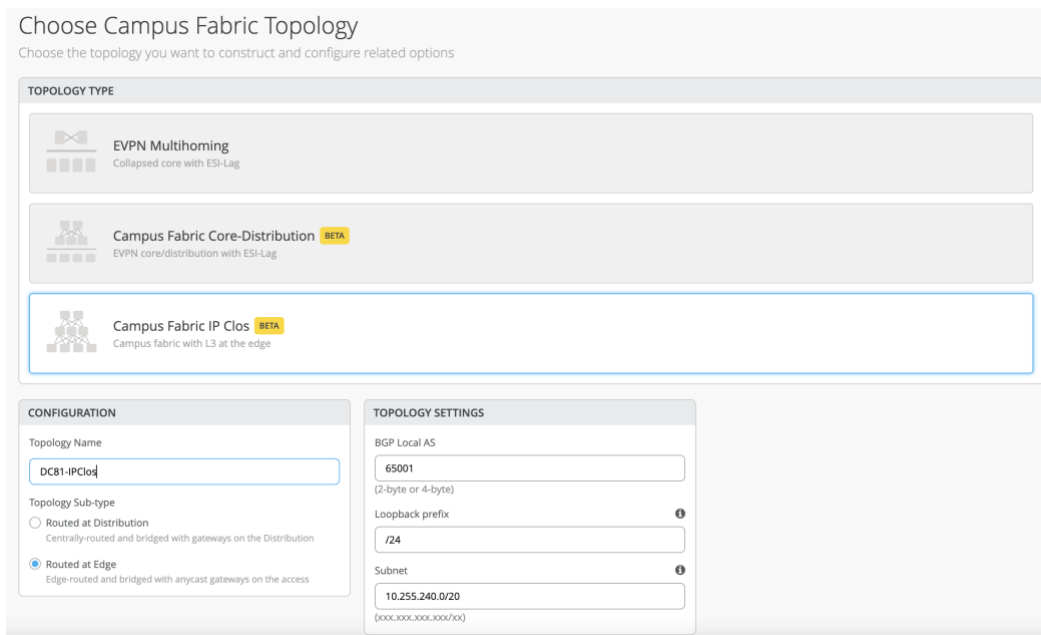
## Campus Fabric Site Build



Note: Campus Fabric Site deployment is the focus of this document

## Choose the campus fabric topology

Select the Campus Fabric IP Clos option below:



Mist provides a section to name the Campus Fabric IP Clos and where the user would like to have L3 boundaries (where Default Gateway exists for each VLAN).

## Configuration

- Provide a name in accordance with company standards

Routed at Edge/Access layer provides a smaller blast radius for broadcast traffic and is ideal for east-west traffic patterns and IP Multicast environments.

Routed at Distribution aligns with north-south traffic patterns and configures the Access Layer as a Layer 2 VXLAN gateway only. This deployment would be preferred for higher scale deployments. Please consult with your Juniper account team on Campus Fabric scale specifics.

## Topology Settings

- BGP Local AS: represents the starting point of private BGP AS numbers that will automatically be allocated per device. The user can use whatever private BGP AS number range suits their deployment, routing policy will be provisioned by Mist to ensure the AS numbers are never advertised outside of the fabric.
- Loopback prefix: represents the range of IP addresses associated with each device's loopback address. The user can use whatever range suits their deployment. VXAN tunnelling through the use of a VTEP is associated with this address.
- Subnet: represents the range of IP addresses utilized for point-to-point links between devices. L The user can use whatever range suits their deployment. Mist breaks this subnet into /31 subnet addressing per link. This number can be modified to suit the specific deployment scale. For example, /24 would provide up to 128 p2p /31 subnets.

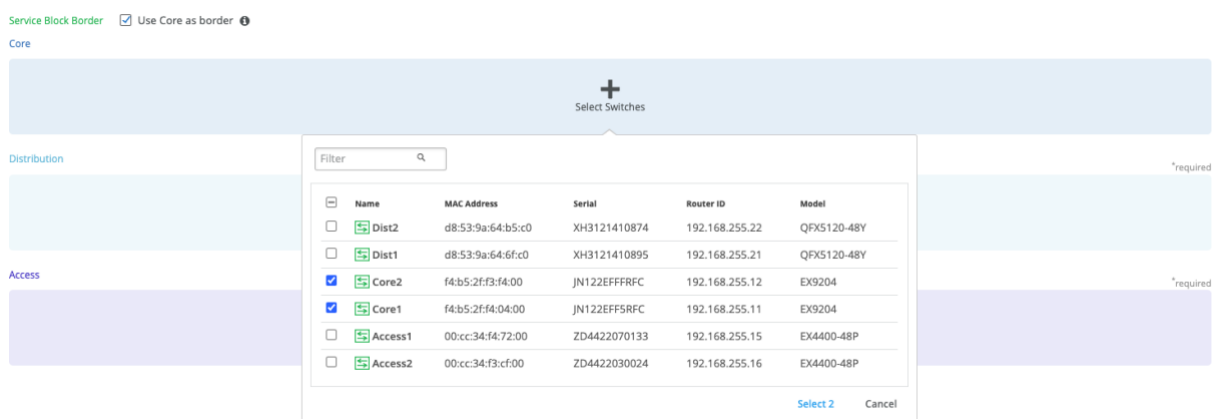
Note: Juniper recommends default settings for all options unless it conflicts with other networks attached to the campus fabric. The point-point links between each layer utilize /31 addressing to conserve addresses.

## Select campus fabric nodes

The user selects devices to participate at each Layer of the Campus Fabric IP Clos. Juniper recommends the user validate each device's presence in the site switch inventory prior to the creation of the Campus Fabric.

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

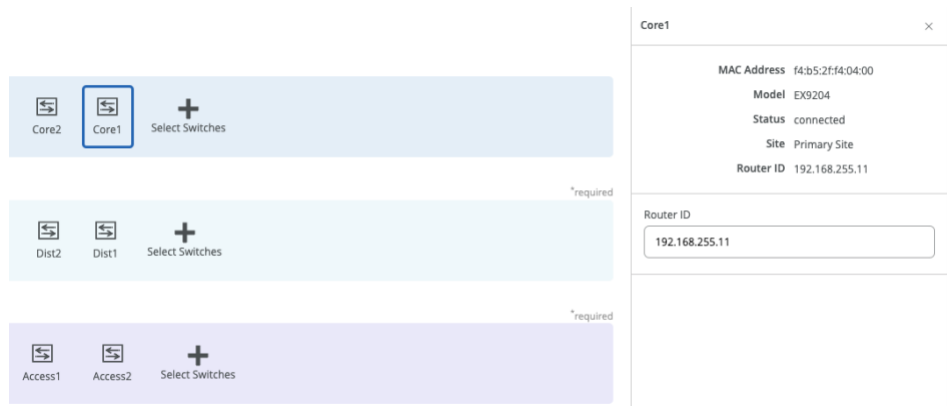
Services Block Router is where the Campus Fabric would interconnect external devices such as firewalls, routers, or critical devices such as DHCP and Radius servers (as an example). Devices to which external services connect to the Campus Fabric are known as Border Leafs. If the user wishes to connect these services/devices to the Campus Fabric IP Clos in a separate device or pair of devices, the Use Core as border option should be unchecked and the devices chosen by choosing the Select Switches option.



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. Users who wish to combine this capability within the Core devices should select the User Core as border option.

Once all layers have selected the appropriate devices, the user must provide a loopback IP address for each device. This loopback is associated with a logical construct called a VTEP; used to source the VXLAN Tunnel. Campus Fabric IP Clos has VTEPs for VXLAN tunnelling on the Access switches and the Core switches when enabling the Core Border option.

The loopback addresses and router-ids should be in the same address space. The host-id of the loopback can be customized to differentiate between core, distribution and access. This can help identify devices if you are troubleshooting or following nethops. The loopback is also used as the router-id and will be used for overlay eBGP peering and VXLAN tunnel termination.



**Note:** The loopback address and router-id should be in the same subnet as provided by Mist.

The loopback prefix is used for import /export policies. The subnet addresses are used for point-to-point links throughout the Fabric. Mist automatically creates policies that import and export loopback addresses used within the Campus Fabric. The selection of fabric type presents the user with default settings, which can be adapted as required.

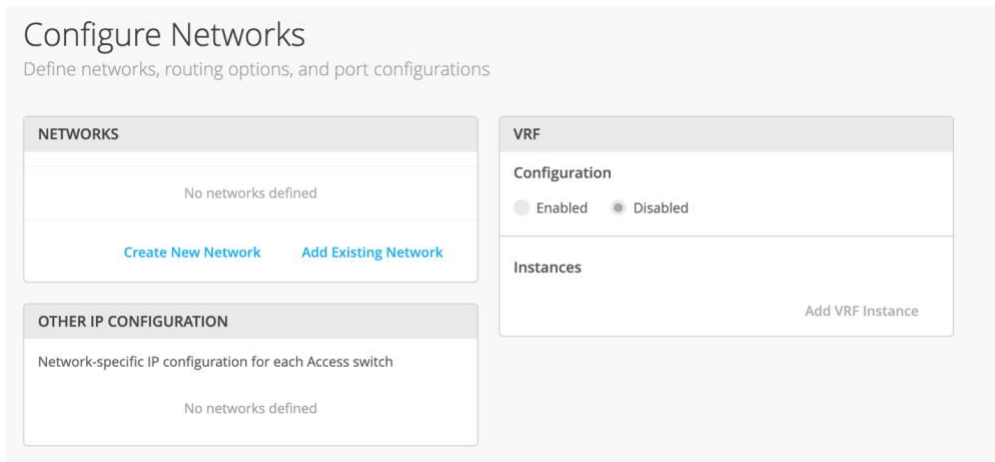
Loopback prefix ⓘ

Subnet ⓘ  
  
(xxx.xxx.xxx.xxx/xx)

## Configure Networks

Mist presents the user with input for Network information such as VLANs and VRF (routing instances for traffic isolation purposes) options. VLANs are mapped to 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.

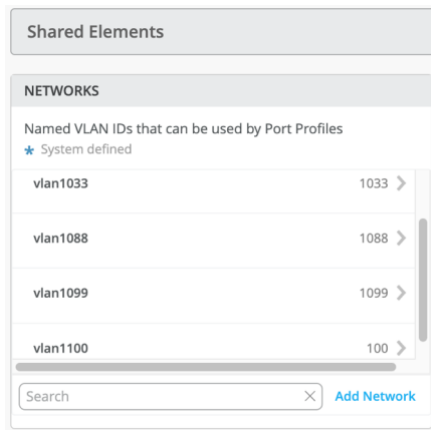




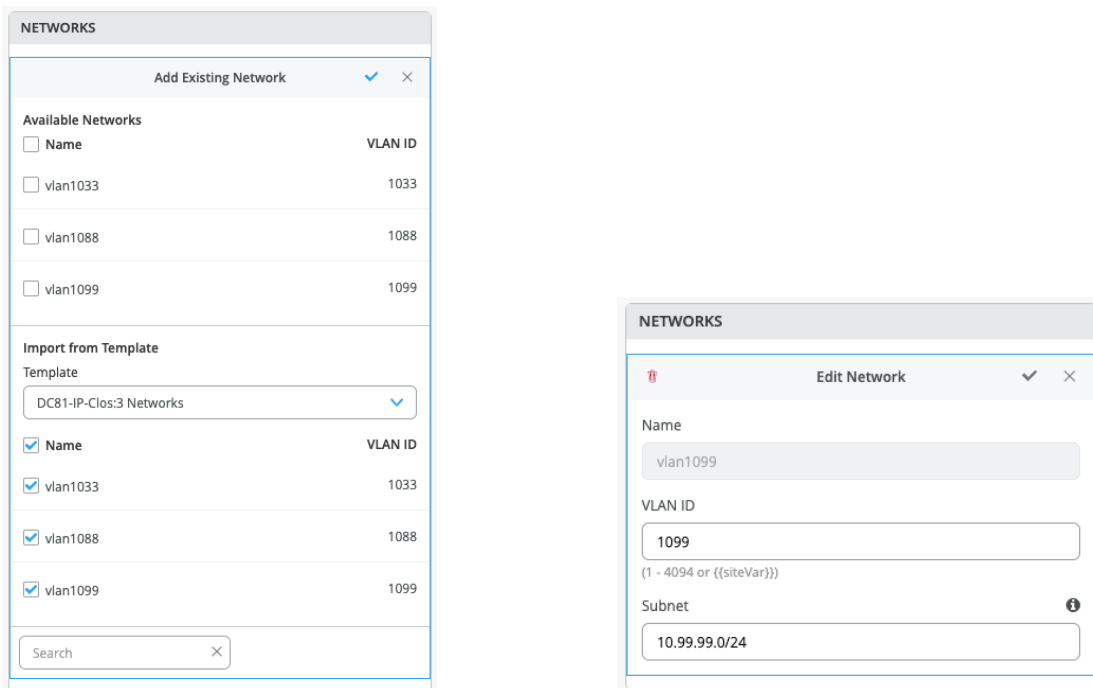
## Networks

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

The Shared Elements section of the campus-fabric template includes the Networks section mentioned above where VLANs are created.



Back to the Campus Fabric build, the user selects the existing template that includes L2 VLAN information. All VLAN and IP information will be inherited from the template



## Other IP Configuration

Mist Wired Assurance provides automatic IP addressing (IRBs) for each of the VLANs. Port Profiles and Port Configuration then associate the VLAN with specified ports. In this case, we selected IP Clos

Routed at Edge at the onset of the Campus Fabric build.

**CONFIGURATION**

Topology Name

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 utilizes anycast addressing for all devices participating in the L3 subnet. Access1 and Access2 switches will be configured with the same IP address for each L3 subnet.

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 the user to group common VLANs into the same VRF or separate VRFs depending on traffic isolation requirements. This example includes 3 VRFs or routing instances: corp-it | developers | guest-wifi. Here, the user builds the first corp-it VRF and selects the pre-defined vlan 1099.

**VRF**

Configuration  
 Enabled  Disabled

Instances  
 No VRF instances defined

[Add VRF Instance](#)

**VRF**

New VRF Instance ✓ ✕

Name

Networks  
 vlan1088  vlan1099  vlan1033

Extra Routes  
 No extra routes defined

[Add Extra Routes](#)

By default, inter-VRF communications is not supported within the Campus Fabric. If inter-VRF communications is required, each VRF can include extra routes such as a Default Route that will instruct the Campus Fabric to use an external router or firewall for further security inspection or routing capabilities. In this example, all traffic is trunked over the ESI-LAG and the Juniper SRX handles inter-VRF routing. Figure 10. Topology

Notice the SRX participates in the VLANs defined within the Campus Fabric and is the gateway of last resort for all traffic leaving the subnet. The user selects the “Add Extra Routes” option to inform Mist to forward all traffic leaving 10.99.99.0/24 to utilize the next hop of the Juniper SRX firewall: 10.99.99.254

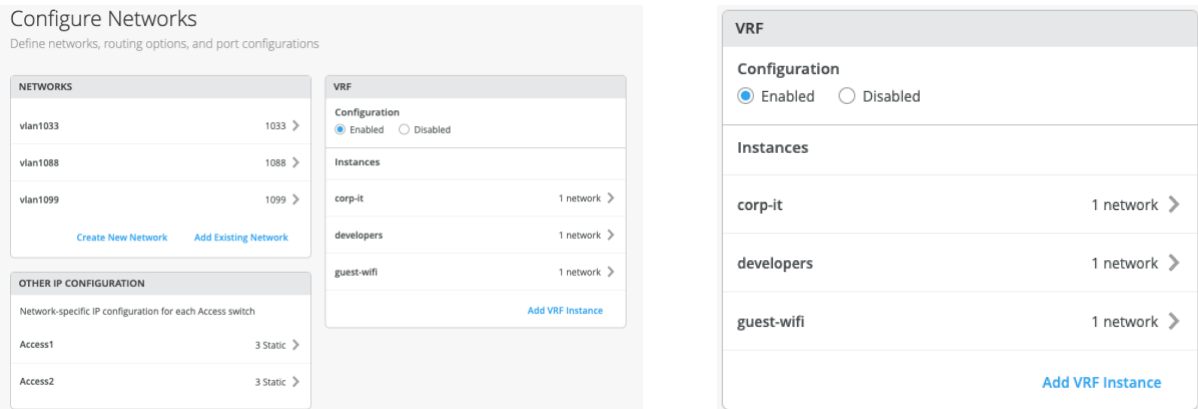
**New Extra Route** ✓ ✕

Route

Via

The user creates 2 additional VRFs

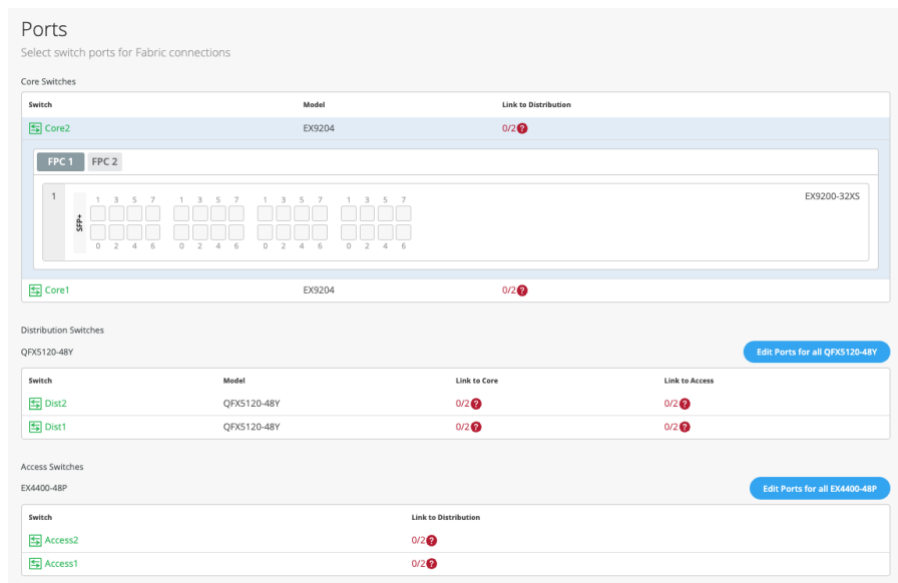
- developers using vlan 1088 with 0.0.0.0/0 utilizing 10.88.88.254
- guest-wifi using vlan 1033 with 0.0.0.0/0 utilizing 10.33.33.254



Now that all VLANs are configured and assigned to each VRF, the user can move to the next step by clicking the Continue button at the upper right section of the Mist UI.

### Configure campus fabric ports

The final step is the selection of physical ports between Core, Distribution and Access Switches.

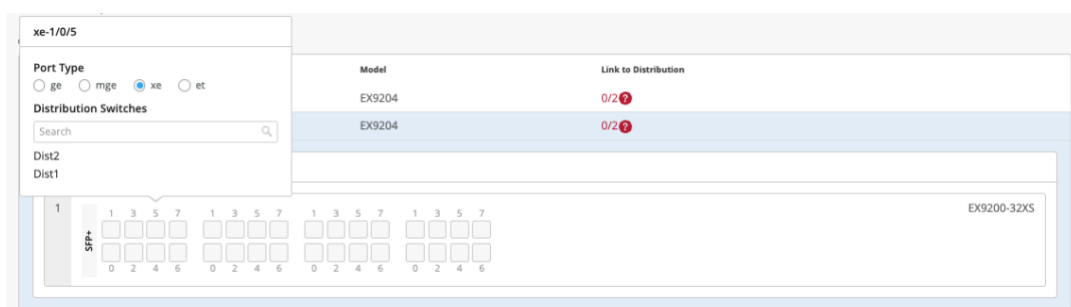


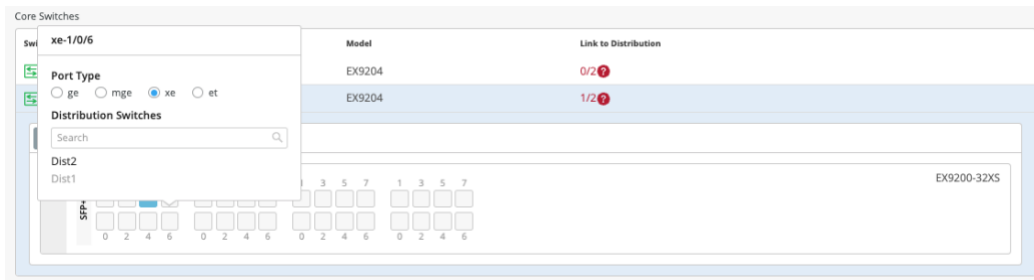
Note: Juniper recommends the user have the output of the show lldp neighbors command from each switch (assuming LLDP is enabled before the switches were selected). This output provides a source of truth for which ports should be selected during at each layer.

### Core Switches

#### Core1:

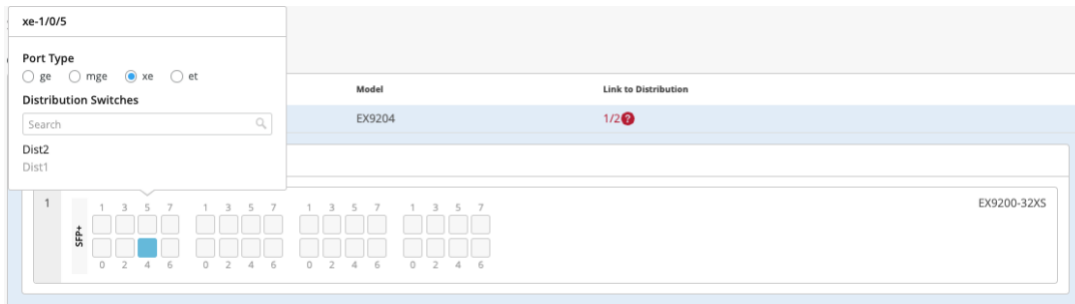
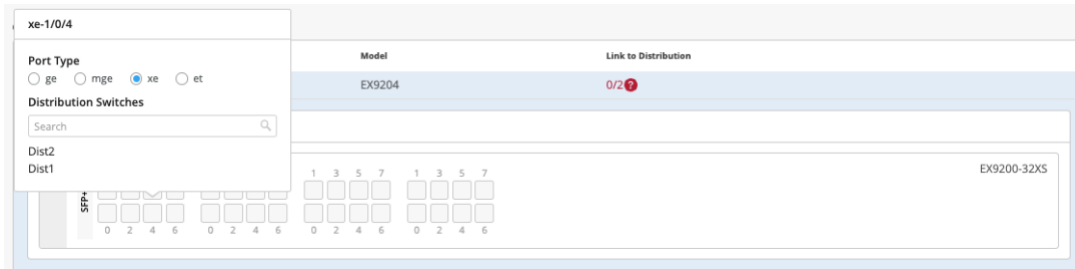
Starting with Core1, the user selects xe-1/0/5 and xe-1/0/6 terminating on Distribution Switches 1 and 2 respectively.





**Core2:**

On Core2, the user selects xe-1/0/4 and xe-1/0/5 terminating on Distribution Switches 1 and 2 respectively:



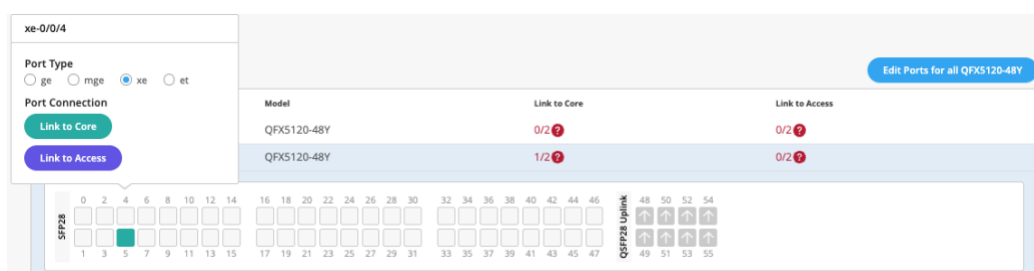
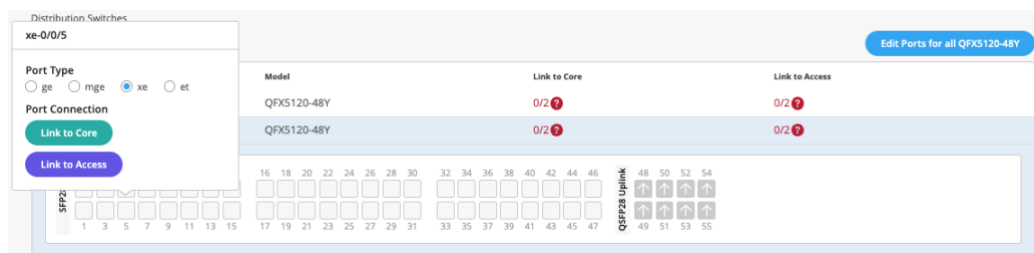
**Distribution Switches**

Now moving on to the Distribution Switches, you will notice 2 interconnect options exist

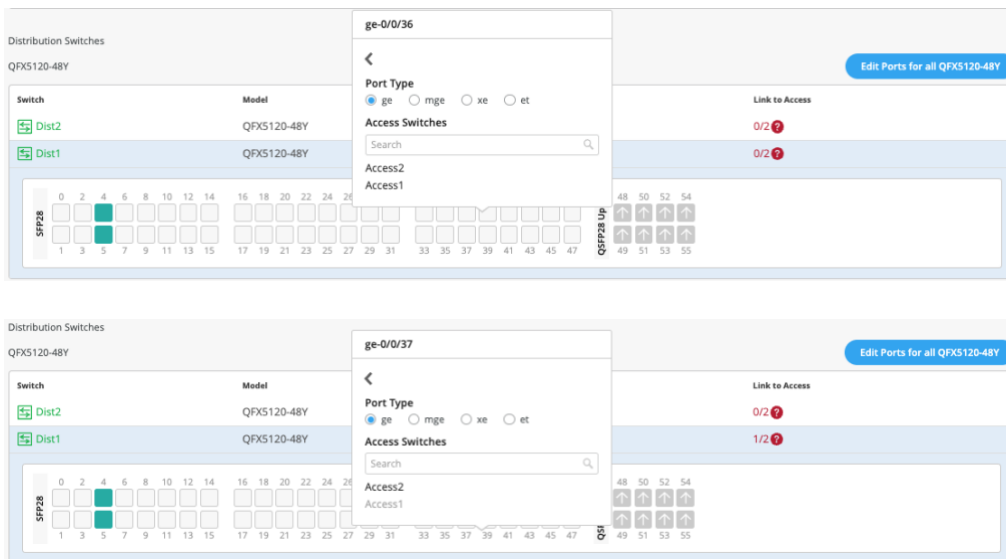
- Link to Core
- Link to Access

**Dist1:**

The user selects Link to Core and choose xe-0/0/5 and xe-0/0/4 terminating on Core Switches 1 and 2 respectively.



The user selects Link to Access and choose ge-0/0/36 and ge-0/0/37 terminating on Access Switches 1 and 2 respectively:



Next, the user selects the following interconnects off **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, the user selects the following interface combinations for Access1 and Access2:

Access1:

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

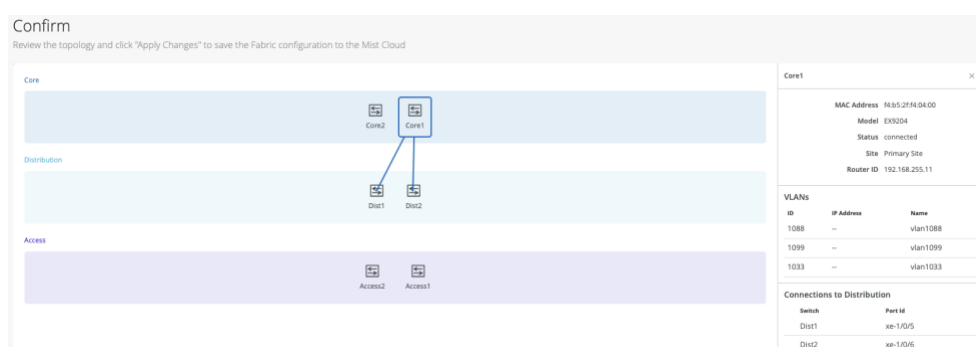
Access2:

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

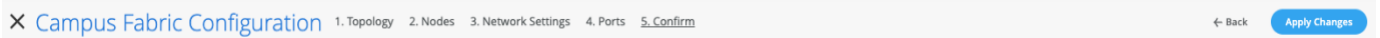
Once the user has completed selecting all requisite port combinations, they will select the Continue button at the upper right-hand corner of the Mist UI.

## Campus Fabric Configuration Confirmation

This last section provides the user with the ability to confirm each device's configuration as shown below:



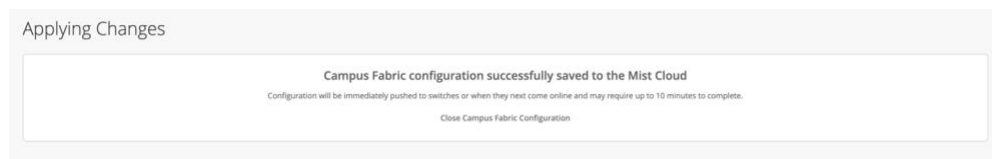
Once the user has completed verification, they will select the Apply Changes option at the upper right-hand corner of the Mist UI



The user is presented a second stage confirmation, confirm to create the fabric.

Mist presents the user with the following banner including the estimated time for the Campus Fabric to be built. The process includes the following:

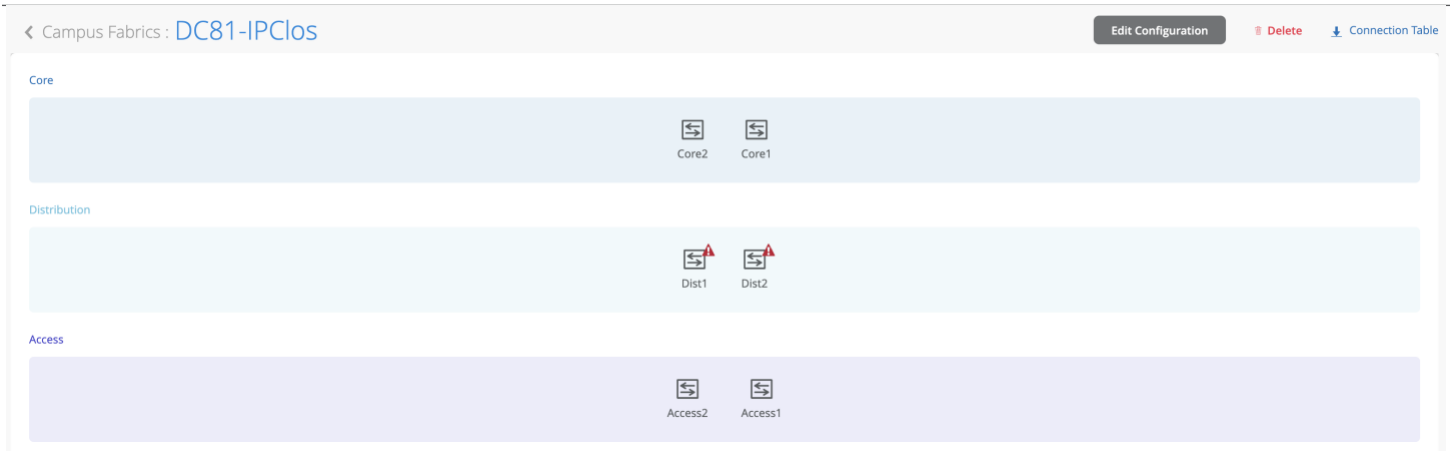
- Mist builds the point-to-point interfaces between all devices with IP addresses chosen from the range presented at the onset of the build.
- Each device is configured with a loopback address from the range presented at the onset of the build.
- eBGP is provisioned at each device with unique BGP autonomous 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 support of customer traffic using EVPN-VXLAN.
- IP addressing of each L3 gateway IRB
- IP addressing of each lo0.0 loopback
- Configuration of routing policies for underlay and overlay connectivity
- Optimized MTU settings for p2p underlay, L3 IRB, and ESI-LAG bundles
- VXLAN to VLAN mapping using VNI addresses that are automatically assigned
- VRF creation of corp-it, developers, and guest-wifi and VLAN associated with each VRF
- VXLAN tunnelling creation between Access devices and Access-Core devices (in support of the northbound SRX firewall that will be 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



Closing this section provides the user with a summary of the newly created Campus Fabric IP Clos



Juniper Mist Wired Assurance provides the user with the ability to 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, the user can download the connection table:



**Disclaimer:** The RED designations depicted on the Distribution Switches will be fixed with a future software update.

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	f4b52ff3f400	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	00cc34f3cf00	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	f4b52ff3f400	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	00cc34f3cf00	Access1	access

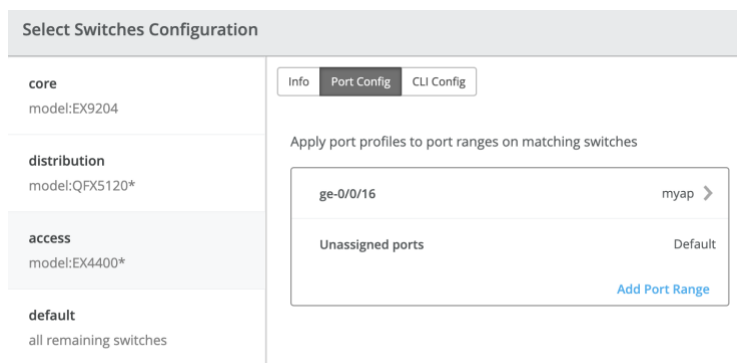
### Apply VLANs to Access ports

As previously discussed, Mist provides the ability to templatize well known services such as Radius, NTP, DNS, etc 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, Desktop1/2 are associated with different ports on each Access Switch which requires the configuration to be applied to Access1/2 respectively. Figure 10. Topology

It is also noteworthy that Mist Access Points 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 Switch Template option is customized to associate each switch with its role: Core, Distribution, and Access. Further, all Access switches (defined by Model EX4400 as an example) associated the AP port profile with ge-0/0/16 without needing to configure each independent switch.



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

**PORT CONFIGURATION**

Port Profile Assignment  
 ★ Site, Template, or System Defined

**Edit Port Range** ✓ ✕

Port Aggregation

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

Enable Dynamic Configuration

The Switch Template definition for vlan1099 is shown below, representing attributes associated with VLANs such as dot1x authentication, QoS, and Power over Ethernet. Vlan1088 and vlan1033 will need to be configured in a similar fashion.

**Edit Port Profile** ✓ ✕

Name

Port Enabled  
 Enabled  Disabled

Description

Mode  
 Trunk  Access

Port Network (Untagged/Native VLAN)

VoIP Network

Use dot1x authentication

Speed

Duplex

Mac Limit  
 (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

Verification of the Campus Fabric IP Clos deployment. Figure 10. Topology Currently there are two desktops that can be used to validate the fabric. Let's take a quick look to see if Desktop1 can connect internally and externally.

```
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:fe74:a06f prefixlen 64 scopeid 0x20<link>
    ether 52:54:00:74:a0:6f txqueuelen 1000 (Ethernet)
    RX packets 28044 bytes 17108274 (17.1 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 26564 bytes 2271495 (2.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@desktop1:~# ip r
default via 10.99.99.1 dev vlan1099
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 -c 2
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=6.45 ms
64 bytes from 10.99.99.1: icmp_seq=2 ttl=64 time=8.86 ms

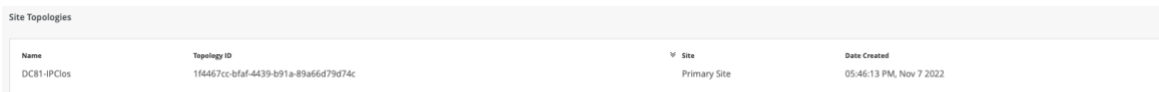
--- 10.99.99.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 6.452/7.653/8.855/1.201 ms
root@desktop1:~# ping 10.99.99.254 -c 2
PING 10.99.99.254 (10.99.99.254) 56(84) bytes of data:
From 10.99.99.99 icmp_seq=1 Destination Host Unreachable
From 10.99.99.99 icmp_seq=2 Destination Host Unreachable

--- 10.99.99.254 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1016ms
```

### Validation steps

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

Start by validating Campus Fabric in the Mist UI, by selecting the Campus Fabric option under the Organization tab on the left-hand side of the UI.



Name	Topology ID	Site	Date Created
DC81-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/Receive traffic on a link-by-link basis
- Telemetry, such as lldp, from each device that verifies the physical build

## 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.
- bfd, bi-directional forwarding, to decrease convergence times during failures
- loopback reachability to support VXLAN tunnelling

Without requiring verification at each layer, the focus can be on Dist1/2 and their eBGP relationships with Access1/2 and Core1/2. If both Dist switches have “established” eBGP peering sessions with each adjacent layer, the user can move to the next phase of verification.

## Action

Verify that BGP sessions are established from Dist1/2 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 UI or using an external application such as Secure CRT or Putty.

## Verification of BGP peering

### Dist1:

Remote Shell can be accessed via the bottom right of the Campus Fabric, from the switch view or via SSH.

```
{master:0}
root@Dist1> show bgp summary

Warning: License key missing; requires 'bgp' license

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
bgp.evpn.0
Peer          AS      InPkt  OutPkt  OutQ   Flaps  Last Up/Dwn  State|#Active/Received/Accepted/Damped...
10.255.240.2  65001   5709   5658    0      0      0 1d 19:13:28  Establ
  inet.0: 2/4/4/0
10.255.240.6  65002   5685   5631    0      1 1d 19:01:29  Establ
  inet.0: 2/4/4/0
10.255.240.11 65005   5649   5597    0      0 1d 18:46:54  Establ
  inet.0: 2/2/2/0
10.255.240.13 65006   5654   5600    0      0 1d 18:47:02  Establ
  inet.0: 2/4/4/0
192.168.255.11 65002   6026   5990    0      1 1d 19:01:20  Establ
  bgp.evpn.0: 14/33/33/0
192.168.255.12 65001   6018   6140    0      0 1d 19:13:25  Establ
  bgp.evpn.0: 2/28/28/0
192.168.255.31 65006   6029   6019    0      0 1d 18:46:59  Establ
  bgp.evpn.0: 21/32/32/0
192.168.255.32 65005   6012   6085    0      0 1d 18:46:52  Establ
  bgp.evpn.0: 22/34/34/0

{master:0}
root@Dist1>
```

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

It also shows the overlay (192.168.255.x) relationships are established and that it is peering at the correct loopback addresses. This demonstrates loopback reachability.

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

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

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

```
{master:0}
root@Dist1> ping 192.168.255.12 count 1
PING 192.168.255.12 (192.168.255.12): 56 data bytes
64 bytes from 192.168.255.12: icmp_seq=0 ttl=64 time=0.910 ms

--- 192.168.255.12 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.910/0.910/0.910/0.000 ms

{master:0}
root@Dist1> ping 192.168.255.32 count 1
PING 192.168.255.32 (192.168.255.32): 56 data bytes
64 bytes from 192.168.255.32: icmp_seq=0 ttl=64 time=4.299 ms

--- 192.168.255.32 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 4.299/4.299/4.299/0.000 ms
```

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

Let's verify the routes are established to the to the Core and other devices across multiple paths. For example, Access1/2 should leverage both paths through Dist1/2 to access Core1/2's loopbacks as well each other's loopbacks.

## Access1: Loopback reachability to Core1 through Dist1/2

```
{master:0}
root@Access1> show route forwarding-table destination 192.168.255.11
Routing table: default.inet
Internet:
Destination      Type RtRef Next hop          Type Index  NhRef Netif
192.168.255.11/32 user  0
                  10.255.240.12 ucst  1898  6 ge-0/0/36.0
                  10.255.240.16 ucst  1899  6 ge-0/0/37.0
```

## Access1: Loopback reachability with Core2 through Dist1/2

```
{master:0}
root@Access1> show route forwarding-table destination 192.168.255.12
Routing table: default.inet
Internet:
Destination      Type RtRef Next hop          Type Index  NhRef Netif
192.168.255.12/32 user  0
                  10.255.240.12 ucst  1898  6 ge-0/0/36.0
                  10.255.240.16 ucst  1899  6 ge-0/0/37.0
```

## Access1: Loopback reachability with Access2 through Dist1/2

```
{master:0}
root@Access1> show route forwarding-table destination 192.168.255.32
Routing table: default.inet
Internet:
Destination      Type RtRef Next hop          Type Index  NhRef Netif
192.168.255.32/32 user  0
                  10.255.240.12 ucst  1898  6 ge-0/0/36.0
                  10.255.240.16 ucst  1899  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 is operational through the verification of eBGP between corresponding layers of the Campus Fabric and that routes are established to Access, Core and Distribution.

## 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, vlan and interface-mode are correct. If pinging the default gateway failed, then troubleshoot underlay connectivity.

## Verification of the EVPN Database on both access switches

```
{master:0}
root@Access1> show evpn database
Instance: default-switch
VLAN  DomainId  MAC address      Active source      Timestamp      IP address
1      00:cc:34:f3:cf:00 192.168.255.32    Nov 07 23:13:46
1      00:cc:34:f4:72:00  irb.0            Nov 07 23:13:20
1      f4:b5:2f:f3:fb:f0 192.168.255.12   Nov 07 23:13:34
1      f4:b5:2f:f4:0b:f0 192.168.255.11   Nov 07 23:13:34
11099  00:00:5e:e4:31:57  irb.1099         Nov 07 23:13:20  10.99.99.1
11099  52:54:00:74:a0:6f  ge-0/0/11.0      Nov 09 11:11:38  10.99.99.99
21088  00:00:5e:e4:31:57  irb.1088         Nov 08 15:29:02  10.88.88.1
21088  52:54:00:f7:12:2d 192.168.255.32   Nov 07 23:13:46  10.88.88.88
21088  f4:a7:39:6b:e3:20 192.168.255.32   Nov 08 04:21:53  10.88.88.10
31033  00:00:5e:e4:31:57  irb.1033         Nov 07 23:13:20  10.33.33.1
31033  5c:5b:35:2e:53:61 192.168.255.32   Nov 08 15:25:52
31033  5c:5b:35:af:29:d5  ge-0/0/16.0      Nov 08 15:25:52

{master:0}
root@Access1> show evpn database | match 52:54:00:74:a0:6f
11099  52:54:00:74:a0:6f  ge-0/0/11.0      Nov 09 11:11:38  10.99.99.99

{master:0}
root@Access1>
```

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

```
root@Access2>
{master:0}
root@Access2> show evpn database
Instance: default-switch
VLAN  DomainId  MAC address      Active source      Timestamp          IP address
1      1            00:cc:34:f3:cf:00  irb.0              Nov 07 23:13:26
1      1            00:cc:34:f4:72:00  192.168.255.31     Nov 07 23:13:46
1      1            f4:b5:2f:f3:fb:f0  192.168.255.12     Nov 07 23:13:46
1      1            f4:b5:2f:f4:0b:f0  192.168.255.11     Nov 07 23:13:46
11099  00:00:5e:e4:31:57  irb.1099          Nov 08 15:31:24    10.99.99.1
11099  52:54:00:74:a0:6f  192.168.255.31     Nov 07 23:16:31    10.99.99.99
21088  00:00:5e:e4:31:57  irb.1088          Nov 07 23:13:26    10.88.88.1
21088  52:54:00:f7:12:2d  ge-0/0/12.0       Nov 07 23:13:27    10.88.88.88
21088  f4:a7:39:6b:e3:20  ge-0/0/12.0       Nov 09 07:50:17    10.88.88.10
31033  00:00:5e:e4:31:57  irb.1033          Nov 07 23:13:26    10.33.33.1
31033  5c:5b:35:2e:53:61  ge-0/0/16.0       Nov 08 15:25:52
31033  5c:5b:35:af:29:d5  192.168.255.31     Nov 08 15:25:52

{master:0}
root@Access2> show evpn database | match 52:54:00:74:a0:6f
11099  52:54:00:74:a0:6f  192.168.255.31     Nov 07 23:16:31    10.99.99.99

{master:0}
root@Access2>
```

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 represented in the Active Source 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 Access and Core switches.

## Access

```
{master:0}
root@Access1> show configuration vlans |display set |display inheritance | match 1099
set vlans vlan1099 vlan-id 1099
set vlans vlan1099 l3-interface irb.1099
set vlans vlan1099 vxlan vni 11099
```

## Core

```
root@Core1> show configuration |display s|match 1099
set groups top routing-instances evpn_vrf vlans vlan1099 vxlan vni 11099

root@Core1> █
```

## Verification of VXLAN tunnelling between Access and Core devices

### Access 1:

```
{master:0}
root@Access1> show ethernet-switching vxlan-tunnel-end-point remote summary
Logical System Name  Id SVTEP-IP      IFL  L3-Idx  SVTEP-Mode  ELP-SVTEP-IP
<default>
  0  192.168.255.31  lo0.0  0
RVTEP-IP            L2-RTT          IFL-Idx  Interface  NH-Id  RVTEP-Mode  ELP-IP      Flags
192.168.255.11     default-switch  618      vtep.32770  1901   RNVE
192.168.255.12     default-switch  617      vtep.32769  1900   RNVE
192.168.255.32     default-switch  619      vtep.32771  1912   RNVE
```

### Access 2:

```
{master:0}
root@Access2> show ethernet-switching vxlan-tunnel-end-point remote summary
Logical System Name  Id SVTEP-IP      IFL  L3-Idx  SVTEP-Mode  ELP-SVTEP-IP
<default>
  0  192.168.255.32  lo0.0  0
RVTEP-IP            L2-RTT          IFL-Idx  Interface  NH-Id  RVTEP-Mode  ELP-IP      Flags
192.168.255.11     default-switch  618      vtep.32770  1901   RNVE
192.168.255.12     default-switch  617      vtep.32769  1900   RNVE
192.168.255.31     default-switch  619      vtep.32771  1902   RNVE
```

Note: Both Access switches display each other in the output as well as Core1/2. The reason VXLAN is supported on Core1/2 is in support of the L2 multihomed connection to the wanrouter, in this case a Juniper SRX. The L2 connection was built through the use of Mist Port Profiles and does not require the user to leave the Campus Fabric section once built.

## Verify Desktop1's MAC address being advertised via BGP

```
(master:0)
root@Access1> show route advertising-protocol bgp 192.168.255.21 evpn-mac-address 00:00:5e:e4:31:57 table bgp.evpn.0
Warning: License key missing; One or more members of the VC require 'bgp' license

bgp.evpn.0: 46 destinations, 72 routes (46 active, 0 holddown, 0 hidden)
Prefix      Nexthop      MED      Lclpref   AS path
2:192.168.255.31:1::11099::00:00:5e:e4:31:57/304 MAC/IP
*           Self                    I
2:192.168.255.31:1::21088::00:00:5e:e4:31:57/304 MAC/IP
*           Self                    I
2:192.168.255.31:1::31033::00:00:5e:e4:31:57/304 MAC/IP
*           Self                    I
*2:192.168.255.31:1::11099::00:00:5e:e4:31:57::10.99.99.1/304 MAC/IP
*           Self                    I
2:192.168.255.31:1::21088::00:00:5e:e4:31:57::10.88.88.1/304 MAC/IP
*           Self                    I
2:192.168.255.31:1::31033::00:00:5e:e4:31:57::10.33.33.1/304 MAC/IP
*           Self                    I
(master:0)
```

And is it being received on the core

```
root@Core1> show route receive-protocol bgp 192.168.255.21 evpn-mac-address 00:00:5e:e4:31:57 table bgp.evpn.0
Warning: License key missing; requires 'bgp' license

bgp.evpn.0: 46 destinations, 90 routes (46 active, 0 holddown, 0 hidden)
Prefix      Nexthop      MED      Lclpref   AS path
2:192.168.255.31:1::11099::00:00:5e:e4:31:57/304 MAC/IP
192.168.255.31 65003 65006 I
2:192.168.255.31:1::21088::00:00:5e:e4:31:57/304 MAC/IP
192.168.255.31 65003 65006 I
2:192.168.255.31:1::31033::00:00:5e:e4:31:57/304 MAC/IP
192.168.255.31 65003 65006 I
2:192.168.255.32:1::11099::00:00:5e:e4:31:57/304 MAC/IP
192.168.255.32 65003 65005 I
* 2:192.168.255.32:1::21088::00:00:5e:e4:31:57/304 MAC/IP
192.168.255.32 65003 65005 I
* 2:192.168.255.32:1::31033::00:00:5e:e4:31:57/304 MAC/IP
192.168.255.32 65003 65005 I
* 2:192.168.255.31:1::11099::00:00:5e:e4:31:57::10.99.99.1/304 MAC/IP
192.168.255.31 65003 65006 I
2:192.168.255.31:1::21088::00:00:5e:e4:31:57::10.88.88.1/304 MAC/IP
192.168.255.31 65003 65006 I
2:192.168.255.31:1::31033::00:00:5e:e4:31:57::10.33.33.1/304 MAC/IP
192.168.255.31 65003 65006 I
```

Lets check to see if the core has Desktop1 mac address

```
root@Core1> show evpn database | match 00:00:5e:e4:31:57
11099 00:00:5e:e4:31:57 192.168.255.31 Nov 07 23:13:34 10.99.99.1
21088 00:00:5e:e4:31:57 192.168.255.31 Nov 08 15:29:02 10.88.88.1
31033 00:00:5e:e4:31:57 192.168.255.31 Nov 07 23:13:34 10.33.33.1
root@Core1>
```

Verify the MAC address mapped to the correct VTEP interface. This is on the core, you can also verify on Access switch

```
root@Core1> show route forwarding-table family ethernet-switching extensive destination 00:00:5e:e4:31:57
Routing table: evpn_vrf.evpn-vxlan [Index 8]
Bridging domain: vlan1033.evpn-vxlan [Index 11]
VPLS:
Enabled protocols: Bridging, ACKed by all peers, EVPN VXLAN,
Destination: 00:00:5e:e4:31:57/48
Learn VLAN: 0
Route reference: 0
Multicast RPF nh index: 0
P2mpidx: 0
IFL generation: 1104
Sequence Number: 0
L2 Flags: control_dyn
Flags: sent to PFE
Epoch: 0
Learn Mask: 0x40000000000000000000000000000000
Nexthop:
Next-hop type: composite Index: 697 Reference: 14
```

```
root@Core1> show route forwarding-table family ethernet-switching
Routing table: default-switch.bridge
VPLS:
Destination      Type RtRef Next hop      Type Index  NhRef Netif
default          perm  0         dscd          526    1
xe-1/0/4.0      intf  0         ucst          668    2 xe-1/0/4.0
xe-1/0/0.0      intf  0         ucst          689    3 xe-1/0/0.0

Routing table: default-switch.bridge
Bridging domain: default.bridge
VPLS:
Enabled protocols: Bridging, ACKed by all peers,
Destination      Type RtRef Next hop      Type Index  NhRef Netif
0x3000b/51       user  0         comp          673    2
0x30009/51       user  0         comp          674    2

Routing table: evpn_vrf.evpn-vxlan
VPLS:
Destination      Type RtRef Next hop      Type Index  NhRef Netif
default          perm  0         dscd          680    1
vtep.32769       intf  0         comp          642    5
vtep.32770       intf  0         comp          697    13
vtep.32771       intf  0         comp          698    11
```

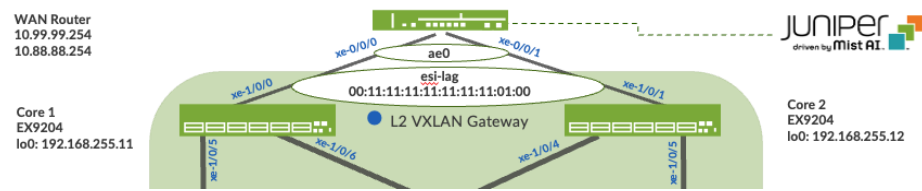
Finally is the VTEP interface up and passing traffic:

```
root@Core1> show interfaces vtep.32770
Logical interface vtep.32770 (Index 357) (SNMP ifIndex 658)
Flags: Up SNMP-Traps Encapsulation: ENET2
VXLAN Endpoint Type: Remote, VXLAN Endpoint Address: 192.168.255.31, L2 Routing Instance: evpn_vrf, L3 Routing Instance: default
Input packets : 41347
Output packets: 6719
Protocol eth-switch, MTU: Unlimited
Flags: Trunk-Mode
```

From an EVPN-VLAN perspective everything is looking correct. Maybe we are looking in the wrong place. Let's look at the connection between Core and WAN router.

### External Campus Fabric connectivity through the Border GW Core EX9204 switches

Remember that the user chose to deploy the Border GW capability on the EX9204 switches during the IP Clos deployment, represented below:



Mist enables the EX9204 to translate between VXLAN traffic within the Campus Fabric and standard ethernet switching for external connectivity, in this case a SRX firewall. Lets verify the ESI status on the Core switches.

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

We forgot to configure the ESI-LAG: 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 SRX facing EX9204 port.



Save the config and then verify the changes on the Core switch.

```
root@Core1> show lacp statistics interfaces
Aggregated interface: ae0
LACP Statistics:      LACP Rx      LACP Tx      Unknown Rx      Illegal Rx
xe-1/0/0              101          103           0                0

root@Core1> show configuration interfaces ae0 |display set |display inheritance
set interfaces ae0 hold-time up 120000
set interfaces ae0 hold-time down 1
set interfaces ae0 esi 00:11:11:11:11:11:01:00
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

root@Core1> show evpn database
Instance: evpn_vrf
VLAN  DomainId  MAC address      Active source      Timestamp      IP address
1      00:cc:34:f3:cf:00  192.168.255.32     Nov 07 23:13:46
1      00:cc:34:f4:72:00  192.168.255.31     Nov 07 23:13:34
1      f4:05:2f:f3:fb:f0  192.168.255.12     Nov 07 22:59:09
1      f4:05:2f:f4:0b:f0  192.0.0.0          Nov 07 22:59:10
11099  00:00:5e:e4:31:57  192.168.255.31     Nov 07 23:13:34  10.99.99.1
11099  52:54:00:74:a0:6f  192.168.255.31     Nov 07 23:16:31  10.99.99.99
11099  f0:1c:2d:c8:e8:f0  00:11:11:11:11:11:01:00  Nov 09 17:40:47  10.99.99.254
21088  00:00:5e:e4:31:57  192.168.255.31     Nov 08 15:29:02  10.88.88.1
21088  52:54:00:f7:12:2d  192.168.255.32     Nov 07 23:13:46  10.88.88.88
21088  f0:1c:2d:c8:e8:f0  00:11:11:11:11:11:01:00  Nov 09 17:40:55  10.88.88.254
21088  f4:07:39:0b:e3:20  192.168.255.32     Nov 08 04:21:53  10.88.88.10
31033  00:00:5e:e4:31:57  192.168.255.31     Nov 07 23:13:34  10.33.33.1
31033  5c:5b:35:2e:53:61  192.168.255.32     Nov 08 15:25:52
31033  5c:5b:35:af:29:d5  192.168.255.31     Nov 08 15:25:52
31033  f0:1c:2d:c8:e8:f0  00:11:11:11:11:11:01:00  Nov 09 17:40:52  10.33.33.254

root@Core1>
```

Note that LACP is up (this infers there is an existing configuration on the SRX firewall).

```
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>
```

Then confirm the EVPN data base 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 have been verified. Desktops communicate with each other through the Campus Fabric, each in an isolated VRF, then forwarded to the SRX firewall through the dual homing ESI-LAG on both Core1/2 for routing between VRFs or routing instances. Internet connectivity was also verified from each Desktop.

## EVPN Insights

Mist Wired Assurance provides the user with realtime 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 screenshots are taken from the Campus Fabric IP Clos build by accessing the Campus Fabric option under the Organization/Wired of the Mist Portal:

< Campus Fabrics : DC81-IPCloS

Core

Distribution

Access

Core1

MAC Address f4:b5:2f:f4:04:00  
 Model EX9204  
 Status connected  
 Site Primary Site  
 Router ID 192.168.255.11

VLANs

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

Connections to Distribution

Switch	RX Bytes	TX Bytes	Link Status
> Dist1	1.5 GB	950.4 MB	Up
> Dist2	2.6 GB	666.9 MB	Up

Remote Shell Switch Insights

< Campus Fabrics : DC81-IPCloS

Core

Distribution

Access

Dist1

MAC Address d8:53:9a:64:6f:c0  
 Model QFX5120-48Y  
 Status connected  
 Site Primary Site  
 Router ID 192.168.255.21

VLANs

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

Connections to Core

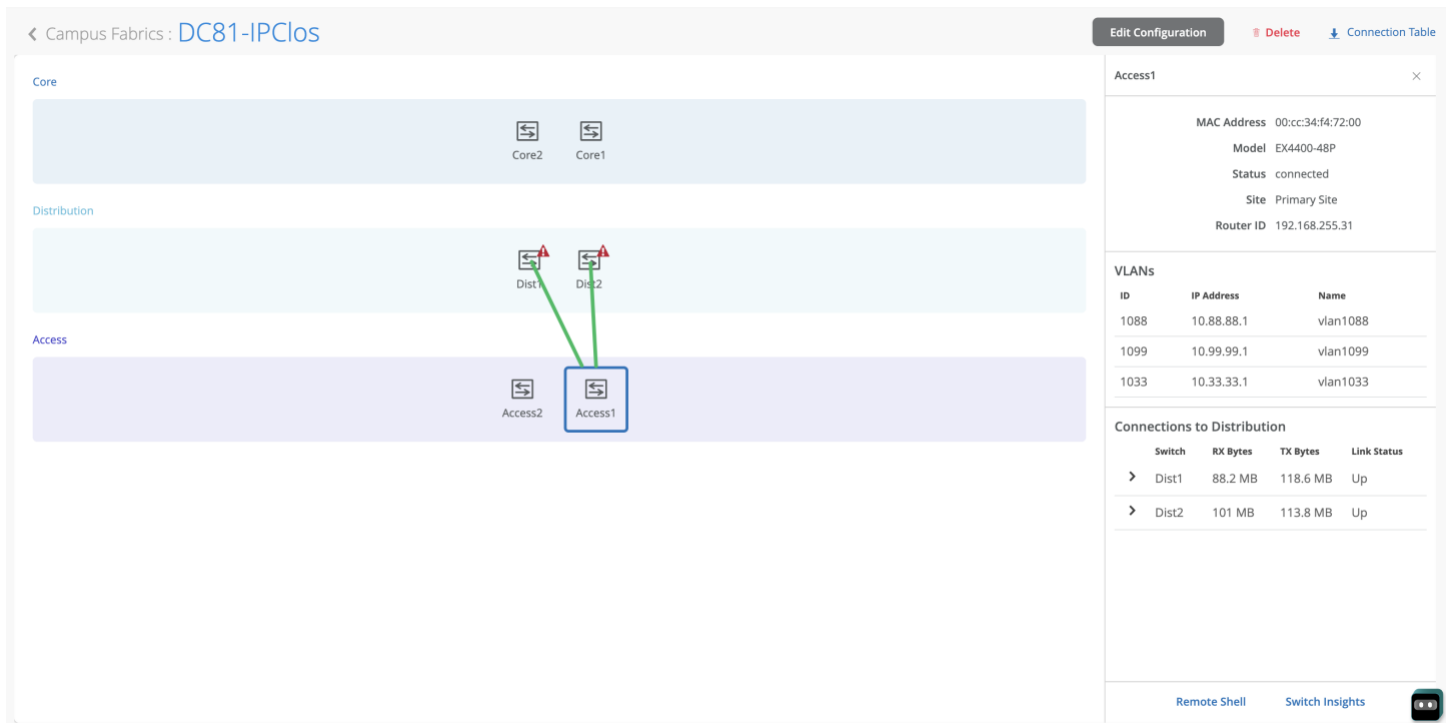
Switch	RX Bytes	TX Bytes	Link Status
> Core2	3.6 GB	33.7 GB	Up
> Core1	2.6 GB	28.5 GB	Up

Connections to Access

Switch	RX Bytes	TX Bytes	Link Status
> Access2	44.4 GB	688.5 MB	Up
> Access1	355.7 MB	487.4 MB	Up

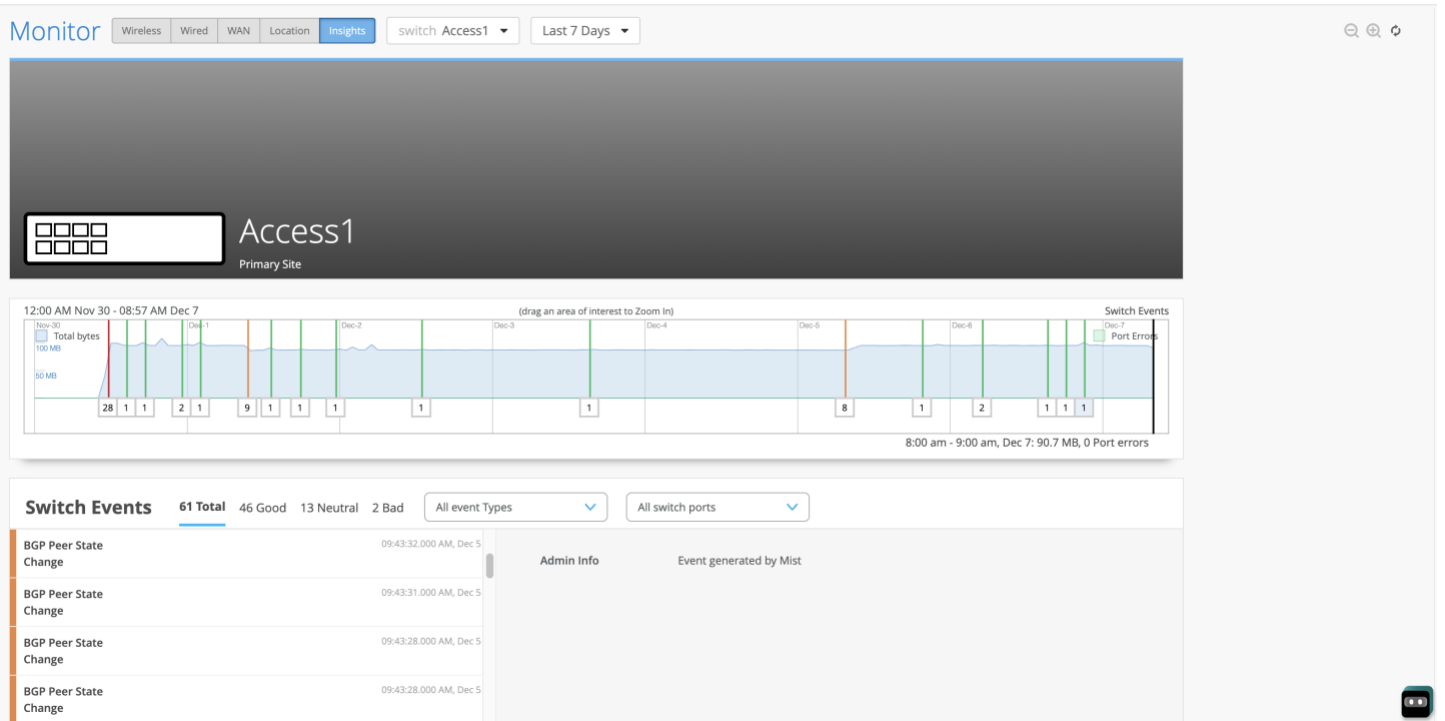
Remote Shell Switch Insights





From this view, Mist also provides remote accessibility into each device's console through the Remote Shell option as well as rich telemetry through the Switch Insights option. Remote Shell has been demonstrated throughout this document when displaying realtime operational status of each device during the verification stage.

Switch Insights of Access1 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 solely via Mist UI. Steps have been added to this document to help you understand the troubleshooting steps if deployment isn't working correctly.

## Appendix

### Campus Fabric IP Clos Configurations

Note: Those items in RED are specific to the device and require uniqueness

#### eBGP Underlay

##### Access1

```
set protocols bgp group evpn_underlay type external
set protocols bgp group evpn_underlay local-as 65006
set protocols bgp group evpn_underlay multipath multiple-as
set protocols bgp group evpn_underlay authentication-key
$9$gVojHq.5n6AaZn/tu1lwY24DiTz36ApoJDkP5F3M8L7wYaJDjqmZGp0O1yrwY2aJDfTz6CtQzCuBlrIGDjHfTAp0IR
Su0EylKx7Uji.TzFn/pu1CAWLXxdV.Pf5QntuORcyCALxdb2gJGDihmfTz3nCTQSreKx7P5TQ69BIEhre1Iclev7Nikqzmz3
set protocols bgp group evpn_underlay family inet unicast
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 log-updown
set protocols bgp group evpn_underlay export evpn_underlay_export
set protocols bgp group evpn_underlay import evpn_underlay_import
set protocols bgp group evpn_underlay neighbor 10.255.240.12 peer-as 65003
set protocols bgp group evpn_underlay neighbor 10.255.240.16 peer-as 65004
```

##### Dist1

```
set protocols bgp group evpn_underlay type external
set protocols bgp group evpn_underlay local-as 65003
set protocols bgp group evpn_underlay multipath multiple-as
set protocols bgp group evpn_underlay authentication-key $9$wLYZUji.Qz624QF/Cu0-VbsJGmfTz69YgJDk.5TIKv8-
V2gJZjH4o9AtuEh-
Vb2gJqmfzn/PfnCp0hcoJZUqm69A0ORCABECyW8aZGimf5QF9Cun6evMWx7ikq.PQ/CtOIE6vWxNbwgoJGUHqmf
TQnmPRhSyW8k.mPz3pOB1hSu0IcSr8LGDjHfT
set protocols bgp group evpn_underlay family inet unicast
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 log-updown
set protocols bgp group evpn_underlay export evpn_underlay_export
set protocols bgp group evpn_underlay import evpn_underlay_import
set protocols bgp group evpn_underlay neighbor 10.255.240.2 peer-as 65001
set protocols bgp group evpn_underlay neighbor 10.255.240.6 peer-as 65002
set protocols bgp group evpn_underlay neighbor 10.255.240.11 peer-as 65005
set protocols bgp group evpn_underlay neighbor 10.255.240.13 peer-as 65006
```

##### Core1

```
set protocols bgp group evpn_underlay type external
set protocols bgp group evpn_underlay local-as 65002
set protocols bgp group evpn_underlay multipath multiple-as
set protocols bgp group evpn_underlay authentication-key $9$deboJGUHf5FwYfT36AtxN-
V4ak.P5Fnbs4ZjHmPSrlvxNws4oGDY2n/9A1IxN-
ws4ik.5z3q.z6CtIR24oJikFn/tpB6/u1RhKvgoaUk.mfTn6AzFyleK8LUjiHqf369pO1zFIK8X-
ds24aJDik.PfzkqBIEhKvjHkq5QCtuOIEAtOREcvMaZGD.P
set protocols bgp group evpn_underlay family inet unicast
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 log-updown
set protocols bgp group evpn_underlay export evpn_underlay_export
set protocols bgp group evpn_underlay import evpn_underlay_import
set protocols bgp group evpn_underlay neighbor 10.255.240.7 peer-as 65003
set protocols bgp group evpn_underlay neighbor 10.255.240.9 peer-as 65004
```

#### eBGP Overlay in support of EVPN-VXLAN

## Access1

```
set protocols bgp group evpn_overlay type external
set protocols bgp group evpn_overlay local-address 192.168.255.31
set protocols bgp group evpn_overlay local-as 65006
set protocols bgp group evpn_overlay multipath multiple-as
set protocols bgp group evpn_overlay authentication-key
$9$gVojHq.5n6AaZn/tu1lwY24DiTz36ApoJDkP5F3M8L7wYaJDjqmZGp0O1yrwY2aJDfTz6CtQzCuBlrIGDjHfTAp0IR
Su0EylKx7Uji.TzFn/pu1CAWLXxdV.Pf5QntuORcyCALxdb2gJGDihmfTz3nCTQSreKx7P5TQ69BIEhre1lclev7Nikq mz3
set protocols bgp group evpn_overlay family evpn signaling
set protocols bgp group evpn_overlay log-updown
set protocols bgp group evpn_overlay bfd-liveness-detection minimum-interval 1000
set protocols bgp group evpn_overlay bfd-liveness-detection multiplier 3
set protocols bgp group evpn_overlay bfd-liveness-detection session-mode automatic
set protocols bgp group evpn_overlay multihop ttl 1
set protocols bgp group evpn_overlay multihop no-nexthop-change
set protocols bgp group evpn_overlay neighbor 192.168.255.21 peer-as 65003
set protocols bgp group evpn_overlay neighbor 192.168.255.22 peer-as 65004
```

## Dist1

```
set protocols bgp group evpn_overlay type external
set protocols bgp group evpn_overlay local-address 192.168.255.21
set protocols bgp group evpn_overlay local-as 65003
set protocols bgp group evpn_overlay multipath multiple-as
set protocols bgp group evpn_overlay authentication-key $9$wLYZUji.Qz624QF/Cu0-VbsJGmfTz69YgJDk.5TIKv8-
V2gJZjH4o9AtuEh-
Vb2gJqmfzn/PfnCp0hcoJZUqm69A0ORCABEcyW8aZGimf5QF9Cun6evMWx7ikq.PQ/CtOIE6vWxNbwgoJGUHqmf
TQnmPRhSyW8k.mPz3p0B1hSu0lcSr8LGDjHfT
set protocols bgp group evpn_overlay family evpn signaling
set protocols bgp group evpn_overlay log-updown
set protocols bgp group evpn_overlay bfd-liveness-detection minimum-interval 1000
set protocols bgp group evpn_overlay bfd-liveness-detection multiplier 3
set protocols bgp group evpn_overlay bfd-liveness-detection session-mode automatic
set protocols bgp group evpn_overlay multihop ttl 1
set protocols bgp group evpn_overlay multihop no-nexthop-change
set protocols bgp group evpn_overlay neighbor 192.168.255.12 peer-as 65001
set protocols bgp group evpn_overlay neighbor 192.168.255.11 peer-as 65002
set protocols bgp group evpn_overlay neighbor 192.168.255.32 peer-as 65005
set protocols bgp group evpn_overlay neighbor 192.168.255.31 peer-as 65006
```

## Core1

```
set protocols bgp group evpn_overlay type external
set protocols bgp group evpn_overlay local-address 192.168.255.11
set protocols bgp group evpn_overlay local-as 65002
set protocols bgp group evpn_overlay multipath multiple-as
set protocols bgp group evpn_overlay authentication-key $9$deboJGUHf5FwYfT36AtxN-
V4ak.P5Fnbs4ZjHmPSrlvxNws4oGDY2n/9A1lxN-
ws4ik.5z3q.z6CtIR24oJikFn/tpB6/u1RhKvgoaUk.mfTn6AzFyleK8LUjiHqf369pO1zFIK8X-
ds24aJDik.PfzkqBIEhKvjHkq5QCtu0IEAtOREcvMaZGD.P
set protocols bgp group evpn_overlay family evpn signaling
set protocols bgp group evpn_overlay log-updown
set protocols bgp group evpn_overlay bfd-liveness-detection minimum-interval 1000
set protocols bgp group evpn_overlay bfd-liveness-detection multiplier 3
set protocols bgp group evpn_overlay bfd-liveness-detection session-mode automatic
set protocols bgp group evpn_overlay multihop ttl 1
set protocols bgp group evpn_overlay multihop no-nexthop-change
set protocols bgp group evpn_overlay neighbor 192.168.255.21 peer-as 65003
set protocols bgp group evpn_overlay neighbor 192.168.255.22 peer-as 65004
```

---

## VXLAN and EVPN Enablement

### Access1

```
set groups top switch-options vtep-source-interface lo0.0
set groups top switch-options route-distinguisher 192.168.255.31:1
set groups top switch-options vrf-target target:65000:1

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
```

### Core1

```
set groups top routing-instances evpn_vrf vtep-source-interface lo0.0
set groups top routing-instances evpn_vrf route-distinguisher 192.168.255.11:1
set groups top routing-instances evpn_vrf vrf-target target:65000:1

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

Note: Distribution switches are lean switches and do not participate in VXLAN tunneling

## VLAN-VXLAN (VNI) mapping

### Access1

```
set vlans default vlan-id 1
set vlans default l3-interface irb.0
set vlans default vxlan vni 10001
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
```

### Core1

```
set groups top routing-instances evpn_vrf vlans default vlan-id 1
set groups top routing-instances evpn_vrf vlans default l3-interface irb.0
set groups top routing-instances evpn_vrf vlans default vxlan vni 10001
set groups top routing-instances evpn_vrf vlans vlan1033 vlan-id 1033
set groups top routing-instances evpn_vrf vlans vlan1033 vxlan vni 11033
set groups top routing-instances evpn_vrf vlans vlan1088 vlan-id 1088
set groups top routing-instances evpn_vrf vlans vlan1088 vxlan vni 11088
set groups top routing-instances evpn_vrf vlans vlan1099 vlan-id 1099
set groups top routing-instances evpn_vrf vlans vlan1099 vxlan vni 11099
```

---

## Routing Policy:

### All Devices

```
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 policy-options policy-statement evpn_underlay_export term 01-loopback from route-filter
192.168.255.0/24 orlonger
set groups top policy-options policy-statement evpn_underlay_export term 01-loopback then accept
set groups top policy-options policy-statement evpn_underlay_export term 02-default then reject
set groups top policy-options policy-statement evpn_underlay_import term 01-loopback from route-filter
192.168.255.0/24 orlonger
set groups top policy-options policy-statement evpn_underlay_import term 01-loopback then accept
set groups top policy-options policy-statement evpn_underlay_import term 02-default then reject
```

### L3 Interfaces (IRB)

#### Access1

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

## Routing Instances

#### Access1

```
set groups top routing-instances evpn_vrf instance-type virtual-switch
set groups top routing-instances evpn_vrf interface ae0.0
set groups top routing-instances evpn_vrf protocols rstp interface ae0 disable
set groups top routing-instances evpn_vrf protocols evpn encapsulation vxlan
set groups top routing-instances evpn_vrf protocols evpn default-gateway no-gateway-community
set groups top routing-instances evpn_vrf protocols evpn extended-vni-list all
set groups top routing-instances evpn_vrf vtep-source-interface lo0.0
set groups top routing-instances evpn_vrf route-distinguisher 192.168.255.11:1
set groups top routing-instances evpn_vrf vrf-target target:65000:1
set groups top routing-instances evpn_vrf vlans default vlan-id 1
set groups top routing-instances evpn_vrf vlans default l3-interface irb.0
set groups top routing-instances evpn_vrf vlans default vxlan vni 10001
set groups top routing-instances evpn_vrf vlans vlan1033 vlan-id 1033
set groups top routing-instances evpn_vrf vlans vlan1033 vxlan vni 11033
set groups top routing-instances evpn_vrf vlans vlan1088 vlan-id 1088
set groups top routing-instances evpn_vrf vlans vlan1088 vxlan vni 11088
set groups top routing-instances evpn_vrf vlans vlan1099 vlan-id 1099
set groups top routing-instances evpn_vrf vlans vlan1099 vxlan vni 11099
```

## Core1

```
set groups top routing-instances guest-wifi instance-type vrf
set groups top routing-instances guest-wifi interface irb.1033
set groups top routing-instances guest-wifi routing-options static route 0.0.0.0/0 next-hop [ 10.33.33.254 ]
set groups top routing-instances guest-wifi routing-options auto-export
set groups top routing-instances guest-wifi route-distinguisher 192.168.255.31:103
set groups top routing-instances guest-wifi vrf-target target:65000:103
set groups top routing-instances guest-wifi vrf-table-label
set groups top routing-instances guest-wifi 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 developers instance-type vrf
set groups top routing-instances developers interface irb.1088
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 auto-export
set groups top routing-instances developers route-distinguisher 192.168.255.31:102
set groups top routing-instances developers vrf-target target:65000:102
set groups top routing-instances developers vrf-table-label
set groups top routing-instances developers 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 corp-it instance-type vrf
set groups top routing-instances corp-it interface irb.1099
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 auto-export
set groups top routing-instances corp-it route-distinguisher 192.168.255.31:101
set groups top routing-instances corp-it vrf-target target:65000:101
set groups top routing-instances corp-it vrf-table-label
set groups top routing-instances corp-it 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
```

## ESI-LAG between Core1/2 and SRX Firewall

### Core1

```
set interfaces ae0 apply-groups esilag
set interfaces ae0 esi all-active
set interfaces ae0 esi 00:11:00:00:00:01:00:01:02:00
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 groups esilag interfaces <*> unit 0 family ethernet-switching interface-mode trunk
set groups esilag interfaces <*> unit 0 family ethernet-switching vlan members [ all ]
set groups evpn_downlink interfaces <*> mtu 9192
```

### SRX

```
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
```