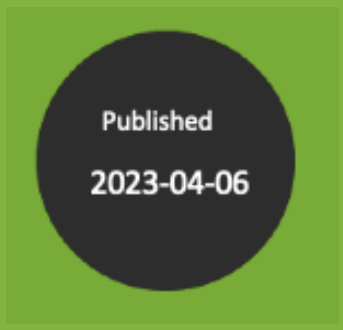
JUNIPER NETWORKS | Engineering Simplicity

Network Configuration Example

Campus Fabric IP Clos Workflow



Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Network Configuration Example Campus Fabric IP Clos Workflow

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement (“EULA”) posted at https://support.juniper.net/support/eula/. By downloading, installing, or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

Campus Fabric IP Clos Workflow 4

About This Configuration Example 4
Scope 4
Documentation Feedback 4
Technology Primer: Campus Fabric IP Clos 4
Use Case Overview 4
Benefits of Campus Fabric: IP Clos 5

Juniper Mist Wired Assurance 7

Campus IP Clos Fabric High Level Architecture 7

Campus Fabric IP Clos Components 8
Juniper Mist Wired Assurance 9
Juniper Mist Wired Assurance Switches Section 10
Templates 10
Topology 11
Create the Campus Fabric 12
Campus Fabric Org Build 12
Campus Fabric Site Build 12
Choose the campus fabric topology 13
Select campus fabric nodes 15
Configure Networks 16
Other IP Configuration 18
Configure campus fabric ports 21
Core Switches 22
Distribution Switches 23
Access Switches 25
Campus Fabric Configuration Confirmation 25

VERIFICATION 30

BGP Underlay 31
EVPN VXLAN verification between Access and Core switches 34
Verification of the EVPN Database on both access switches 35
Verification of VXLAN tunnelling between Access and Core devices 36
External Campus Fabric connectivity through the Border GW Core EX9204 switches 39

EVPN Insights 42

Summary 45

Appendix 46

Configuration of the Underlay IP Fabric 46
Configuration of the EVPN VXLAN Overlay and Virtual Networks 51
Configuration of the Layer 2 ESI-LAG between the core switches and SRX firewall 58

Campus Fabric IP Clos Workflow

About This Configuration Example

Scope

Use this network configuration example (NCE) of Juniper's Campus Fabric IP Clos workflow

This example covers the workflow associated with building a Campus Fabric IP Clos

Documentation Feedback

We encourage you to provide feedback so that we can improve our documentation. Send your comments to design-center-comments@juniper.net. Include the document or topic name, URL or page number, and software version (if applicable).

Technology Primer: Campus Fabric IP Clos

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 (Internet of Things) 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 (Artificial Intelligence) 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. MC-LAG (multi-chassis link aggregation group) is a good example of a single-vendor technology that addresses the collapsed core deployment model. In this model, 2 chassis-based platforms are typically in the core of a customer's network; deployed to handle all L2/L3 requirements while providing an active/backup resiliency environment. MC-LAG does not interoperate between vendors, creating lock-in, and is limited to 2 devices.

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 a Layer 3 IP-based underlay network and an EVPN-VXLAN overlay network. Broadcast, unknown unicast, and multicast, commonly known as BUM (Broadcast, Unknown unicast, and Multicast) traffic, is handled natively by EVPN 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. ESI-LAG utilizes standards-based LACP (Link Aggregation Control Protocol) to interconnect with legacy switches.

Benefits of Campus Fabric: IP Clos

With the 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 learning mechanisms inherent with ethernet switching technologies. The traditional ethernet switching approach is inefficient because it leverages broadcast and multicast technologies to announce MAC (Media Access Control) 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 considering 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 VTEPs (VXLAN Tunnel Endpoint) that encapsulate and decapsulate the VXLAN traffic. VTEP (VXLAN Tunnel Endpoint) stands for VXLAN tunnel endpoint and represents the 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, by stretching the Layer 2 VXLAN network with routing occurring in the access device instead of the Core or Distribution layers. IP Clos network encompasses the distribution, core, and access layers of your topology.

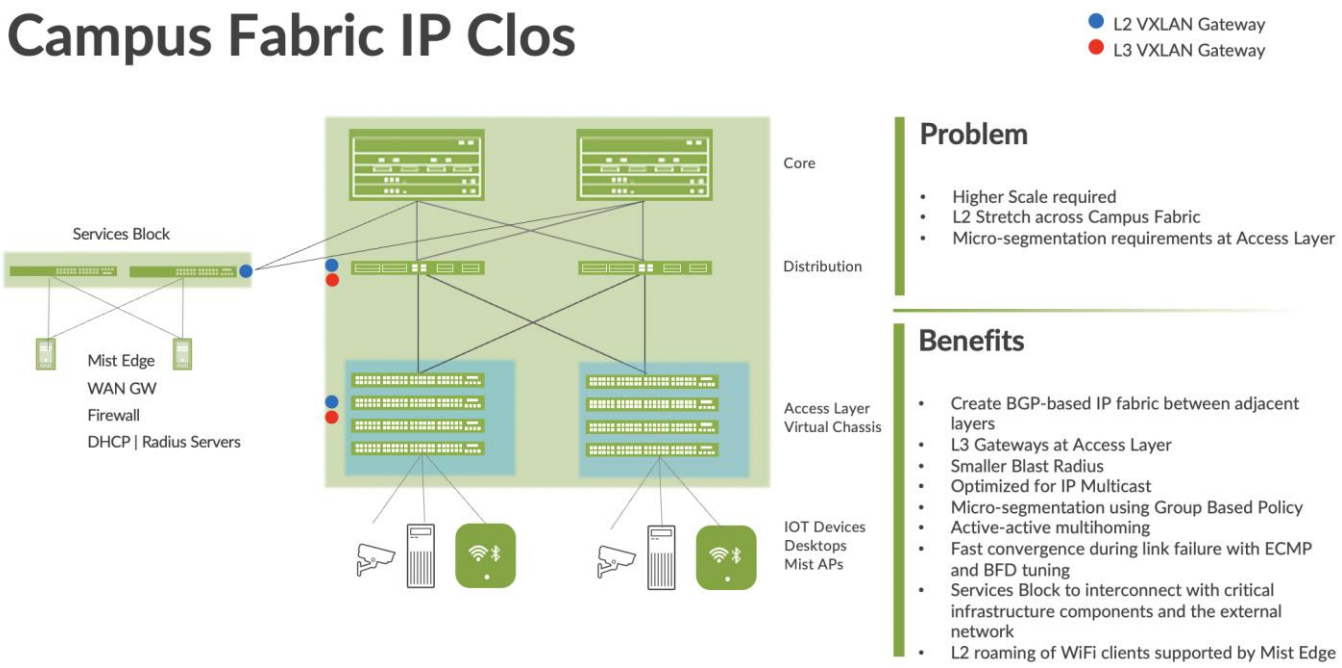


Figure 1 Campus fabric IP Clos

An EVPN-VXLAN fabric solves the problems of previous architectures 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 leaves 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. For example, in a Campus Fabric IP Clos, core switches do not learn the device endpoint addresses, rather they only learn the addresses of the Access layer switches.
- Consistency—A universal EVPN-VXLAN-based architecture across disparate campus and data-center deployments enables a seamless end-to-end network for endpoints and applications.
- Group Based Policies - With GBP you can enable micro-segmentation 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 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. Juniper uses optimal BGP timers between the adjacent layers of the Campus Fabric with BFD (Bidirectional Forwarding Detection) (fast convergence in case of a node or link failure) and ECMP (Equal cost multipath).
  https://www.juniper.net/documentation/us/en/software/junos/sampling-forwarding-monitoring/topics/concept/policy-configuring-per-packet-load-balancing.html

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 is 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 rich Junos streaming telemetry that enable the insights for switch health metrics and anomaly detection, as well as Mist AI capabilities.

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

Mist Cloud services are 100% programmable using open APIs (Application Programming Interfaces) (Application Programming Interface) for full automation and/or integration with your Operational Support Systems, such as: IT applications, such as Ticketing Systems, IP Management Systems, etc.

Juniper Mist delivers unique capabilities for the WAN (Wide Area Network), LAN (Local Area Network) and Wireless networks

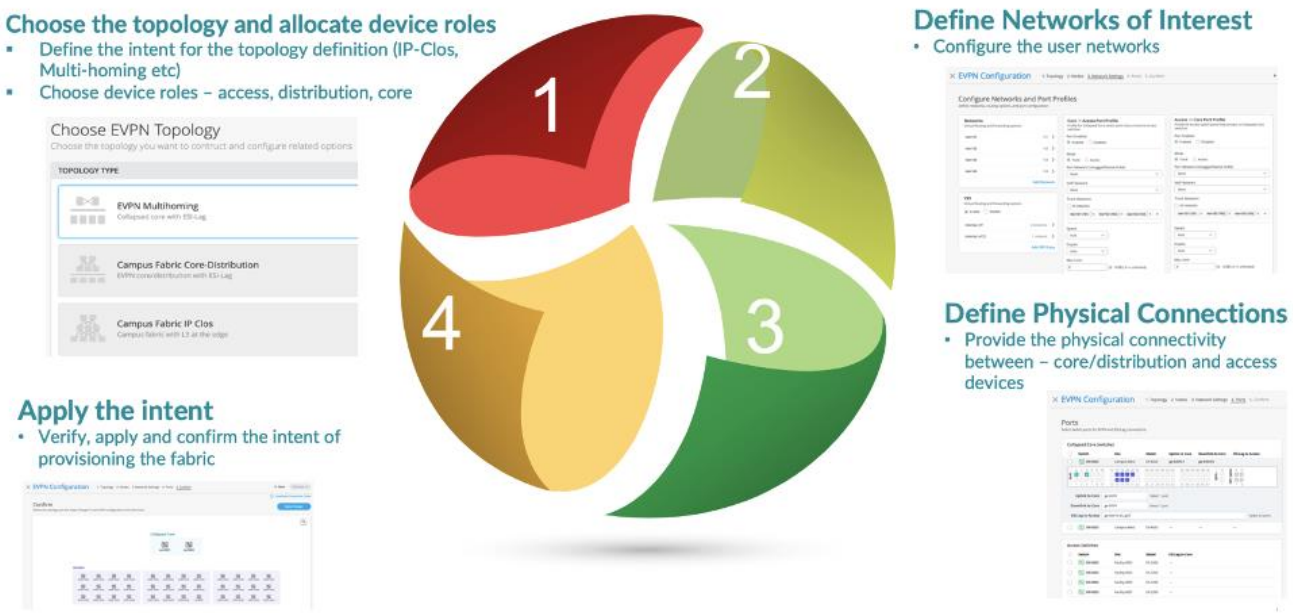
- UI (User Interface) or API (Application Programming Interface) driven configuration at scale
- Service Level Expectations (SLE) for key performance metrics such as throughput, capacity, roaming, and uptime.
- Marvis—An integrated AI engine that provides rapid troubleshooting of Full Stack network issues, 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 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. In a Campus Fabric deployment, the use of EVPN VXLAN supports native traffic isolation using routing-instances; commonly called VRFs (Virtual Routing and Forwarding) for macro-segmentation purposes.

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



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 wired clients

NOTE: Advanced GBP micro-segmentation features require 22.4R1 and later code

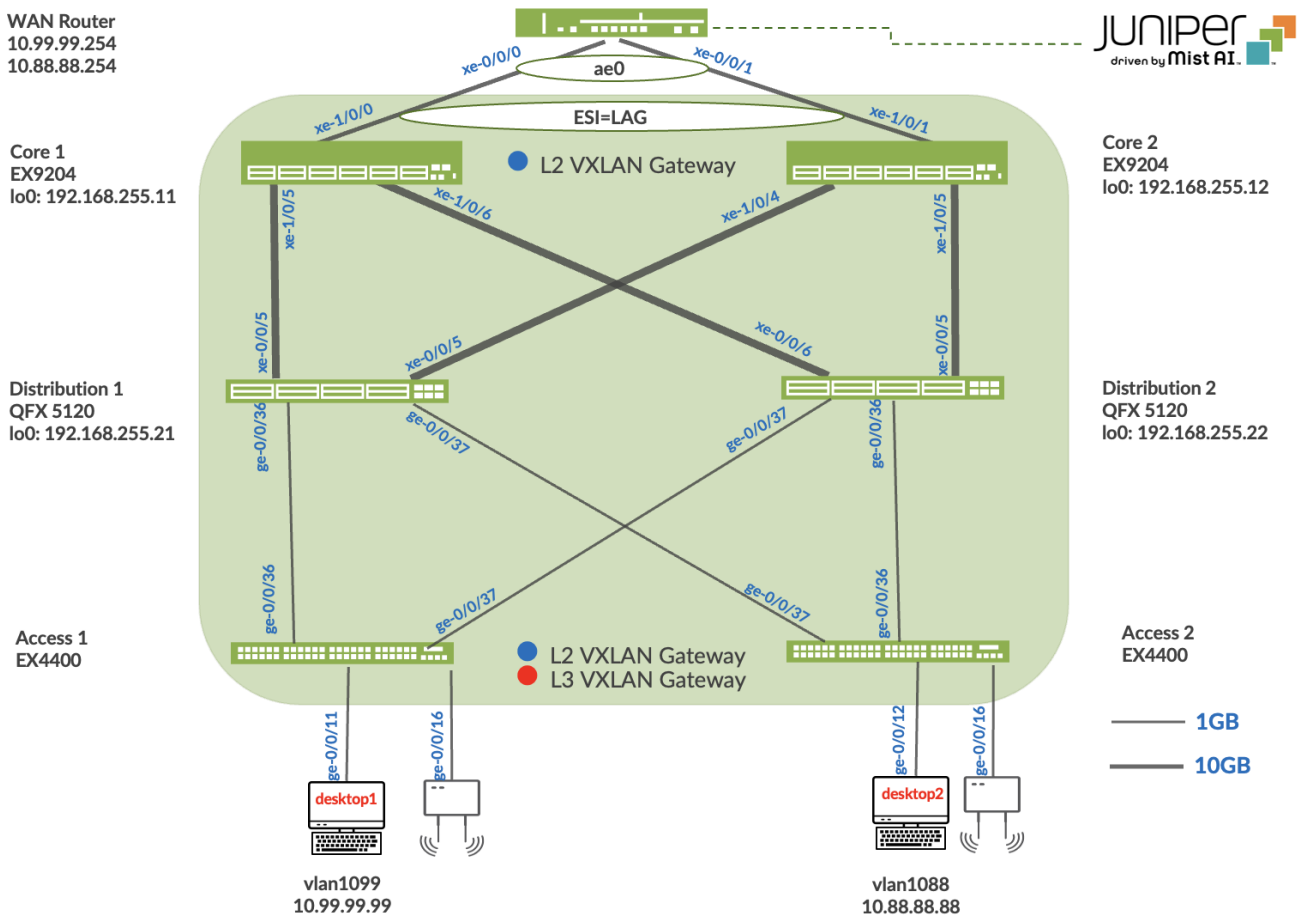


Figure 2. Topology

Juniper Mist Wired Assurance

Wired Assurance, through the Mist UI, can be used to centrally manage all Juniper switches. Juniper Mist 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 including switch version and PoE (Power Over Ethernet) compliance, switch-AP affinity, and VLAN (Virtual LAN) insights.

Juniper Switch Onboarding to the Mist Cloud:

https://www.juniper.net/documentation/us/en/software/nce/nce-214-midsize-branch-mist-pwp/topics/topic-map/nce-214-midsize-branch-mist-example_part2.html

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 (Virtual Routing and Forwarding) 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 (Layer 3) gateway IRB (Integrated Routing and Bridging)
- IP addressing of each lo0.0 loopback
- Configuration of routing policies for underlay and overlay connectivity
- Optimized MTU (Maximum Transmission Unit) 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 descriptively named to represent the respective layers in the fabric to facilitate building and operating the fabric.

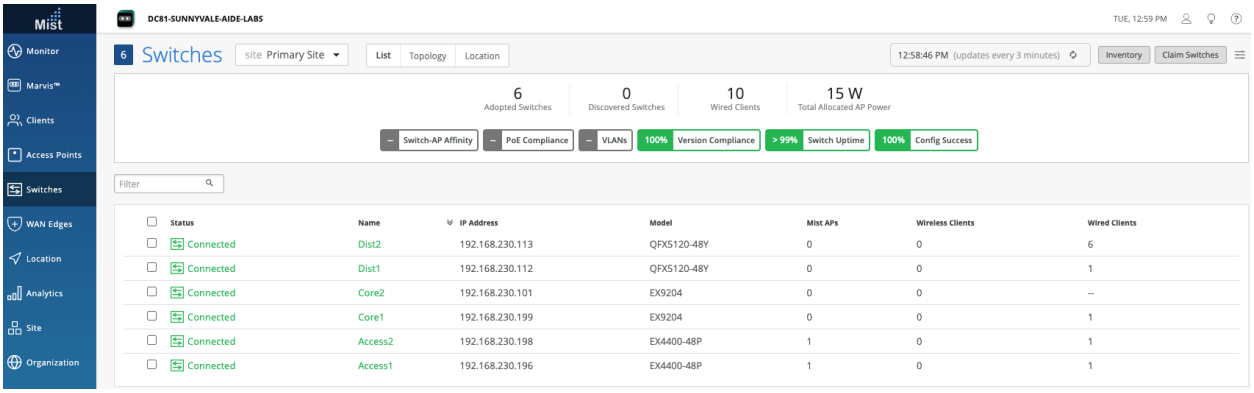


Figure 3. Switch Inventory

Templates

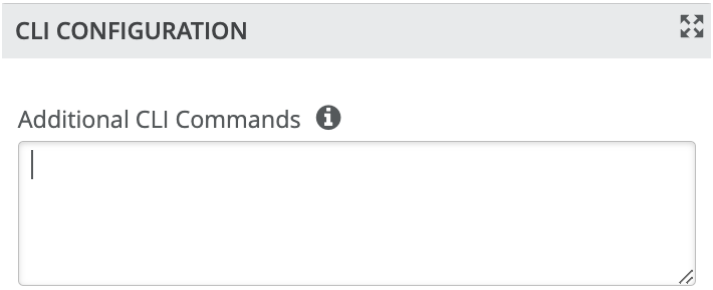
A key feature of switch management through the Juniper Mist cloud is the ability to use templates and a hierarchical model to group the switches and make bulk updates. Templates provide uniformity and convenience, while the hierarchy (Organization, Site, and Switch) provides both scale and granularity.

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 Site and Organizational 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. Of course, individual switches can also have their own unique configurations.

You can include individual CLI (Command Line Interface) 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 replace 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. The CLI Command option can be access from the Switch Template or individual Switch configuration:



Under Organization and Switch Templates, we utilize the following template.

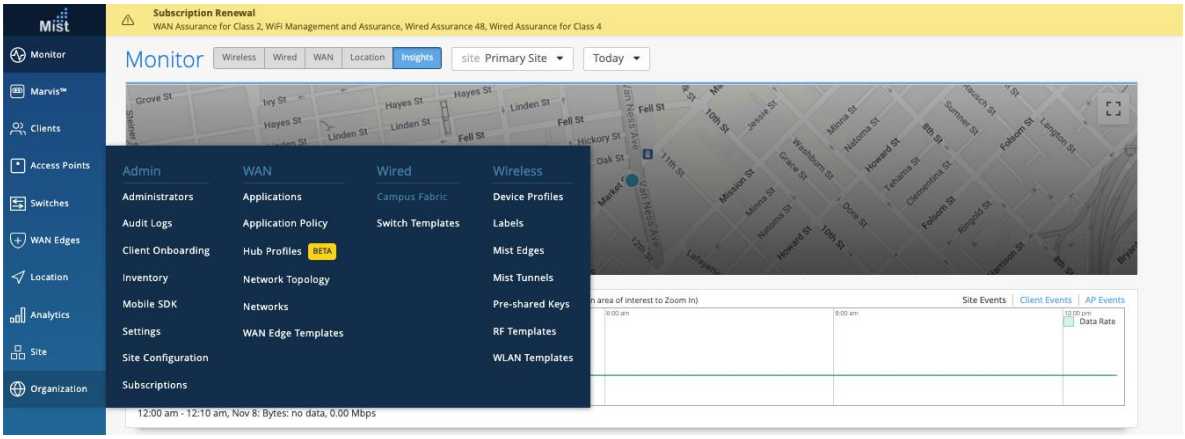
Topology

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 routers can be standalone or built as an HA (High Availability) cluster.

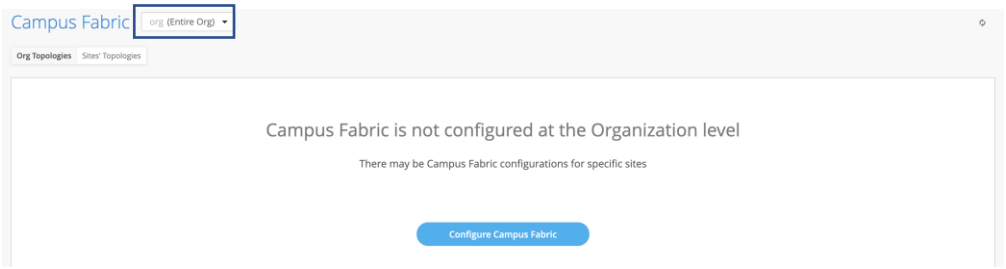
Create the Campus Fabric

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

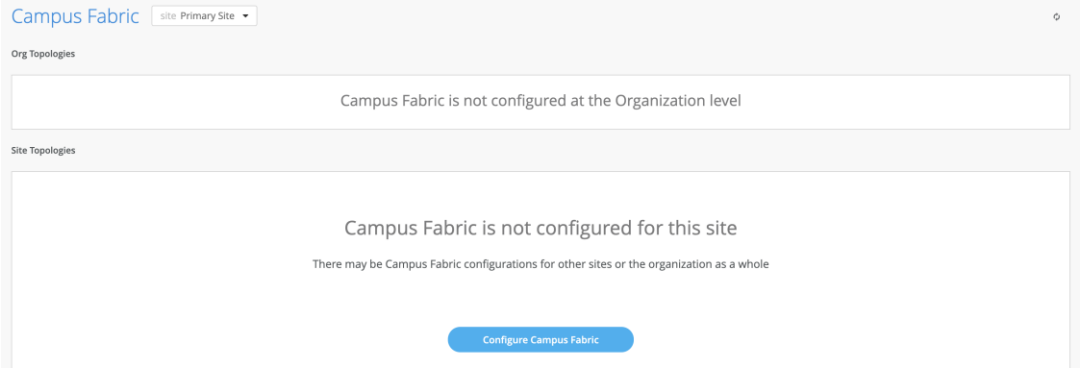


Mist provides the option of deploying a Campus Fabric at the Org or Site level noted on the upper left hand Campus Fabric pull down menu shown below. For example, those who are building a Campus wide architecture with multiple buildings, each building housing distribution and access switches, could consider 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



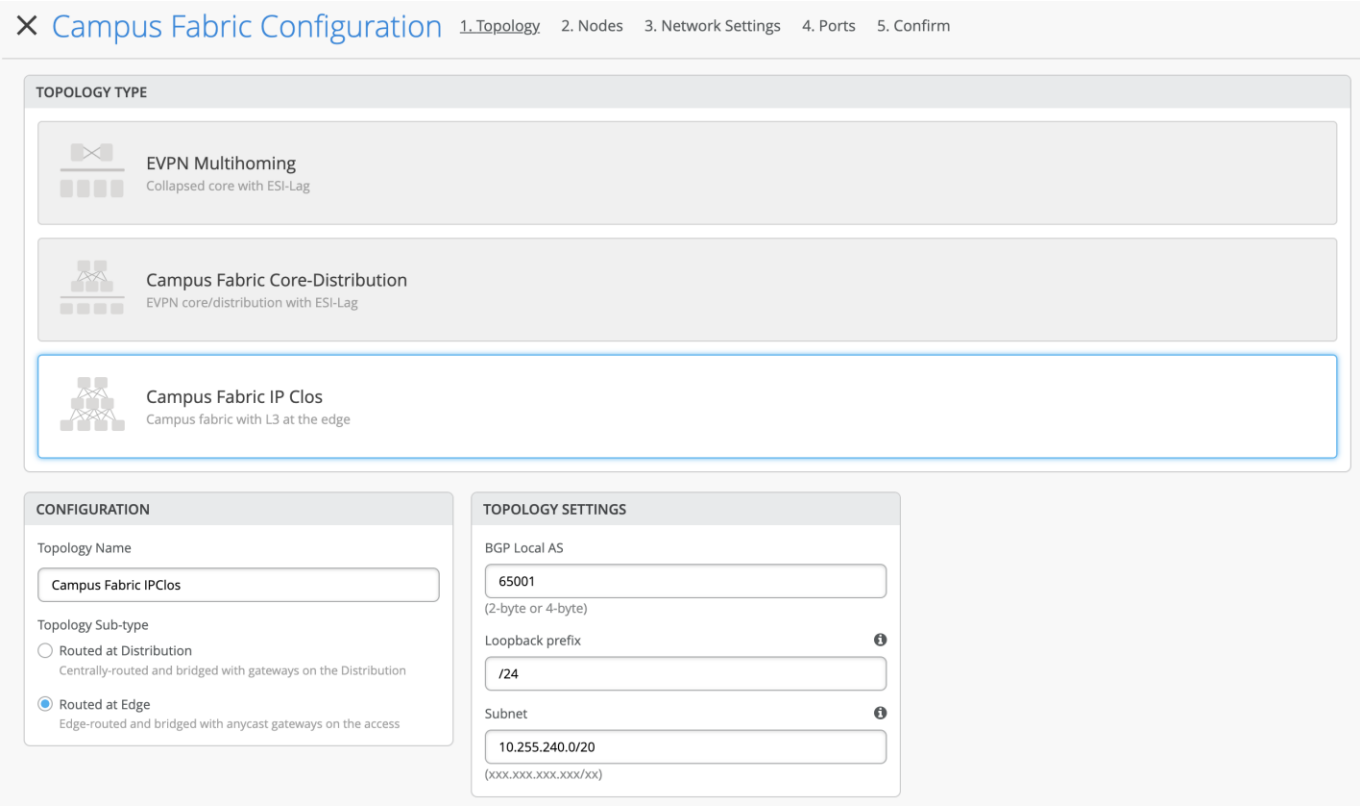
Campus Fabric Site Build



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

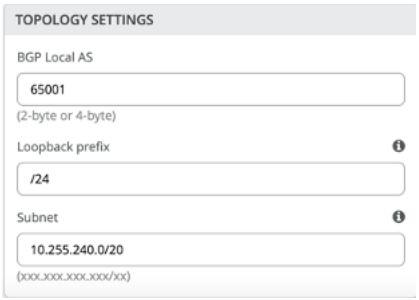
- Provide a name in accordance with company standards

NOTE: 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.

NOTE: Routed at Distribution aligns with north-south traffic patterns and configures the Access layer 2 VXLAN gateways only. This deployment is preferred for higher scale environments.

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 using 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.



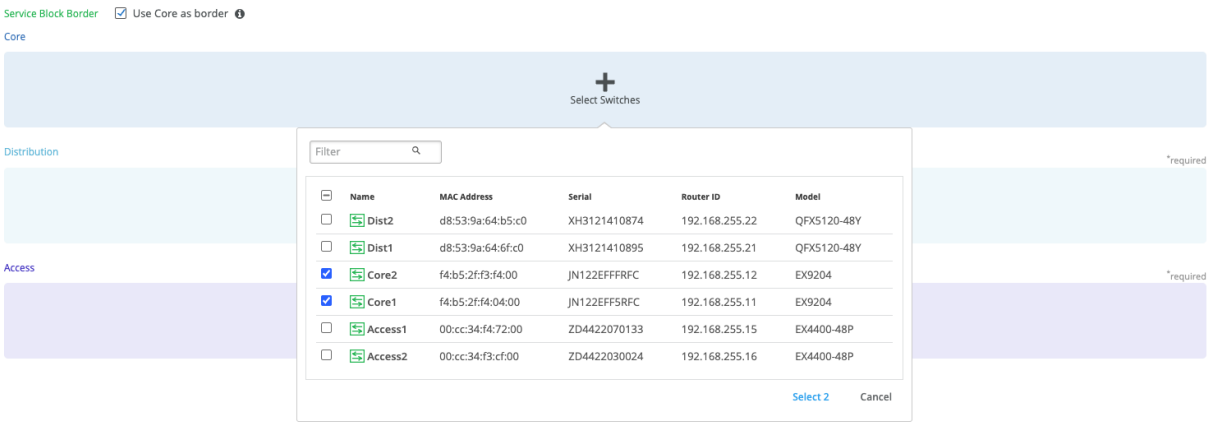
NOTE: Juniper recommends default settings for all options unless it conflicts with the surrounding environment. The point-point links between each layer utilize /31 addressing to conserve addresses.

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.

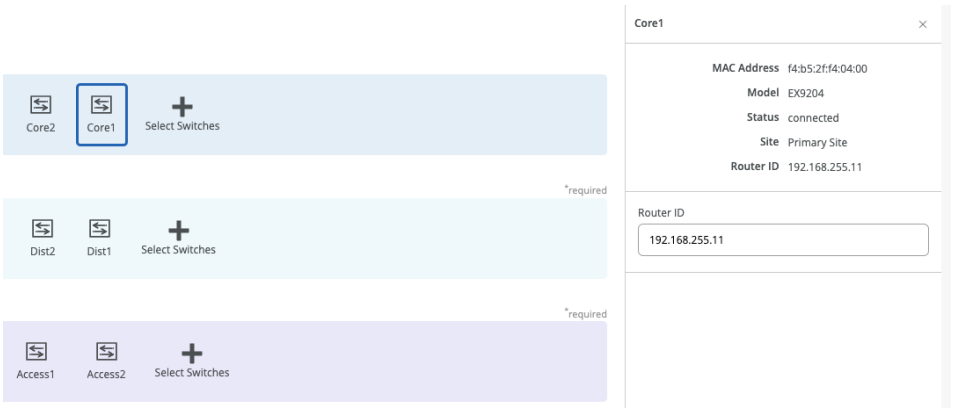
Services Block Router is where the Campus Fabric would interconnect external devices such as firewalls, routers, or critical devices such as DHCP and Radius servers (as an example). Devices to which external services connect to the Campus Fabric are known as Border Leafs. If the user wishes to connect 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 functionality on a dedicated pair of switches (recommended for resiliency) 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 Core as a border option (this is option is chosen per this document)

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 router-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 next hops. 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.

Configure Networks

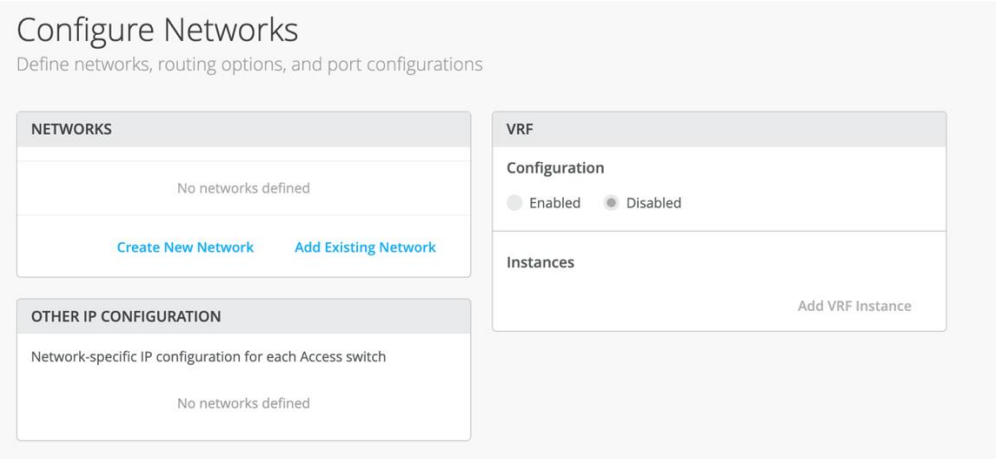
Mist presents the user with input for Network information such as VLANs and VRF (routing instances for traffic isolation purposes) options. VLANs are mapped to VNIs (Virtual Network Identifier) and can optionally be mapped to VRFs to provide customers a way to logically separate traffic patterns such as IoT devices from Corp IT.

VRF

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

Routing Instance Overview:
https://www.juniper.net/documentation/us/en/software/junos/routing-overview/topics/concept/routing-instances-overview.html

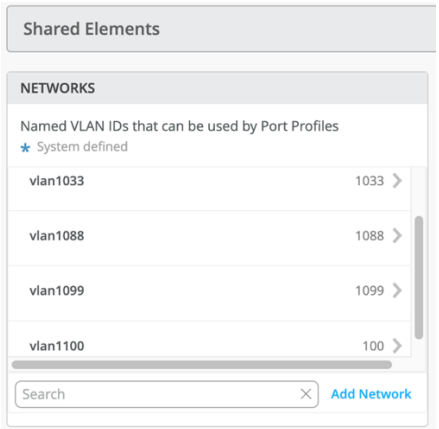
VLANs can be placed into a common VRF where all VLANs within each VRF have full connectivity amongst themselves and other external networking resources. A common use case is the isolation of Guest Wi-Fi traffic from most Enterprise domains save Internet connectivity. By default, the Campus Fabric provides complete isolation between VRFs forcing inter-VRF communications to traverse a Firewall or security compliance. This aligns with most Enterprise security use-cases and compliance and is represented in this document.



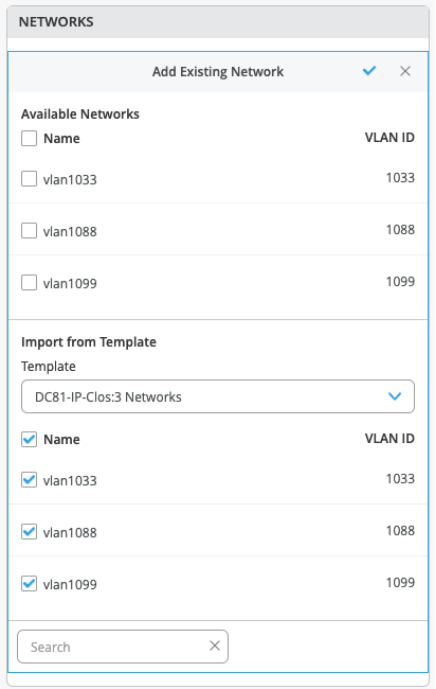
Networks

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

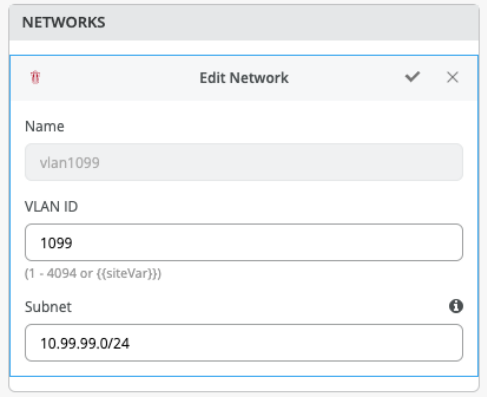
The Shared Elements section of the campus-fabric template includes the Networks section mentioned above where VLANs are created. This can be found under the Organization/Switch Templates section, then choose the appropriate template:



Back to the Campus Fabric build, the user selects the “Add Existing Network” option that includes L2 (Layer 2) VLAN information. All VLAN and IP information will be inherited from the template



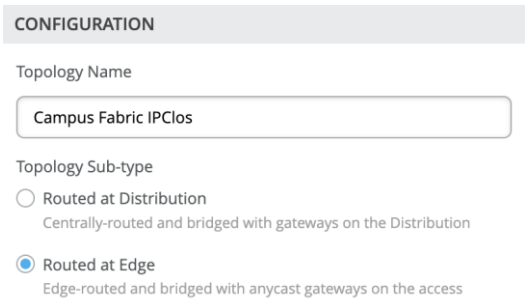
Networks can be edited, added from scratch or from an existing template:



Other IP Configuration

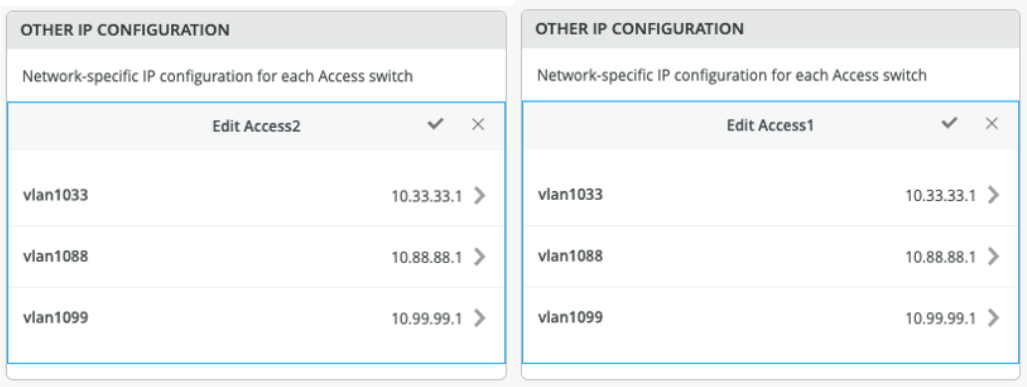
Mist Wired Assurance provides automatic IP addressing (IRBs (Integrated Routing and Bridging)) for each of the VLANs. Port Profiles and Port Configuration then associate the VLAN with specified ports. In this case, we selected IP Clos

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

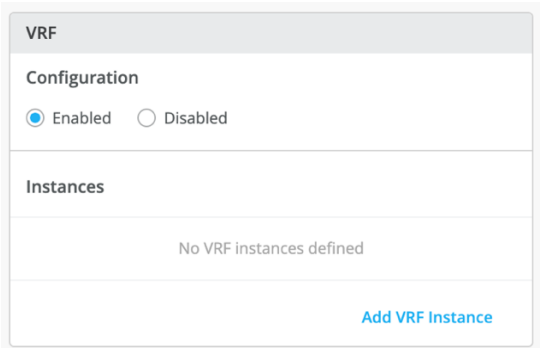


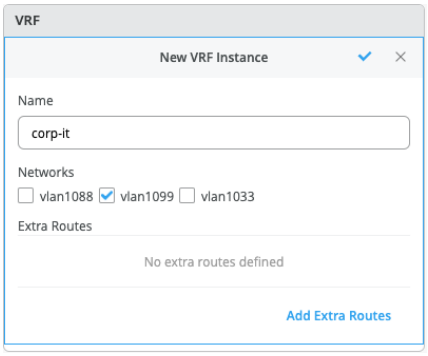
This option utilizes anycast addressing for all devices participating in the L3 subnet. In this case, Access1 and Access2 switches will be configured with the same IP address for each L3 subnet.

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



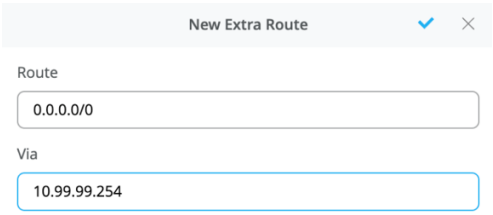
By default, all VLANs are placed in the default VRF. The VRF option allows the user to group common VLANs into the same VRF or separate VRFs depending on traffic isolation requirements. This example includes 3 VRFs or routing instances: corp-it | developers | guest-wifi. Here, the user builds the first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





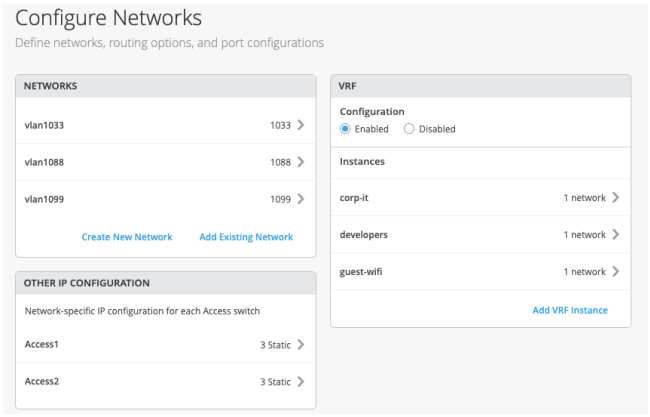
By default, inter-VRF communications is not supported within the Campus Fabric. If inter-VRF communications is required, each VRF can include extra routes such as a Default Route that will instruct the Campus Fabric to use an external router or firewall for further security inspection or routing capabilities. In this example, all traffic is trunked over the ESI-LAG and the Juniper SRX handles inter-VRF routing. Figure 2. Topology

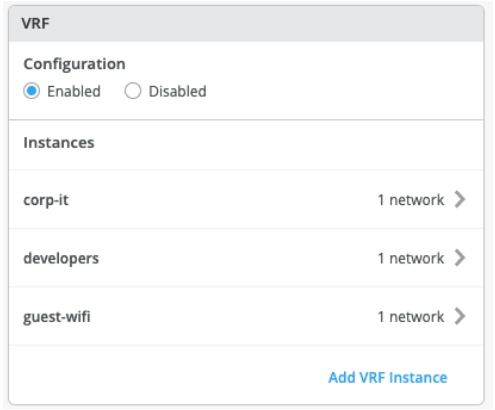
Notice the SRX participates in the VLANs defined within the Campus Fabric and is the gateway of last resort for all traffic leaving the subnet. The user selects the “Add Extra Routes” option to inform Mist to forward all traffic leaving 10.99.99.0/24 to utilize the next hop of the Juniper SRX firewall: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



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

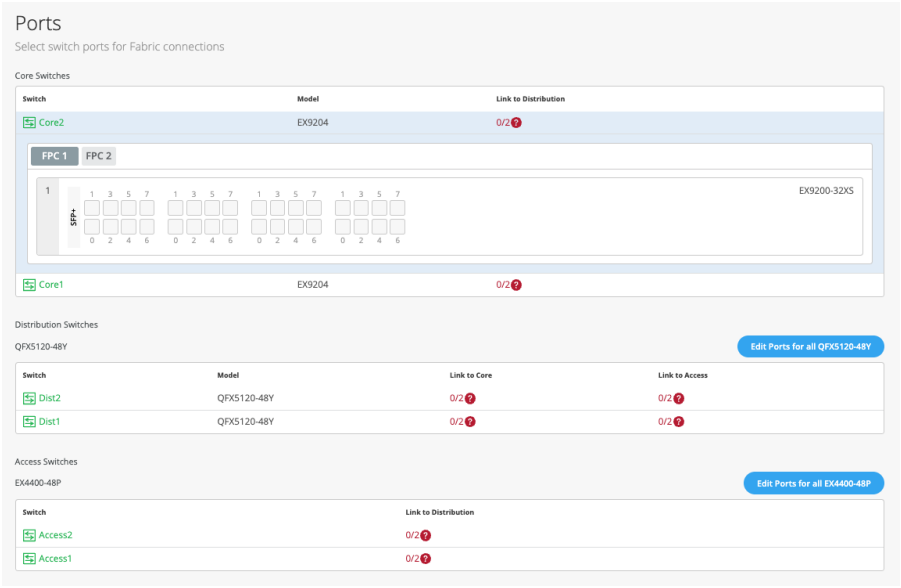




Now that all VLANs are configured and assigned to each VRF, the user can move to the next step by clicking the Continue button at the upper right section of the Mist UI.

Configure campus fabric ports

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

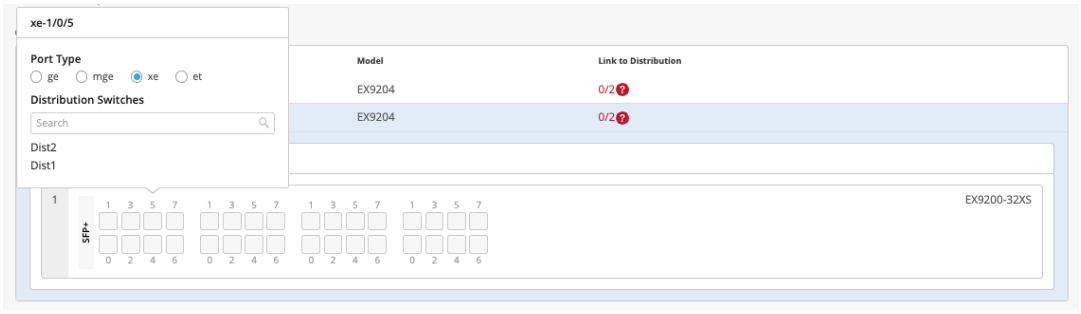


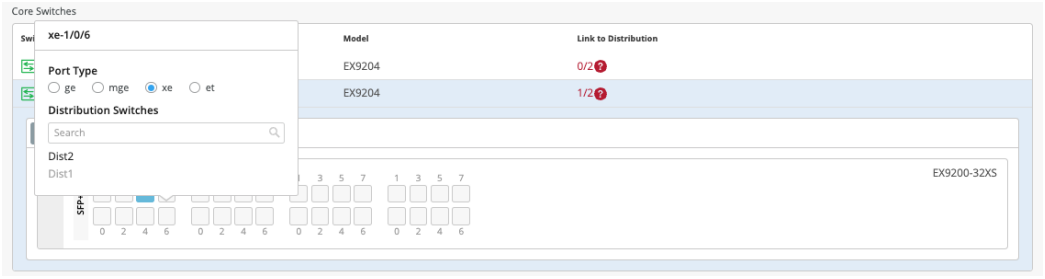
Note: Juniper recommends the user have the output of the show lldp neighbors command from each switch. If a Juniper enables LLDP (Link Layer Discovery Protocol) out of the box and provides additional LLDP attributes when the switch is added to a Campus Fabric. This output provides a source of truth for which ports should be selected at each layer.

Core Switches

Core1:

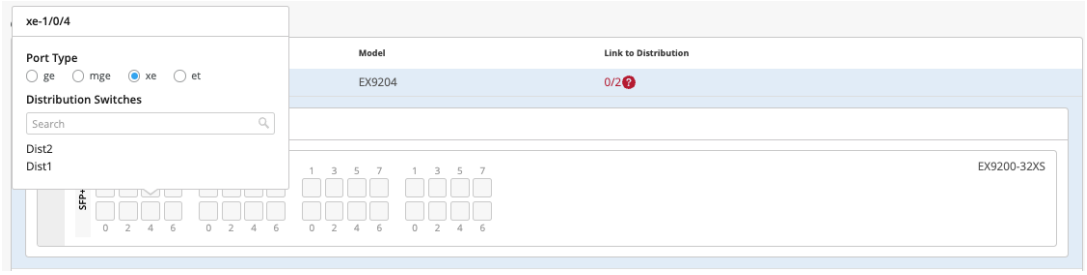
Starting with Core1, the user selects xe-1/0/5 and xe-1/0/6 terminating on Distribution Switches 1 and 2 respectively.

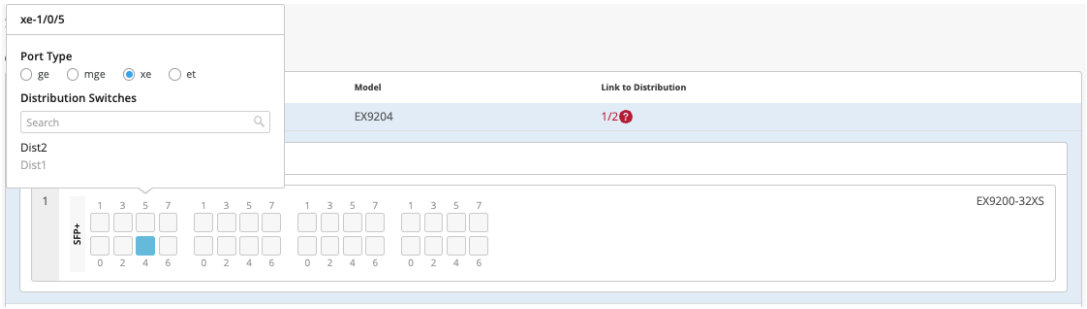




Core2:

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





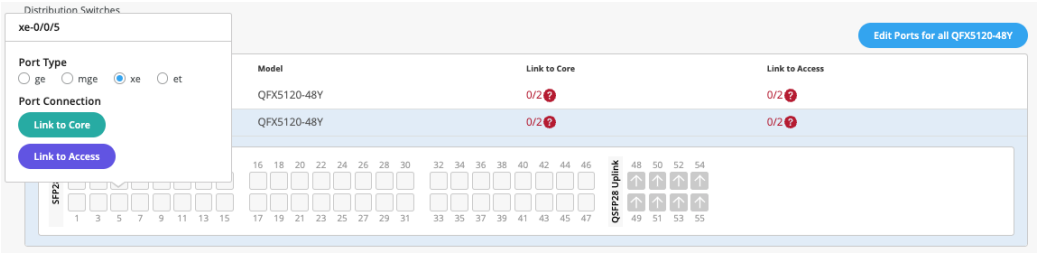
Distribution Switches

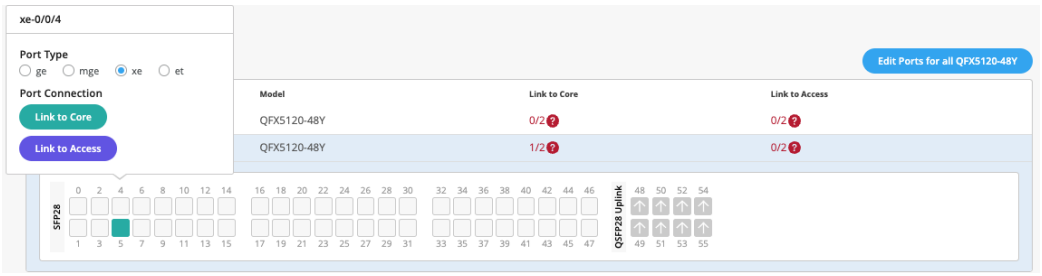
Now moving on to the Distribution Switches, you will notice 2 interconnect options exist

- Link to Core
- Link to Access

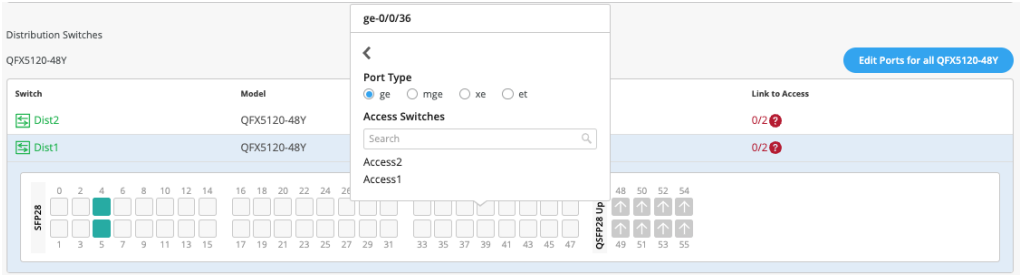
Dist1:

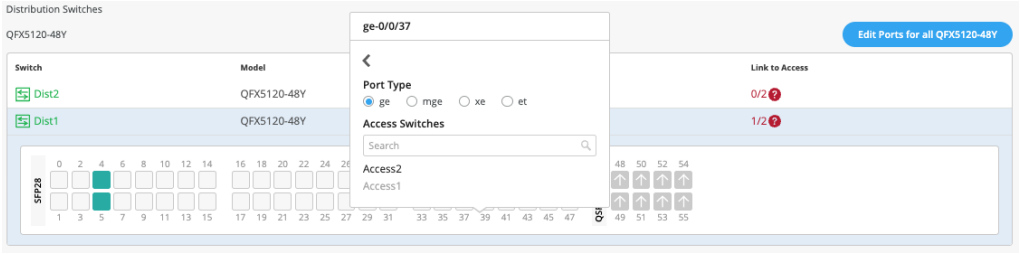
The user selects Link to Core and chooses xe-0/0/5 and xe-0/0/4 terminating on Core Switches 1 and 2 respectively.





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

NOTE: Juniper's QFX 5120-48Y is an example of a switch that is targeted for the Distribution layer in a Campus Fabric. This device supports blocks of 4 ports per PHY; Ports 0-3, 4-7, etc. All ports within the same PHY must operate at the same speed.

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

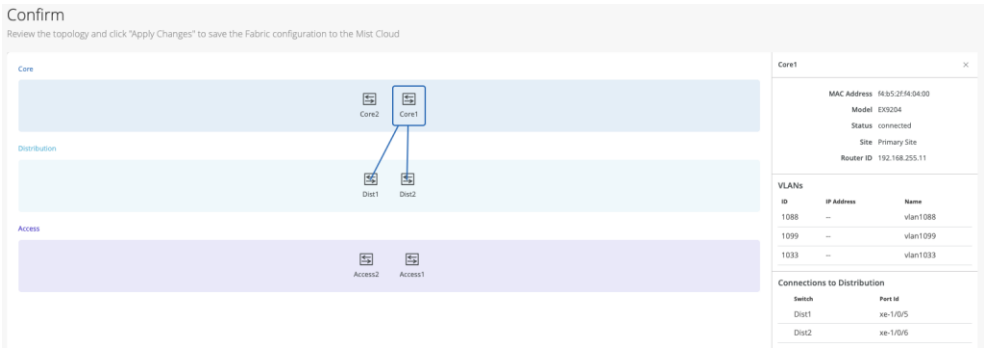
Access2:

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

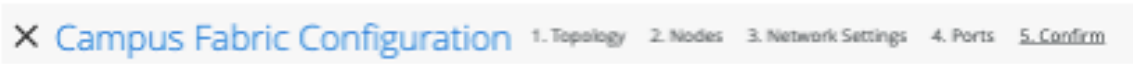
Once the user has completed selecting all requisite port combinations, they will select the Continue button at the upper right-hand corner of the Mist UI.

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

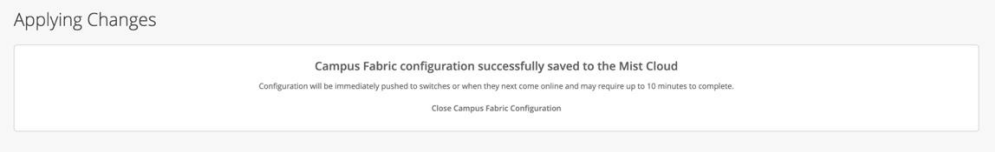


The user is presented with a second stage confirmation, confirm to create the fabric.

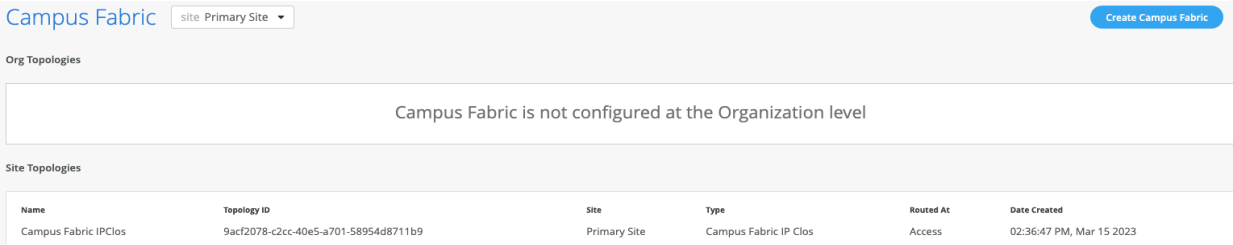
Mist presents the user with the following banner including the estimated time for the Campus Fabric to be built. The process includes the following:

- Mist builds the point-to-point interfaces between all devices with IP addresses chosen from the range presented at the onset of the build.

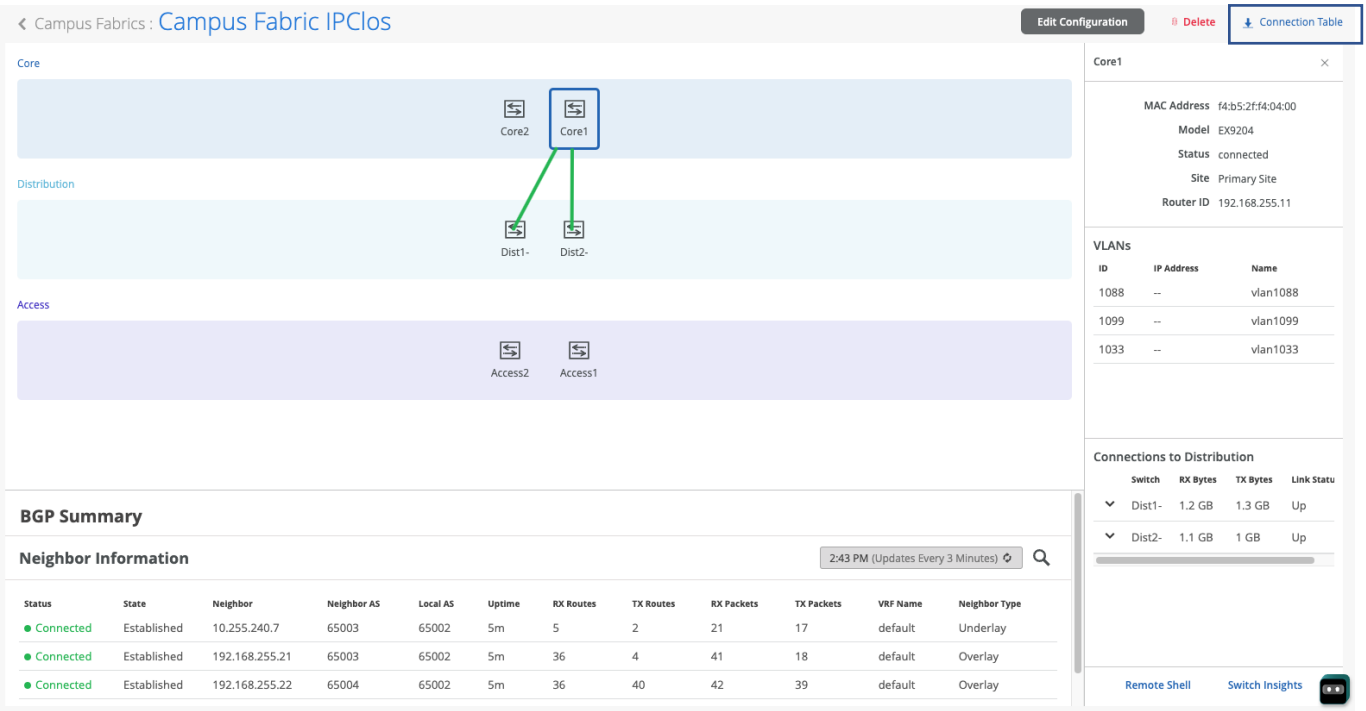
- Each device is configured with a loopback address from the range presented at the onset of the build.
- eBGP is provisioned at each device with unique BGP autonomous system numbers. The primary goal of the underlay is to leverage ECMP for load balancing traffic on a per packet level for device loopback 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 (Virtual Network Identifier) 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:



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	JN122EFF5RFC	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	JN122EFF5RFC	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

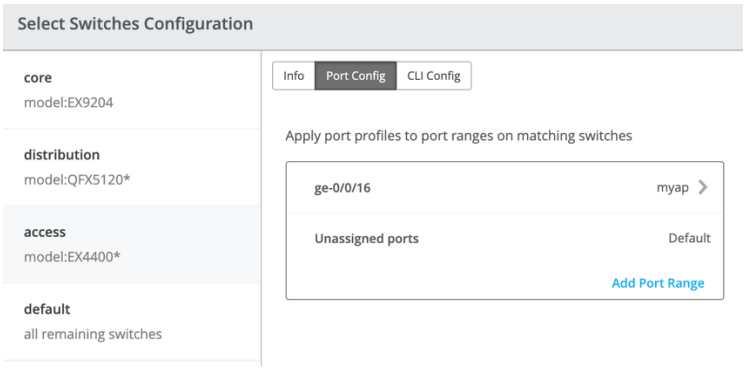
Apply VLANs to Access ports

As discussed, Mist can templatize well-known services such as Radius, NTP, DNS (Domain Name System), 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.

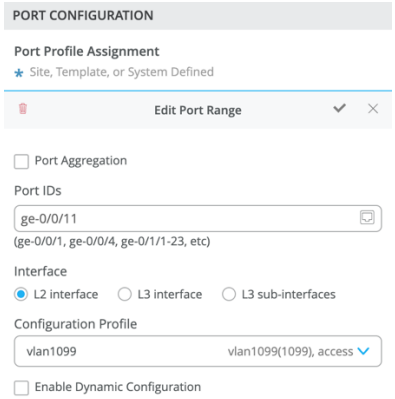
In this case, Desktop1/2 are associated with different ports on each Access Switch, which requires the configuration to be applied to Access1/2. Figure 2. Topology

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

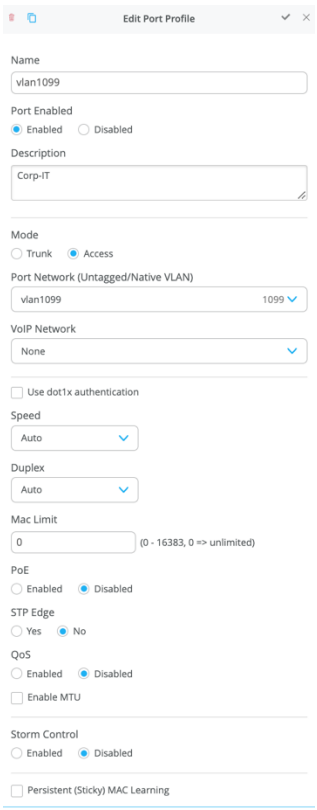
Organization/Switch template option is customized to associate each switch with its role: Core, Distribution, and Access. Further, all Access switches (defined by Model EX4400 as an example) associated the AP (Access Point) port profile with ge-0/0/16 without needing to configure each independent switch.



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. 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 below, representing attributes associated with VLANs such as dot1x authentication, QoS (Quality of Service), and Power over Ethernet. Vlan1088 and vlan1033 will need to be configured in a similar fashion.


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

Verification of the Campus Fabric IP Clos deployment. Figure 2. Topology

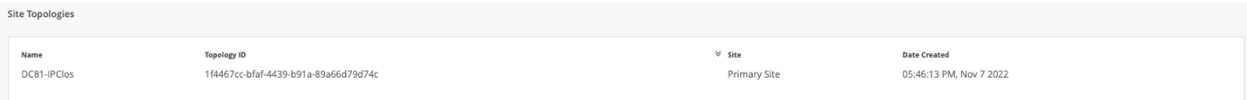
Currently, there are two desktops that can be used to validate the Campus Fabric. Let us take a quick look to see if Desktop1 can connect internally and externally. A third-party tool such as SecureCRT can be used to validate each desktop's configuration with Desktop1 shown below:

```
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, +2 errors, 100% packet loss, time 1016ms
```

Validation steps

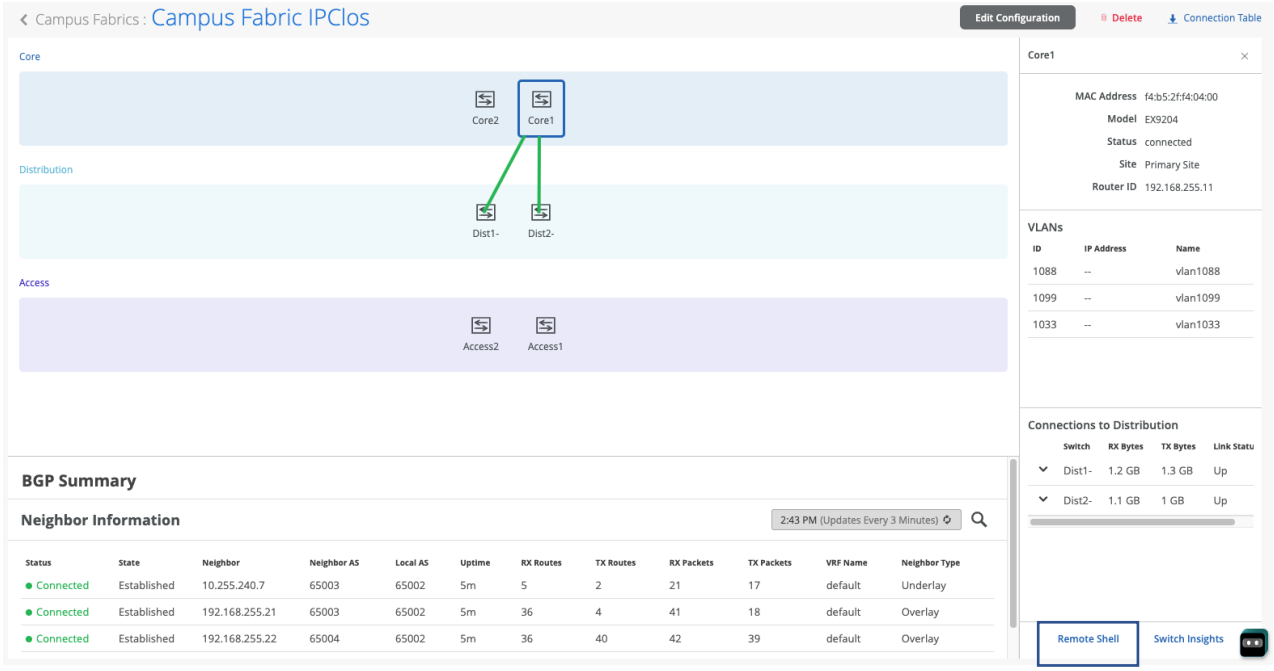
- confirmed local IP address, vlan and default gateway were configured on Desktop1
- can ping default gateway – that tells us we can reach access switch
- ping to WAN router failed (10.99.99.254) – we need to troubleshoot.

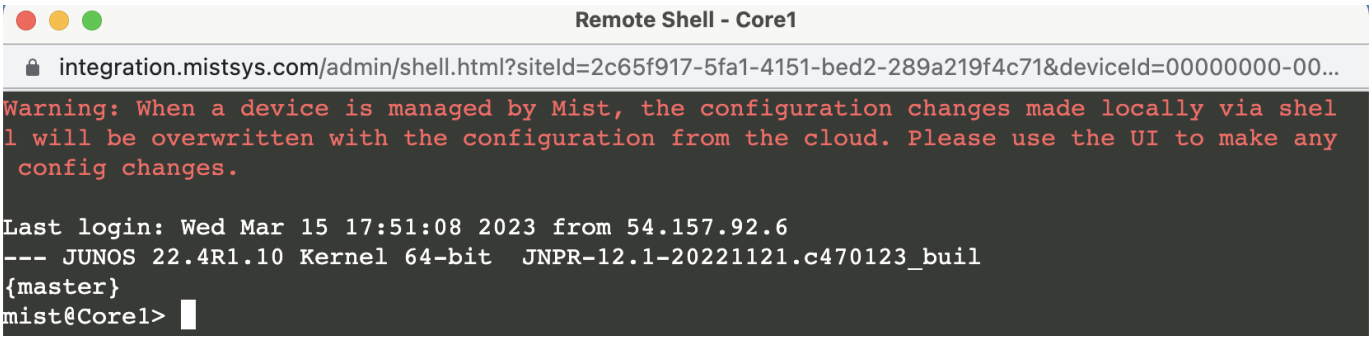
Start by validating Campus Fabric in the Mist UI, by selecting the Campus Fabric option under the Organization tab on the left-hand side of the UI.



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
- BGP peering as well as loopback VXLAN reachability

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 SecureCRT or Putty.

Verification of BGP peering

Dist1:

From SwitchàUtilities, Remote Shell can be accessed via the bottom right of the Campus Fabric, from the switch view or via SSH (Secure Shell).

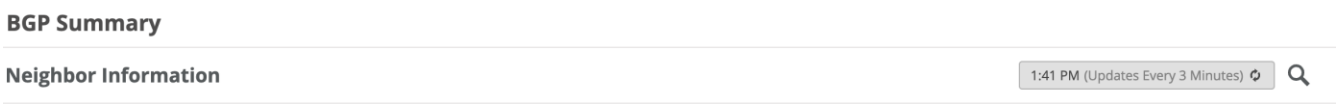
```
{master:0}
root@Dist1> show bgp summary

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
                      14          8          0          0          0          0
bgp.evpn.0
                     127         59          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/Received/Accepted/Damped...
10.255.240.2          65001       5709       5658       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.

The Campus Fabric build illustrates per device real-time BGP peering status shown below from Dist1:



Status	State	Neighbor	Neighbor AS	Local AS	Uptime	RX Routes	TX Routes	RX Packets	TX Packets	VRF Name	Neighbor Type
● Connected	Established	10.255.240.11	65005	65003	7m	3	4	21	20	default	Underlay
● Connected	Established	10.255.240.13	65006	65003	7m	3	5	21	23	default	Underlay
● Connected	Established	192.168.255.31	65006	65003	7m	25	22	35	34	default	Overlay
● Connected	Established	10.255.240.2	65001	65003	7m	4	5	25	23	default	Underlay
● Connected	Established	10.255.240.6	65002	65003	7m	2	5	20	22	default	Underlay
● Connected	Established	192.168.255.11	65002	65003	7m	4	37	22	44	default	Overlay
● Connected	Established	192.168.255.12	65001	65003	7m	36	37	48	46	default	Overlay
● Connected	Established	192.168.255.32	65005	65003	7m	19	27	32	40	default	Overlay

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

```
{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 a Campus Fabric IP Clos

Let us 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 and each other's.

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                    ulst   131073     8
                            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                    ulst   131073     8
                            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                    ulst   131073     8
                              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

NOTE: At this point BGP Underlay and Overlay is operational through the verification of eBGP between adjacent layers of the Campus Fabric and that routes are established to Access, Distribution, and Core switches.

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          00:cc:34:f3:cf:00  irb.0                          Nov 07 23:13:26
     1          00:cc:34:f4:72:00  192.168.255.31                 Nov 07 23:13:46
     1          f4:b5:2f:f3:fb:f0  192.168.255.12                 Nov 07 23:13:46
     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 us 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 and Core2. The reason VXLAN is supported on both Cores is due to the L2 multihomed connection to the wan router, in this case a Juniper SRX firewall. The L2 connection, called an ESI-LAG, was built using 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

```
root@Access1> show route advertising-protocol bgp 192.168.255.21 evpn-mac-address 52:54:00:74:a0:6f table bgp.evpn.0

Warning: License key missing; One or more members of the VC require 'bgp' license


bgp.evpn.0: 46 destinations, 46 routes (46 active, 0 holddown, 0 hidden)
  Prefix                Nexthop              MED     Lclpref    AS path
  2:192.168.255.31:1::11099::52:54:00:74:a0:6f/304 MAC/IP
*                       Self                                    I
  2:192.168.255.31:1::11099::52:54:00:74:a0:6f::10.99.99.99/304 MAC/IP
*                       Self                                    I

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

And is it being received on the core

```
root@Core1> show route receive-protocol bgp 192.168.255.21 evpn-mac-address 52:54:00:74:a0:6f table bgp.evpn.0

Warning: License key missing; requires 'bgp' license


bgp.evpn.0: 46 destinations, 53 routes (46 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
  2:192.168.255.31:1::11099::52:54:00:74:a0:6f/304 MAC/IP
*                         192.168.255.31                          65003 65006 I
  2:192.168.255.31:1::11099::52:54:00:74:a0:6f::10.99.99.99/304 MAC/IP
*                         192.168.255.31                          65003 65006 I

{master}
root@Core1>
```

Let us check to see if the core has Desktop1 mac address

```
root@Core1> show evpn database | match 52:54:00:74:a0:6f
     11099      52:54:00:74:a0:6f  192.168.255.31                Feb 09 18:59:46  10.99.99.99

{master}
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 52:54:00:74:a0:6f
Routing table: evpn_vs.evpn-vxlan [Index 8]
Bridging domain: vlan1099.evpn-vxlan [Index 53]
VPLS:
Enabled protocols: Bridging, ACKed by all peers, EVPN VXLAN,

Destination:  52:54:00:74:a0:6f/48
  Learn VLAN: 0                       Route type: user
  Route reference: 0                  Route interface-index: 346
  Multicast RPF nh index: 0
  P2mpidx: 0
  IFL generation: 277                 Epoch: 0
  Sequence Number: 0                  Learn Mask: 0x40000000000000000000000000000000000000000
  L2 Flags: control_dyn
  Flags: sent to PFE
  Nexthop:
  Next-hop type: composite            Index: 670     Reference: 14

{master}
root@Core1>
```

```
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      536     1
xe-1/1/0.0         intf     0                    ucst      656     2 xe-1/1/0.0
xe-1/1/1.0         intf     0                    ucst      657     2 xe-1/1/1.0

Routing table: evpn_vs.evpn-vxlan
VPLS:
Destination        Type RtRef Next hop           Type Index    NhRef Netif
default            perm     0                    dscd      597     1
vtep.32769         intf     0                    comp      668     9
vtep.32770         intf     0                    comp      669     5
vtep.32771         intf     0                    comp      670    14
```

Finally, the VTEP interface is up and passing traffic:

```
root@Core1> show interfaces vtep.32771
  Logical interface vtep.32771 (Index 346) (SNMP ifIndex 578)
    Flags: Up SNMP-Traps Encapsulation: ENET2
    VXLAN Endpoint Type: Remote, VXLAN Endpoint Address: 192.168.255.31, L2 Routing Instance: evpn_vs, L3 Routing Instance: default
    Input packets : 4138
    Output packets: 335
    Protocol eth-switch, MTU: Unlimited
      Flags: Trunk-Mode

{master}
root@Core1>
```

From an EVPN-VLAN perspective everything is looking correct. Maybe we are looking in the wrong place. Let us 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:

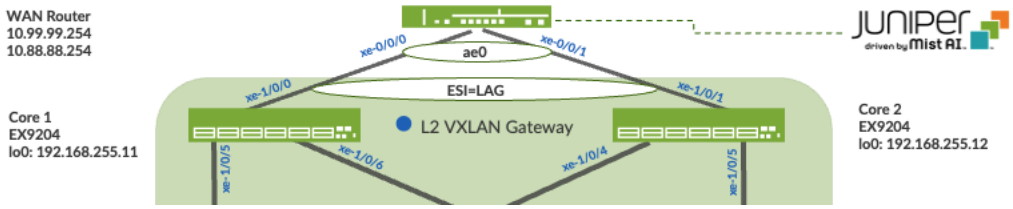


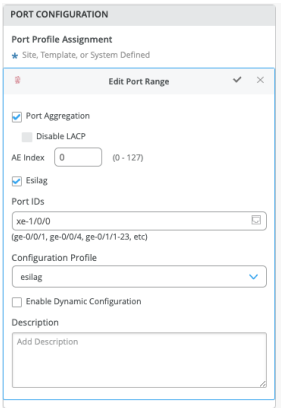
Figure 4. Layer 2 ESI-LAG supporting active-active load balancing

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. Let us verify the ESI (Ethernet Segment Identifier) 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 00:11:11:11:11:11:11:01:00
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:b5:2f:f3:fb:f0  192.168.255.12                 Nov 07 22:59:09
     1          f4:b5:2f:f4:0b:f0  irb.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  00:11:11:11:11:11:11:01:00  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  00:11:11:11:11:11:11:01:00  Nov 09 17:40:55  10.88.88.254
     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  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  00:11:11:11:11:11:11:01:00  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.

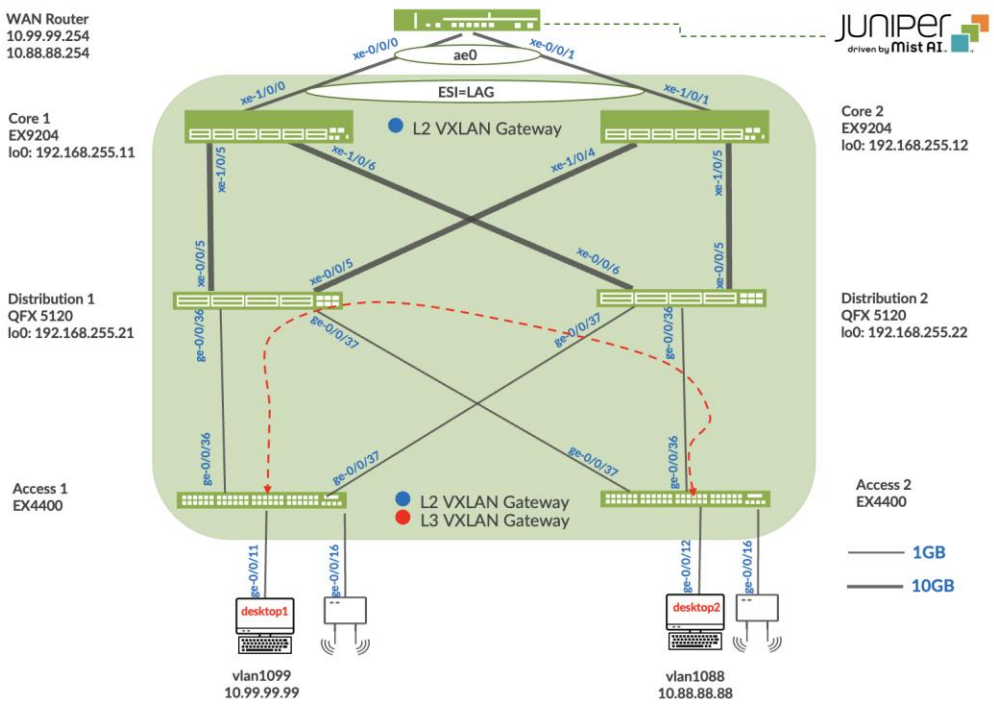
```
root@Core1> show lacp statistics interfaces
Aggregated interface: ae0
    LACP Statistics:       LACP Rx     LACP Tx   Unknown Rx   Illegal Rx
      xe-1/0/0                2165        2166            0            0

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 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

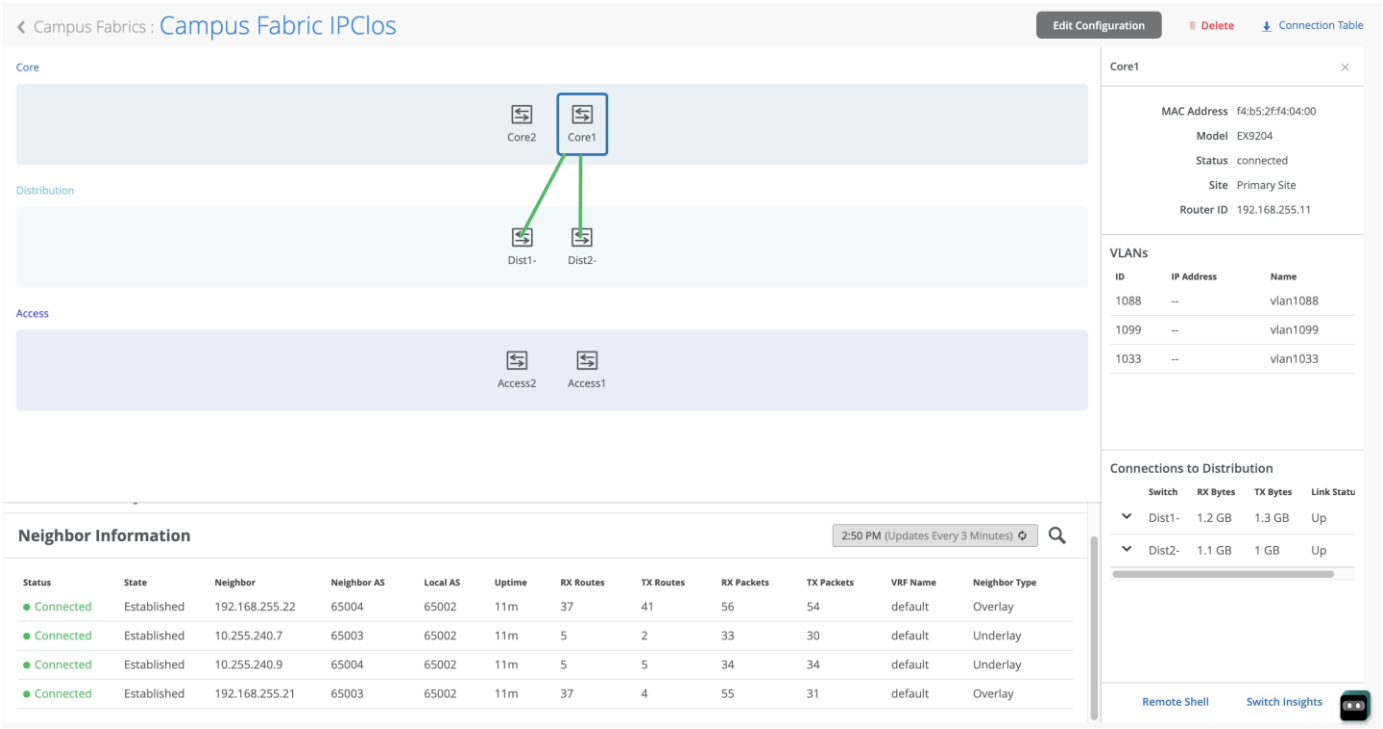
```
root@desktop1:~# ping 10.88.88.88 -c 2
PING 10.88.88.88 (10.88.88.88) 56(84) bytes of data.
64 bytes from 10.88.88.88: icmp_seq=1 ttl=62 time=4.68 ms
64 bytes from 10.88.88.88: icmp_seq=2 ttl=62 time=0.590 ms

--- 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:~#
```



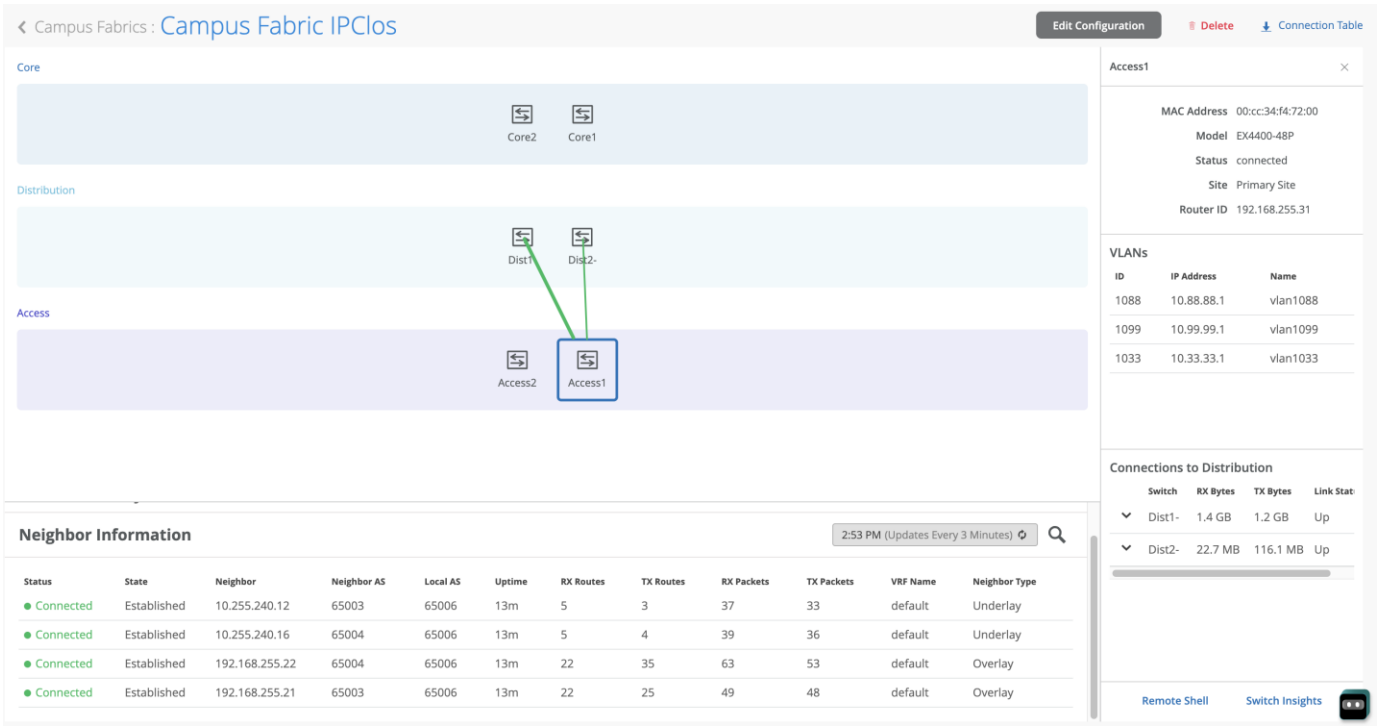
NOTE: Connectivity within and outside of the Campus fabric have been verified. Desktops communicate with each through the Fabric, each in an isolated VRF, then forwarded to the SRX firewall through the ESI-LAG on both Core devices when accessing services outside of the Campus Fabric. The Campus Fabric performs total isolation between VRF by default while utilizing the SRX firewall to accept or discard inter-VRF communications.

EVPN Insights

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



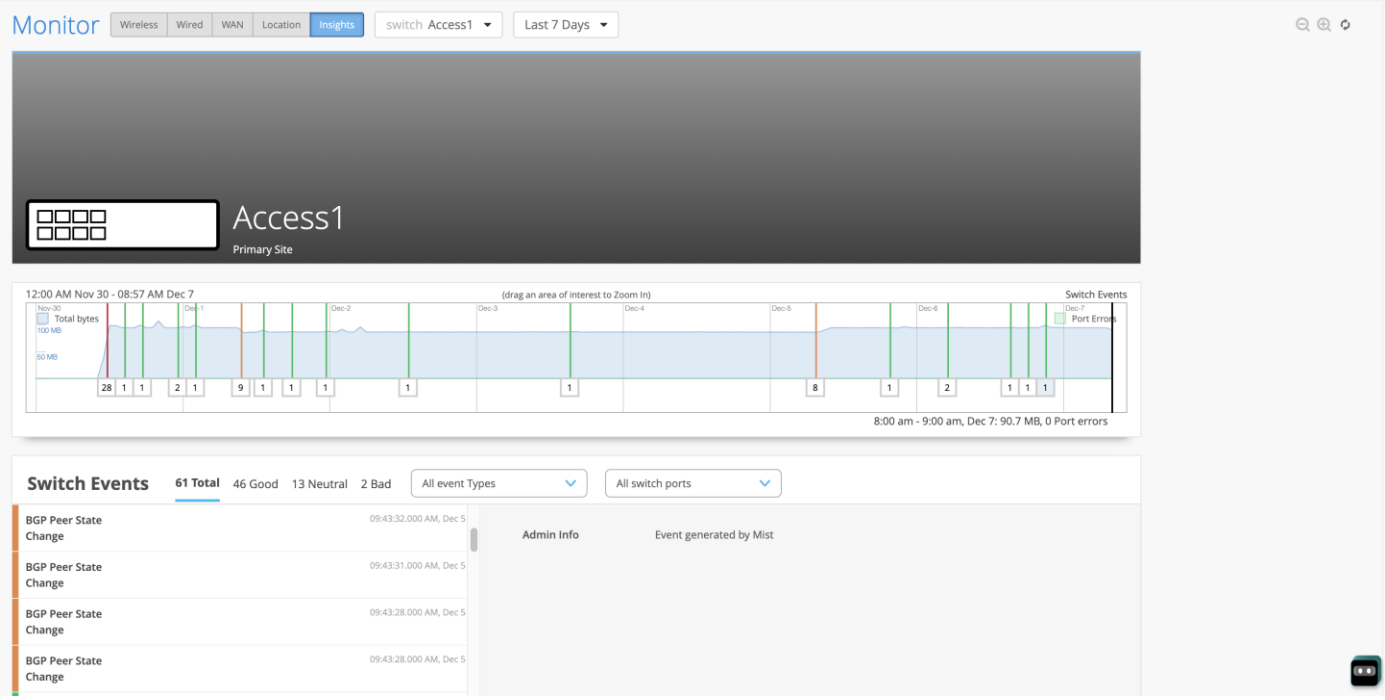
NOTE: For brevity's sake, the full BGP peering table is not shown



NOTE: For brevity's sake, the full BGP peering table is not shown

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 real-time 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 is not working correctly.

Appendix

Configuration of the Underlay IP Fabric

This section displays the configuration output from the 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 Prefix (/24)
- Subnet (10.255.240.0/20) – point to point interfaces between adjacent layers

Mist enables per-packet (Junos defines this as per-flow) load-balancing using ECMP and fast convergence of BGP in the event of a link or node failure using BFD

Core1 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 and router ID

```
set groups top interfaces lo0 unit 0 family inet address 192.168.255.11/32.
set groups top routing-options router-id 192.168.255.11
```

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 "$9$deboJGUHf5FwYfT36AtxN-
V4ak.P5Fnbs4ZjHmPSrlvxNws4oGDY2n/9A1IxN-
ws4ik.5z3q.z6CtIR24oJikFn/tpB6/u1RhKvgoaUk.mfTn6AzFyleK8LUjiHqf369pO1zFlK8X-
ds24aJDik.PfzkqBIEhKvjHkq5QCtu0IEAtOREcvMaZGD.P"
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 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 and router ID

```
set groups top interfaces lo0 unit 0 family inet address 192.168.255.12/32
set groups top routing-options router-id 192.168.255.12
```

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 "$9$71-
24aJD.mTdb.PQF/98XxNYgjHqmTz-
VYoGDkqEcSl8XdVY2aZbwz3n/0O8XxdVYUjHm5QiH5F69OBwY24UjTz39CuF3A0BIrls2gJjHk.PzF/5ThSyrvMJG
UDi.QFnCp05TSrvWx7VwYg4ZUjHq.5jiuO1IrlGDjimf69AtO1/9pB1RlegoaZHq"
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 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 and router ID

```
set groups top interfaces lo0 unit 0 family inet address 192.168.255.21/32
set groups top routing-options router-id 192.168.255.21
```

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 "$9$wLYZUji.Qz624QF/Cu0-
VbsJGmfTz69YgJDk.5TlKv8-V2gJZjH4o9AtuEh-
Vb2gJqmfzn/PfnCp0hcoJZUqm69A0ORCABEcyW8aZGimf5QF9Cun6evMWx7ikq.PQ/CtOIEn6vWxNbwgoJGUHqmfT
QnmPRhSyW8k.mPz3p0B1hSu0IcSr8LGDjHfT"
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 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 192.168.255.22/32
set groups top routing-options router-id 192.168.255.22
```

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
"$9$GpDmfTQntpBjHtu1RSyoJZU.P69ApBIDi.5FnCA7-
dwoJji.mTzHkIEhSMWoJZji.369pO1/9ORcyW8k.mf36BIEyrvRElM8XbwqmPQ69CtuIRSOBNdVb2gQF3n/t1RhrK
MOBdb24ZGik.Pfz369AtO6/vWLXbwFn6/p0cyleWLSyK8LxwsP5Tz9A"
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 core switches and the two access 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

```
set groups top interfaces lo0 unit 0 family inet address 192.168.255.31/32
set groups top routing-options router-id 192.168.255.31
```

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
"$9$gVojHq.5n6AaZn/tu1IwY24DiTz36ApoJDkP5F3M8L7wYaJDjqmZGp0O1yrwY2aJDfTz6CtQzCuBIrlGDjHfT
Ap0IRSu0EylKx7Uji.TzFn/pu1CAWLXxdV.Pf5QntuORcyCALxdb2gJGDiHmfTz3nCTQSreKx7P5TQ69BIEhre1Ic
lev7Nikqmz3"
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 core switches and the two access 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

```
set groups top interfaces lo0 unit 0 family inet address 192.168.255.32/32
set groups top routing-options router-id 192.168.255.32
```

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
"$9$4qaik.mT6/tJG69p0IRs2gojHQFn/tuaZjqfT3nWLXNs2JZji.PGUuOBIrls2gJZj5QF/ApzFA01RleUjik5Q
tuOREy0Ohrev7NDiHmQF369u0IAt8Xx7Vbmf5Tz6p0BESrAtX7Vwg4ZUjHkP5QFn6AQzylKv7NfTQz/C1RhclKIRS
eKMN-Hq.PFn"
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 configuration output from the Mist Cloud 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 provisions L3 IRB interfaces on the Access layer (if the Routed at Distribution option was chosen during the initial phases of the Campus Fabric build, the L3 IRB interfaces would be on the Distribution switches)

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 tunnelling and VLAN to VXLAN mapping enabled based on the selection of the Core as a Border option.

Core1 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 192.168.255.11
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 "$9$deboJGUHf5FwYfT36AtxN-
V4ak.P5Fnbs4ZjHmPSrlvxNws4oGDY2n/9A1IxN-
ws4ik.5z3q.z6CtIR24oJikFn/tpB6/u1RhKvgoaUk.mfTn6AzFyleK8LUjiHqf369pO1zFlK8X-
ds24aJDik.PfzkqBIEhKvjHkq5QCtu0IEAtOREcvMaZGD.P"
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 192.168.255.21 peer-as 65003
set protocols bgp group evpn_overlay neighbor 192.168.255.22 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 192.168.255.11:1
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 used for traffic isolation

```
set groups top routing-instances guest-wifi instance-type vrf
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 interfaces irb.1033
set groups top routing-instances guest-wifi route-distinguisher 192.168.255.11:103
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 route-distinguisher 192.168.255.11:102
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 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 interfaces irb.1088
set groups top routing-instances corp-it route-distinguisher 192.168.255.11:101
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 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 192.168.255.12
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 "$9$deboJGUHf5FwYfT36AtxN-
V4ak.P5Fnbs4ZjHmPSrlvxNws4oGDY2n/9A1IxN-
ws4ik.5z3q.z6CtIR24oJikFn/tpB6/u1RhKvgoaUk.mfTn6AzFyleK8LUjiHqf369pO1zFlK8X-
ds24aJDik.PfzkqBIEhKvjHkq5QCtu0IEAtOREcvMaZGD.P"
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 192.168.255.21 peer-as 65003
set protocols bgp group evpn_overlay neighbor 192.168.255.22 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 192.168.255.12:1
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 used for traffic isolation

```
set groups top routing-instances guest-wifi instance-type vrf
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 interfaces irb.1033
set groups top routing-instances guest-wifi route-distinguisher 192.168.255.12:103
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 route-distinguisher 192.168.255.12:102
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 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 interfaces irb.1088
set groups top routing-instances corp-it route-distinguisher 192.168.255.12:101
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 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 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-nexthop-change
set protocols bgp group evpn_overlay local-address 192.168.255.21
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 "$9$wLYZUji.Qz624QF/Cu0-
VbsJGmfTz69YgJDk.5TlKv8-V2gJZjH4o9AtuEh-
Vb2gJqmfzn/PfnCp0hcoJZUqm69A0ORCABEcyW8aZGimf5QF9Cun6evMWx7ikq.PQ/CtOIEn6vWxNbwgoJGUHqmfT
QnmPRhSyW8k.mPz3p0B1hSu0IcSr8LGDjHfT"
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 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
```

Dist2 Configuration

2. 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-nexthop-change
set protocols bgp group evpn_overlay local-address 192.168.255.22
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 "$9$wLYZUji.Qz624QF/Cu0-
```

```
VbsJGmfTz69YgJDk.5TlKv8-V2gJZjH4o9AtuEh-
Vb2gJqmfzn/PfnCp0hcoJZUqm69A0ORCABEcyW8aZGimf5QF9Cun6evMWx7ikq.PQ/CtOIEn6vWxNbwgoJGUHqmfT
QnmPRhSyW8k.mPz3p0B1hSu0IcSr8LGDjHfT"
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 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
```

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-nexthop-change
set protocols bgp group evpn_overlay local-address 192.168.255.31
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
"$9$gVojHq.5n6AaZn/tu1IwY24DiTz36ApoJDkP5F3M8L7wYaJDjqmZGp0O1yrwY2aJDfTz6CtQzCuBIrlGDjHfT
Ap0IRSu0EylKx7Uji.TzFn/pu1CAWLXxdV.Pf5QntuORcyCALxdb2gJGDiHmfTz3nCTQSreKx7P5TQ69BIEhre1Ic
lev7Nikqmz3"
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 192.168.255.21 peer-as 65003
set protocols bgp group evpn_overlay neighbor 192.168.255.22 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 192.168.255.31:1
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 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 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 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 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 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 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
```

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 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 192.168.255.32
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
"$9$gVojHq.5n6AaZn/tu1IwY24DiTz36ApoJDkP5F3M8L7wYaJDjqmZGp0O1yrwY2aJDfTz6CtQzCuBIrlGDjHfT
Ap0IRSu0EylKx7Uji.TzFn/pu1CAWLXxdV.Pf5QntuORcyCALxdb2gJGDiHmfTz3nCTQSreKx7P5TQ69BIEhre1Ic
lev7Nikqmz3"
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 192.168.255.21 peer-as 65003
set protocols bgp group evpn_overlay neighbor 192.168.255.22 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 192.168.255.32:1
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 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 route-distinguisher 192.168.255.32:103
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 route-distinguisher 192.168.255.32:102
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 route-distinguisher 192.168.255.32:101
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 Layer 2 ESI-LAG between the core switches and SRX firewall

This section displays the configuration output from the Mist Cloud for the enablement of the Layer 2 ESI LAG (Link Aggregation Groups) between the core switches and SRX firewall. This Mist profile enables all VLANs on the ethernet bundle with requisite ESI and LACP configuration

options. From the perspective of the SRX firewall, the ethernet bundle that is configured on the SRX views the ESI-LAG as a single MAC address with the same LACP system-id. This enables load hashing between the core and SRX without requiring L2 loop free detection protocols such as RSTP (Rapid Spanning Tree Protocol) (Rapid Spanning Tree Protocol).

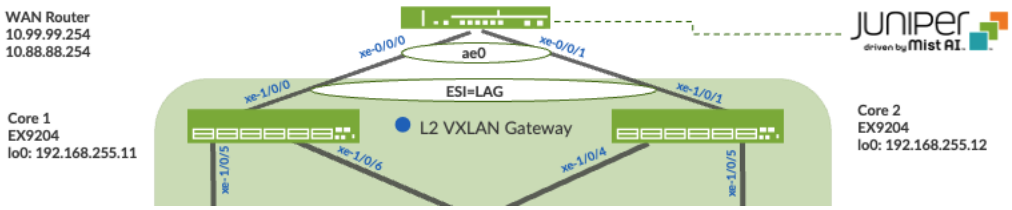


Figure 5. Layer 2 ESI-LAG supporting active-active load balancing

Core 1 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 ae1
set interfaces xe-1/0/0 unit 0 family ethernet-switching storm-control default

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 interfaces ae1 apply-groups esilag
set interfaces ae1 esi 00:11:00:00:00:01:00:01:02:01
set interfaces ae1 esi all-active
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp periodic fast
set interfaces ae1 aggregated-ether-options lacp system-id 00:00:00:31:57:01
set interfaces ae1 aggregated-ether-options lacp admin-key 1
```

Core 2 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 ae1
set interfaces xe-1/0/1 unit 0 family ethernet-switching storm-control default

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 interfaces ae1 apply-groups esilag
set interfaces ae1 esi 00:11:00:00:00:01:00:01:02:01
```

```
set interfaces ae1 esi all-active
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp periodic fast
set interfaces ae1 aggregated-ether-options lacp system-id 00:00:00:31:57:01
set interfaces ae1 aggregated-ether-options lacp admin-key 1
```

SRX Firewall Configuration

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

```
set interfaces ae0 apply-groups lan
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 mtu 9014
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 unit 1033 description vlan1033
set interfaces ae0 unit 1033 vlan-id 1033
set interfaces ae0 unit 1033 family inet address 10.33.33.254/24
set interfaces ae0 unit 1088 description vlan1088
set interfaces ae0 unit 1088 vlan-id 1088
set interfaces ae0 unit 1088 family inet address 10.88.88.254/24
set interfaces ae0 unit 1099 description vlan1099
```