

3-Stage EVPN/VXLAN Fabric with Juniper Apstra—Juniper Validated Design (JVD)

Published 2025-01-15

Table of Contents

About this Document | 1

Solution Benefits | 1

Use Case and Reference Architecture | 3

Solution Architecture | 4

Configuration Walkthrough | 10

Validation Framework | 78

Test Objectives | 81

Results Summary and Analysis | 83

Recommendations | 85

Tested Optics | 85

Revision History | 88

3-Stage EVPN/VXLAN Fabric with Juniper Apstra— Juniper Validated Design (JVD)

your requirements. Juniper Networks Validated Designs provide you with a comprehensive, end-to-end blueprint for deploying Juniper solutions in your network. These designs are created by Juniper's expert engineers and tested to ensure they meet your requirements.

Using a validated design, you can reduce the risk of costly mistakes, save time and money, and ensure that your network is optimized for maximum performance.

About this Document

This document details a Juniper Validated Design (JVD) to provision a 3-stage EVPN/VXLAN fabric with Juniper Apstra using Apstra's Data Center Architecture design feature, consisting of two spines, three server leaf switches, and two border leaf switches. The validation was done using several combinations of device models, which are listed in the document. This document is intended for an audience familiar with Juniper technologies such as the Junos OS, QFX switches, and Juniper Apstra.

NOTE: Nomenclature Note: Edge-routed bridging (ERB) is the Juniper terminology for a network architecture that is referred to elsewhere in the industry as distributed VXLAN routing with EVPN or the distributed gateways model.

Solution Benefits

IN THIS SECTION

- Juniper Validated Design Benefits | 2
- Juniper Apstra | 2

This document offers comprehensive guidance on deploying a modern 3-stage fabric with EVPN-VXLAN. The 3-Stage Fabric with Juniper Apstra is designed to meet the needs of most of Juniper's customers, has been extensively tested by Juniper, and is deployed by customers across the globe. Advanced JVD testing by Juniper combined with widespread adoption simplifies troubleshooting and shortens the support cycle, leading to a more stable data center fabric, reducing operational costs.

The 3-Stage Fabric with Juniper Apstra is Juniper's "standard candle" data center network architecture. Like all Juniper data center JVDs it is based on best practices as determined by Juniper's subject matter experts, and Juniper support teams have extensive training and resources necessary to support networks based on JVDs.

Juniper Validated Design Benefits

JVDs are a prescriptive blueprint for building a data center fabric with well-documented capabilities and appropriate product selection. JVDs must pass rigorous testing with real-world workloads to achieve validation, verifying that all products in the Building Blocks JVD work together as expected and mitigating the risk faced while deploying a network. The core benefits of JVDs are:

- Repeatability—Unlock value with repeatable network designs. Because JVDs are prescriptive designs
 used by multiple customers, all JVD customers benefit from lessons learned through lab testing and
 real-world deployments.
- Reliability—Layered testing with real traffic. JVDs are quantified and integrated best practice designs based on carefully chosen hardware platforms and software versions and tested with real-world traffic.
- Accelerated Deployment—Ease installation with step-by-step guidance. Simplify deployment with guidance, automation, and prebuilt integrations.
- Accelerated Decision-Making—Leave behind costly bespoke networks. Bridge business and technology in designs that meet the needs of most customers and consider how features behave and operate in real-world applications and conditions.
- Best Practice Networks—Better outcomes for a better experience. Juniper Validated Designs have known characteristics and performance profiles to help you make informed decisions about your network.

Juniper Apstra

Apstra is a multi-vendor, intent-based network fabric management solution that provides closed-loop automation and assurance. Apstra translates business intent and technical objectives to essential policy

and device-specific configuration. Apstra continuously self-validates and resolves issues to assure compliance. The core benefits of Apstra are:

- Intent-based networking—Automates configuration generation and continuously validates operating state versus intent.
- Network Automation—Apstra is a multi-vendor network automation platform that is continuously updated to work with the latest hardware and exhaustively tested using modern DevOps practices.
- Recoverability—Built-in rollback capability restores known-working configuration in a fraction of the time.
- Day 2+ Management—Apstra's rich analytics capabilities, including Flow Data, reduce Mean Time to Resolution (MTTR).
- Simplicity—Apstra simplifies network management. For example, by reducing the complexity of data center interconnection (DCI), making it easy to unify multiple data centers while isolating failure domains for high availability and resilience.

Use Case and Reference Architecture

Traditional data center designs required chassis-based switches, which were complex to manage. As business demands change, data centers are changing. Today's data centers need to support virtualization, span multiple geographies, incorporate hybrid cloud elements, and provide infrastructure for AI workloads. With each new class of workload, managing and supporting these data centers is getting ever more complex. The Edge-Routed Bridging (ERB) network architecture at the heart of the 3-Stage Fabric with Juniper Apstra simplifies data center design by distributing the traditional network chassis into a switching fabric that is far more resilient and flexible.

ERB is a distributed network architecture that can meet nearly any network requirement. As a result, ERB underlies all Juniper data center-validated designs. With Juniper data center switches running Junos OS and Juniper Apstra as an orchestration platform to manage these switches, customers can now provision, manage, and monitor data centers using Juniper Apstra software, as shown in Figure 1 on page 4 and DCI capable Border. The ERB design is flexible, and with Juniper Apstra adding or removing leaf switches is easier.

Juniper Apstra is a multi-vendor Intent-Based Network System (IBNS). Apstra orchestrates data center deployments and manages small to large-scale data centers through Day-0 to Day-2 operations. It is an ideal tool for building data centers for AI clusters, providing invaluable Day-2 insights through monitoring and telemetry services. It supports provisioning 3-stage or 5-stage data center designs, including collapsed fabric for even smaller data center designs.

Figure 1: 3-Stage Architecture: Lean Spines, Server leaf Switches and DCI Capable Border Leaf Switches



Solution Architecture

IN THIS SECTION

- VRF Characteristics | 6
- Juniper Hardware and Software Components | 7
- Validated Functionality | 9
- Additional Functionality | 10

The 3-Stage Fabric with Juniper Apstra is an EVPN/VXLAN-based validated design based upon the ERB network architecture. Using an ERB network architecture provides the design increased resilience by assigning specific functions to each device role and ensuring that each device role can be scaled independently of the others. Each network switch participating in the design must occupy one of three roles:

• Server Leaf Switches

The leaf switch focuses on learning and advertising the local MAC Addresses to other remote switches through the BGP EVPN control plane. This means leaf switches can discover all the "remote" hosts without flooding the overlay with ARP/ND requests.

• Border Leaf Switches

Although a border leaf can function as a server leaf switch, it can also act as a gateway to external networks and hence require DCI features. DCI features include connecting to network overlays such as VMware NSX-T, MACSEC, deep buffers, and so on.

• Spine Switches

The spine switch only performs IP forwarding and relaying of routes to all server and border leaf switches. As a result, spine switches in ERB network architectures are referred to as lean spines.

The use of an ERB network architecture and the associated switch roles not only simplifies the data center design but also provides flexibility at the leaf layer so that new leaf switches can be introduced as traffic throughput increases. Another aspect of this design is the use of non-modular switches, such as the 1U QFX5130-32CD, which can perform high throughput functions at the leaf layer.

To summarize, the ERB network architecture, which underlies the 3-stage fabric with Juniper Apstra, can be thought of as a distributed chassis. In an ERB network, leaf switches are roughly analogous to a "line card" in a traditional modular chassis, while the lean spine means the network fabric is more flexible and resilient than a single modular chassis switch. This creates a network more capable and flexible than a traditional modular chassis-based switch, without requiring the purchase or maintenance of a modular chassis-based switch for most enterprise data center scenarios.

For those data centers looking for scale that can only be achieved with chassis-based switches, the Juniper Validation process does take this into account, validating modular chassis switch combinations in ERB network roles. The result is validated network fabrics that can scale from the needs of individual racks up to serving entire data centers and beyond.

Figure 1 on page 4 depicts the hardware in various roles such as spine, leaf, and border leaf. This JVD will walk through the high-level steps required to configure a 3-stage Data center, with QFX5220-32CD switches in the spine role, QFX5130-32CD switches in the border leaf role, and QFX5120-48Y switches in the server leaf role. These switches in these roles are considered the baseline design of this JVD, though other switches are qualified for these roles, as documented below.

Below is the reference architecture of 3-Stage Fabric with Juniper Apstra.



Figure 2: 3-Stage Reference Design with Baseline Devices

VRF Characteristics

RED VRF

- VLANs 400-649 with IRB v4/v6
 - on DC1-SNGL-LEAF1 single access port
 - on DC1-ESI-LEAF1 single access port, AE1 and AE2
 - on DC1-ESI1-LEAF2 single access port, AE1 and AE2
 - on DC1-BRDR-LEAF1 to distribute routes to external-router
 - on DC1-BRDR-LEAF2 to distribute routes to external-router
- VLANs 400-649 on each test port with 10 unique MAC/IP per VLAN
- DHCP client on TP3
- External DHCP server on TP17

Blue VRF

- VLANs 3500-3749 with IRB v4/v6
 - on DC1-SNGL-LEAF1 single access port
 - on DC1-ESI-LEAF1 single access port, AE1 and AE2

- on DC1-ESI1-LEAF2 single access port, AE1 and AE2
- on DC1-BRDR-LEAF1 to distribute routes to external-router
- on DC1-BRDR-LEAF2 to distribute routes to external-router
- VLANs 3500-3749 on each test port with 10 unique MAC/IP per VLAN
- DHCP client on TP3, TP4, TP5
- External DHCP server on TP2

Juniper Hardware and Software Components

For this solution, the Juniper products and software versions are as below.

The design documented in this JVD is considered the baseline representation for the validated solution. As part of a complete solutions suite, we routinely swap hardware devices with other models during iterative use case testing. Each switch platform validated in this document goes through the same rigorous role-based testing using specified versions of Junos OS and Apstra management software.

Juniper Hardware Components

The following switches are tested and validated to work with the 3-Stage Fabric with Juniper Apstra JVD in the following roles:

Supported Devices and Positioning						
Solution	Server Leaf Switches	Border Leaf Switches	Spine			
3-stage EVPN/VXLAN (FRB)	QFX5120-48Y-8C*	QFX5130-32CD*	QFX5220-32CD*			
()	QFX5110-48S	QFX5700	QFX5120-32C			
	EX4400-24MP#	ACX7100-48L	QFX5210-64CD			
		ACX7100-32C	QFX5200-32C			
		PTX10001-36MR				

Table 1: Supported Devices and Positioning

Table 1: Supported Devices and Positioning (Continued)

Supported Devices and Posi	tioning		
Solution	Server Leaf Switches	Border Leaf Switches	Spine
		QFX10002-36Q	

* marked are baseline devices

NOTE: There is a scale limitation on EX4400 switches that affects the whole fabric. Refer to the Multi-dimensional Scale Numbers Tested Table 6 on page 84 for scale numbers with EX4400. The version used for validation for EX4400 was 22.4R3.25 as this version supports MAC-VRF feature. Please contact Juniper account representative for more information about EX4400 setup and scale.

For more information on validated devices refer to Devices Under Test(Validated Devices) Table 5 on page 81.

For the purposes of this JVD document, the following switches are used for the configuration walkthrough:

Table 2: Hardware	for 3-Stage	Data Center	JVD Reference	Design
-------------------	-------------	-------------	---------------	--------

Juniper Hardware for 3-Stag	ge Design		
Juniper Products	Role	Hostname	Software or Image Version
QFX5220-32CD	Spine	dc1-spine1 & dc1-spine2	Junos OS Evolved Release 23.4R2-S3
QFX5120-48Y	Server Leaf	dc1-single-leaf1 dc1-esi-001-leaf1 dc1-esi-001-leaf2	Junos OS Release 23.4R2-S3
QFX5130-32CD	Border Leaf	dc1-border-leaf1 dc1-border-leaf2	Junos OS Evolved Release 23.4R2-S3

NOTE: All devices listed in Supported Devices and Positioning Table 1 on page 7 are validated against Junos OS Release 23.4R2-S3 release. The validated Junos OS release for PTX10001-36MR is Junos OS Evolved Release 23.4R2-S3, for ACX7100-32C and ACX7100-48L is Junos OS Evolved Release 23.4R2-S3.

Table 3: Juniper Software and Version

Juniper Software	
Juniper Products	Software or Image Version
Juniper Apstra	4.2.1-207

Validated Functionality

The 3-Stage Fabric with Juniper Apstra was validated using the following parameters in its configuration:

- This JVD consists of a 3-stage CLOS with an ERB network architecture using EVPN-VXLAN.
- Servers will be connected and tested both in single-homed and multi-homed configurations.
- In the case of multihomed ESI servers, LACP is enabled between the servers and the leaf switches.
- Configure ESI on aggregated ethernet interfaces for multi-homed devices.
- ECMP is configured across the fabric to minimize traffic loss.
- Both the overlay and underlay of the fabric are built using eBGP.
- Learn and advertise EVPN Type 2 and Type 5 routes.
- BFD is enabled for both underlay eBGP and overlay eBGP.
- Asymmetric IRB is enabled with anycast IP address on L3-enabled leaf switches for inter-VLAN routing. For more information on the IRB model for inter-subnet forwarding in EVPN, refer to the EVPN VXLAN Guide.
- Both IPv4 and IPv6 are enabled; however, IPv6 is only used for loopback.

 Inter-VRF connectivity is configured using external router to allow route leaking between VRFs, however, to achieve this configuration Apstra Connectivity templates were used to connect to the external router.

Additional Functionality

The below features are not considered part of, nor are described in, this JVD; however, they are validated:

- DCI between data centers.
- Interoperability with NSX-T Edge Gateway.
- Host connectivity between fabric-connected hosts created by Apstra towards NSX-managed hosts.

Configuration Walkthrough

IN THIS SECTION

- Apstra: Configure Apstra Server and Apstra ZTP Server | 12
- Apstra: Management of Junos OS Device | 12
- Apstra Web UI: Create Agent Profile | 13
- Apstra Web UI: Enter IP Address or IP Address Range for Bulk Discovery of Devices | 13
- Apstra Web UI: Add Pristine Configuration and Upgrade Junos OS | 14
- Apstra Fabric Provisioning | 17
- Apstra Web UI: Identify and Create Logical Devices, Interface Maps with Device Profiles | 18
- Device Profiles | 20
- Spine Logical Device and Corresponding Interface Maps | 22
- Server Leaf switches Logical Device and Interface Maps | 23
- Border Leaf Switches Logical Device and Interface Maps | 26
- Generic Servers Logical Device | 28
- External Routers | 28
- Apstra Web UI: Racks, Templates, and Blueprints—Create Racks | 28

- Create Templates | 32
- Blueprint | 33
- Apstra Web UI: Provisioning and Defining the Network | 35
- Assign Resources | 35
- Assign Interface Maps to Switches | 37
- Assign the System IDs and the Correct Management IPs | 38
- Review Cabling | 39
- Configlet and Property Sets | 41
- Fabric Setting | 42
- Commit the Configuration | 46
- Apstra Fabric Configuration Verification | 47
- Configure Overlay Network | 51
- Create Virtual Networks in Red and Blue Routing Zones | 53
- Verify Overlay Connectivity for Blue and Red Network | 56
- Verify that ERB is Configured on Leaf Switches | 58
- Verify the Leaf Switch Routing Table | 59
- Configure External Router and Inter-VRF Routing | 62
- Apstra UI: Blueprint Dashboard, Analytics, probes, Anomalies | 73
- Analytics Dashboard, Anomalies, Probes and Reports | 73

This walkthrough summarizes the steps required to configure the 3-Stage Fabric with Juniper Apstra JVD. For more detailed step-by-step configuration information, refer to the Juniper Apstra User Guide. Additional guidance in this walkthrough is provided in the form of Notes.

This walkthrough details the configuration of the baseline design, as used during validation in the Juniper data center validation test lab. The baseline design consists of QFX5220-32CD switches in the spine role, QFX5130-32CD switches in the border leaf role, and QFX5120-48Y switches in the server leaf role. The goal of JVD is to provide options so that any of these switch platforms can be replaced with a validated switch platform for that role, as described in Supported Devices and Positioning Table 1 on page 7. In order to keep this walkthrough a manageable length, only the baseline design platforms will be used for the purposes of this document.

Apstra: Configure Apstra Server and Apstra ZTP Server

This document does not cover the installation of Apstra. For more information about installation, refer to the Juniper Apstra User Guide.

The first step is to configuration of the Apstra server. A configuration wizard launches upon connecting to the Apstra server VM for the first time. At this point, passwords for the Apstra server, Apstra UI, and network configuration can be configured.

Apstra: Management of Junos OS Device

There are two methods of adding Juniper devices into Apstra: manually or in bulk using ZTP:

To add devices manually (recommended):

In the Apstra UI navigate to Devices > Agents > Create Offbox Agents.

This requires minimum configuration of root password and management IP to be configured on the devices.

To add devices through ZTP:

From the Apstra ZTP server, to add devices, refer to the Juniper Apstra User Guide for more information on the ZTP of Juniper devices.

For this setup, a root password and management IPs were already configured on all switches prior to adding the devices to Apstra. To add switches to Apstra, first log into the Apstra Web UI, choose a method of device addition as per above, and provide the appropriate username and password preconfigured for those devices.

NOTE: Apstra pulls the configuration from Juniper devices called a pristine configuration. The Junos configuration 'groups' stanza is ignored when importing the pristine configuration, and Apstra will not validate any group configuration listed in the inheritance model, refer to the Use Configuration Groups to Quickly Configure Devices. However, it's best practice to avoid setting loopbacks, interfaces (except management interface), routing-instances (except management-instance). Apstra will set the protocols LLDP and RSTP when device is successfully Acknowledged.

Apstra Web UI: Create Agent Profile

For the purposes of this JVD lab, the root user and password are the same across all devices. Hence, an agent profile is created as below. Note that this also obscures the password, which keeps it secure.

- 1. Navigate to **Devices** > Agent Profiles.
- 2. Click Create Agent Profile.



er eute Agen			
Profile Parameters	5		
	Name *		
	root_user		
	Platform		
	Junos		×
	Username Set username?		
oot User	root		
	Password Set password?		
oot Password	•••••		•
Open Options			
	Кеу	Value	
		No options	
	Add an option		
ackages 🗿			
	▶ Query: All		
	0 selected Name \$		Version \$
			Create Another? Creat

Apstra Web UI: Enter IP Address or IP Address Range for Bulk Discovery of Devices

An IP address range can be provided to bulk-add devices into Apstra.

1. Navigate to **Devices** > Agents.

2. Click Create Offbox Agents.

Figure 4: Create Offbox Agent

Agent Parameters		
Device Address *		
10.6.1.153		
Operation Mode		
FULL CONTROL	TELEMETRY ONLY	
Platform *		
Junos		×
Username		
Set username?		
Password		
Set password?		
Agent Profile		
root_user		•
root_user		
Fackages		
		From Agent Profile
Query: All	< >	Agent Profile is not selected

Apstra Web UI: Add Pristine Configuration and Upgrade Junos OS

From **Devices > Managed Devices**, add the pristine configuration by collecting from the device or pushing from Apstra. The configuration applied as part of the pristine configuration should be the base configuration or minimal configuration required to reach the devices with the addition of any users, static routes to the management switch, and so on. This creates a backup of the base configuration in Apstra and allows devices to be reverted to the pristine configuration in case of any issues.

Figure 5: Add Pristine Configuration

☆ 倄 → Devices → M	lanaged Devices > 10.6.1.52	Pristine Config			Collect. Pristine/ba	se∙	
E Device	Agent	Pristine Config	Telemetry			ľ	Î
				Revert to F	Pristine U	pdate From D	evice
This is the pre	e-Apstra config on the device						
committed_configuration	Z	1 ve: 2 sy: 3	rsion 22.3R1.11; stem { root-authentication { encryphed_nassword "\$6	¢åCY±NRYDSKna76n&U7hHmfM]RB#	1 ብህ 2 ጀω <u>ሰ</u> 4 ጥ⊻		
Edit-Pristine- configuratio			enursmedeb4X98010 an	ANY ANY DALADON ADDINALORA	ALL AND TROUBLE		

NOTE: If pristine configuration is updated using Apstra as shown in above Figure 5 on page 15 then ensure to run **Revert to Pristine**.Important: A maintenance window is required to perform any device upgrade as upgrades can be disruptive.Best practice recommendations for Upgrade:Upgrade devices using the Junos OS CLI as outlined in the Junos OS Software Installation and Upgrade Guide, along with the Junos OS version release notes, as Apstra currently only performs basic upgrade checks. However, this JVD summarizes Upgrade steps if Apstra is intended to be used for upgrades.In case if a device is added to Blueprint, then set the device to "undeploy" and unassign its serial number from blueprint and commit the changes, which reverts it back to pristine Configuration. Then proceed to upgrade. Once upgrade is complete, add the device back to Blueprint.

Apstra allows the upgrade of devices. However, Apstra performs basic checks and issues the upgrade command. To upgrade the device from Apstra, refer to the following figure.

vice	0	£ .	4 1		ŵ	Agent	~	e e,	¥	Đ	Ξ.	
	a shad i	have in the		coloctod	only	uncol	acted a	Upgra	ue Os	mage		
iter sei	lected	by 🔘 a	all O	selected	only	 unsel 	ected o	opgra	de OS	mage	Device In	nformation
selected	lected Manaj	by 🔵 a	all O	Device Key	only y \$	O unsel	ected o	only	He	ostname	Device Ir	offormation OS \$

To register a Junos OS image on Apstra, either provide a link to the repository where all OS images are stored or upload the OS image as shown below. In the Apstra UI, navigate to **Devices > OS Images** and click **Register OS Images**.

Figure 7: Upload OS Image

ය 🔏 ▶ Devices ▶ OS Images	☆ 😤 → Devices → OS Images								
The partition aosservervg-root: free 19.30GB / tota	al 23.06GB				1. Register	OS Image			
••• Q					1-20 of	20 < >			
Name 🗘	Platform \$	Type 🖨	Size \$	Description	Checksum	Actions			
jinstall-host-qfx-5e-flex-x86-64-22.2R3-S3.14-secure-signed.tgz $^{\underline{URL}}$	JUNOS	external	N/A	QFX5K 22.2R3-S3.14 (JVD)		° •			

Figure 8: Register OS Image by Uploading or Provide Image URL

	\$
Register Device OS Image	
Platform *	
	•
Description *	s
 Upload Image Provide Image URL Image * 	.s
Drag and drop file here or choose file by clicking the button.	Choose File
Checksum	······································
SHA512 checksum (128 characters)	22
	Upload

Apstra Fabric Provisioning

Check Discovered Devices and Acknowledge the Devices.

Devices > Managed Devices

Once the offbox agent has been added and the device information has been collected, click the checkbox interface to select all the devices and then click acknowledge. This places the switch under the management of the Apstra server.

Finally, ensure that the pristine configuration is collected once again as Apstra adds the configurations for LLDP and RSTP.

Figure 9: Acknowledge Devices to Manage in Apstra

☆ A → Devices → Managed Devices			
	Create Onbox Agent(s)	Create Offbox Agent(s)	Advanced Settings
→ Query: All		1-22	of 22 << 1 > >>
Device 🖉 F 4 🖬 🗹 🗑 Agent 🗸 🛱	Ŧ	Columns (15/17) 👻	Page Size: 25 •
Acknowledge selected systems selected only unselected only			

Once a switch is acknowledged, the status icon under the "Acknowledged?" table header changes from a red X to a green checkmark. Verify this change for all switches. If there are no changes, repeat the procedure to acknowledge the switches again.

		Agent Information							
Management IP \$	Device Key \$	Device Profile \$	OS \$	State \$	Comms \$	Acknowledged? \$	Type \$	Agent Profile \$	Apstra Version \$
10.92.72.128	XC3623120013	Juniper_QFX5220-32CD	Junos 22.2R3-S3.13-EVO	IS-ACTIVE	¥	0	OFFBOX	root_user	AOS_4.2.1_OB.207
10.92.72.37	XC3622260031	Juniper_QFX5220-32CD	Junos 22.2R3-S3.13-EVO	IS-ACTIVE	¥	0	OFFBOX	root_user	AOS_4.2.1_OB.207
10.92.76.31	YR3622410026	Juniper_QFX5130-32CD	Junos 22.2R3-S3.13-EVO	IS-ACTIVE	¥	0	OFFBOX	root_user	AOS_4.2.1_OB.207
10.6.1.150	XH3719030120	Juniper_QFX5120-48Y	Junes 22.2R3-S3.18	IS-ACTIVE	¥	0	OFFBOX	root_user	AOS_4.2.1_OB.207
10.92.76.30	YR3622410005	Juniper_QFX5130-32CD	Junos 22.2R3-S3.13-EVO	IS-ACTIVE	¥	0	OFFBOX	root_user	AOS_4.2.1_OB.207
10.6.1.58	XH3719090097	Juniper_QFX5120-48Y	Junos 22.2R3-S3.18	IS-ACTIVE	Ψ	0	OFFBOX	root_user	AOS_4.2.1_OB.207
10.6.1.57	XH3719090062	Juniper_QFX5120-48Y	Junes 22.2R3-S3.18	IS-ACTIVE	¥	0	OFFBOX	root_user	AOS_4.2.1_OB.207

Figure 10: Devices Managed by Apstra

NOTE: After a device is managed by Apstra, all device configuration changes should be performed using Apstra. Do not perform configuration changes on devices outside of Apstra, as Apstra may revert those changes.

Apstra Web UI: Identify and Create Logical Devices, Interface Maps with Device Profiles

In the following steps, we define the 3-stage fabric with the Juniper Apstra baseline architecture and devices. Before provisioning a blueprint, a replica of the topology is created. In the following steps, we define the ERB data center reference architecture and devices:

- This involves selecting logical devices for spine, leaf, and border leaf switches. Logical devices are abstractions of physical devices that specify common device form factors such as the amount, speed, and roles of ports. Vendor-specific information is not included, which permits building the network definition before selecting vendors and hardware device models. The Apstra software installation includes many predefined logical devices that can be used to create any variation of the logical device.
- Logical devices are then mapped to device profiles using interface maps. The ports mapped on the interface maps match the device profile and the physical device connections. Again, the Apstra software installation includes many predefined interface maps and device profiles.
- Finally, the racks and templates are defined using the configured logical devices and device profiles, which are then used to create a blueprint.

The Juniper Apstra User Guide explains the device lifecycle, which must be understood when working with Apstra blueprints and devices.

NOTE: The 3-stage design provisioning steps use the Apstra Data Center Reference design.

Navigate to **Design** > **Logical Devices**, then review the devices listed based on the number of ports and speed of ports. Select the device that most closely resembles the device that should be added, then clone the logical device.

NOTE: System added or default logical devices cannot be changed.

The following table shows the device roles, logical device types, ports, and connections created for the 3-Stage Fabric with Juniper Apstra JVD lab in this document. The Port Groups column depicts the minimum connections required for this lab. This will vary from the actual port groups these switches can provide.

Table 4: Logical Device Port Speeds and Connection for Each Fabric Device

Device Role	Port Group Connections1	Port Groups2	Connected To
Spine	Superspine/Spine/Leaf/Access/	5 x 100 Gbps (each	2 Border Leaf switches
	Generic	spine)	3 Server Leaf switches

Device Role	Port Group Connections1	Port Groups2	Connected To
Server Leaf (single)	Superspine/Spine/Leaf/Access/ Generic	2 X 100 Gbps 5 x 10 Gbps	2 Spine 2 Servers (Generic)
Server Leaf switches (2 ESI leaf switches)	Superspine/Spine/Leaf/Access/ Generic	4 X 100 Gbps (both leaf switches) 5 X 10 Gbps	2 Spine 4 Servers (Generic)
Border Leaf switches	Superspine/Spine/Leaf/Access/ Generic	6 X 10 Gbps 4 X 100 Gbps (both leaf switches)	6 Servers 2 Spine

Table 4: Logical Device Port Speeds and Connection for Each Fabric Device (Continued)

1 For port group connections, these can vary depending on the role and devices connected.

2 For port groups, the number of ports can vary depending on connections and speed.

Device Profiles

For all devices covered in this document the device profiles (defined in Apstra found under **Devices** > **Device Profiles**) were exactly matched by Apstra while adding devices into Apstra, as covered in "Apstra: Management of Junos OS Device" on page 12. During the validation of the supported devices, there are instances where device profiles had to be custom-made to suit the line card setup on the device, for instance, QFX5700. For more information on device profiles, refer to the Apstra User Guide for Device Profiles.

NOTE: The device profiles covered in this JVD document are not modular chassis-based. For modular chassis-based devices such as QFX5700, the line card Profiles and Chassis Profile are available in Apstra and linked to the device profile. These cannot be edited; however, they can be cloned, and custom profiles can be created for line card, Chassis, and Device profile as shown below in Figure 11 on page 21 and Figure 12 on page 21.

Figure 11: QFX5700 Device Profile Linked to Chassis Profile and Linecard Profile

Edit Device Profile

	Device profiles you intend to u	need to accurately model various characteristics of a switch model. Make sure you update the profile to match t se this profile for.
Updating	the device profile	ports may not be allowed because it is referenced by MUST-DC1-Border_QFX5700-BL1-10_6_1_113 interface
Summa	ry	Type Modular -
Linecar	ds	Name *
		MUST-DC1-QFX5700-10_6_1_113_BL1
		Chassis Profile
		Juniper QFX5700 2 slot

Figure 12: QFX5700 Device profile linked to Linecard Profile

Edit Device Profile

Device profiles need to accurately model various characteristics of a switch model. Make sure you update the profile to match the new switch model(s) you intend to use this profile for.										
Updating the de	vice profile p	oorts may not be a	llowed because it is referenced by MUST-DC1-Border_QFX5700-BL1-10_6_1_113 interface map.							
Summary		Installed Lineca	rds							
Linecards		Slot 0	MUST_QFX5700_Juniper JNP-FPC-16C-LC	T						
	Slot 1 MUST_QFX5700_Juniper JNP-FPC-20Y-ODD-LC									
	+ Add Linecard									
			Upd	late						

Spine Logical Device and Corresponding Interface Maps

The spine logical device is based on QFX5220-32CD (Junos OS). For the purposes of this solution, seven 100G links are used to connect to leaf switches. As shown in Figure 13 on page 22 12 ports of 100 Gbps are enough for five spine to leaf connections.

Figure 13: Apstra Logical Device Spine Configuration

ame																			
MUST_QF	X5220-1	2x10	0G_10	x10G	_5x4	0_5x4	00_2×	25G											
PANEL#	1																		
TOTAL PORT GROUPS														Cor	nected t				
34 ports			12 x 100 Gbps Superspine • Spine • Leaf • Unused			Sup Acc	10 x 10 Gbps 5 x 40 Gbps Superspine • Spine • Leaf • Superspine • Spine • Leaf • Access • Unused • Generic Access • Unused • Generic				2 x Superspin Access •	25 Gbp e • Spine • Unused • C)S Leaf ● Generic						
				Supe	5 x ·	400 e • Sp Unuse	Gbp ine • ed	DS Leaf •											
1 3	5 7	9	11	13 1	5 1	7 19	21	23	25 2	27 2	9	31 3	3						
	A A	10	12	14 1	6 1	8 20	22	24	26 2	28 3	ю :	32 3	4						

The spine logical device ports are mapped to the Device Profiles using the Interface map as shown below. The ports mapped on the interface maps match the device profile and the physical device connections.

Figure 14: Spine Interface Map

☆ 希 → Design → Interface Maps → MUST_QFX5220-32CD-Sp	bine									
← back to list					ß					
Name	MUST_QFX5220-32CD-Spine									
Logical device	MUST_QFX5220-12x100G_10x10G_5	x40_5x400_2x25G 🅐								
Device profile	Juniper_QFX5220-32CD I									
Interface map preview										
SUMMARY					Co					
12 x 100 Gbps 10 > Superspine • Spine • Leaf • Unused Superspine • Spine • Leaf	★ 10 Gbps aaf • Access • Unused • Generic	5 x 40 Gb Superspine • Spine • Leaf • Acce	Cbps 2 x 25 Gbps 5 x 400 Access • Unused • Generic Superspine • Spine • Leaf • Access • Unused • Generic Superspine • Spine							
INTERFACES Click or Hearfords to lagger the debis 4 3 4 4 1 10 12 20										
MAPPING										
Logical Device Click on port to taggle referenced interface details			Device Profile Click on port to taggle reference	ed interface details						

Server Leaf switches Logical Device and Interface Maps

For the purposes of this JVD, there are three QFX5120-48Y server leaf switches. Two of them are ESIsupporting switches, and one of them is a non-ESI LAG switch. All three server leaf switches are connected to each spine using 100 GB interfaces, and the 10 GB interfaces connect to the generic servers.

For a single (non-redundant) leaf switch, no ESI is used, and only LACP (Active) is configured.

Figure 15: Apstra Single Leaf Logical Device

Logical Device Preview

Logical Device Preview

lame				
MUST-single-QFX5120-54	x10-2x100-2x40			
PANEL #1				
TOTAL	PORT GROUPS			Connected to -
58 ports	54 x 10 Gbps Superspine • Spine • Leaf • Access • Peer • Unused •	2 x 100 Gbps Superspine • Spine • Leaf • Access • Peer • Unused •	2 x 40 Gbps Superspine • Spine • Leaf • Access • Peer • Unused •	
	Generic	Generic	Generic	
1 3 5 7 9	1 13 15 17 19 21 23	25 27 29 31 33 35 3	7 39 41 43 45 47 49	51 53 55 57
2 4 6 8 10	12 14 16 18 20 22 24	26 28 30 32 34 36 3	8 40 42 44 46 48 50	52 54 56 58

For ESI (redundant) leaf switches, ESI Lag is used for multi-homing. ESI lag is configured under the Rack in **Design** > **Rack Types**.

Figure 16: Apstra Server Leaf Switches Logical Device

Name MUST-QFX5120-52x10-4x100-2x40										
PANEL #1										
TOTAL	PORT GROUPS	Connected to -								
58 ports	52 x 10 Gbps 4 x 100 Gbps 2 x 40 Gbps Superspine • Spine • Leaf • Superspine • Spine • Leaf • Superspine • Spine • Leaf • Access • Peer • Unused • Access • Peer • Unused • Access • Peer • Unused • Generic Generic Generic									
1 3 5 7 9 11	13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57									
2 4 6 8 10 12	14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58									

The server leaf logical device is mapped to the device profile as below.

Figure 17: Single Server Leaf Switches Interface Map

Interface Map Preview

Name	MUST_singleLeaf_ESI5120-54x10-2x100						
Logical device	MUST-single-QFX5120-54x10-2x100-2x40						
Device profile	Juniper_QFX5120-48Y						
Interface map preview							
SUMMARY	Connected to -						
54 x 10 Gbps 2 x 100 Gbps Superspine • Spine • Leaf • Access • Peer • Unused • Generic Superspine • Spine • Leaf • Access • Peer • Unused • Generic							
54 x 10 C Superspine • Spine • Leaf • Access	bps 2 x 100 Gbps • Peer • Unused • Generic Superspine • Spine • Leaf • Access • Peer • Unused • Generic						
54 x 10 G Superspine • Spine • Leaf • Access 2 x 40 G Superspine • Spine • Leaf • Access	Sbps 2 x 100 Gbps • Peer • Unused • Generic Superspine • Spine • Leaf • Access • Peer • Unused • Generic bps • Peer • Unused • Generic						
54 x 10 G Superspine • Spine • Leaf • Access 2 x 40 G Superspine • Spine • Leaf • Access INTERFACES Click on interface to taggle	Sipps 2 x 100 Gbps • Peer • Unused • Generic Superspine • Spine • Leaf • Access • Peer • Unused • Generic bps • Peer • Unused • Generic • the details • the details						

Unused interfaces (4)

Figure 18: Server Leaf Switches Interface Map for ESI Leaf Switches

Interface Map Preview

Name	MUST_Leaf_ESI5120-52x10-4x100-2x40						
Logical device	MUST-QFX5120-52x10-4x100-2x40						
Device profile	Juniper_QFX5120-48Y						
Interface map preview							
SUMMARY	Connected to +						
52 x 10 Gbps 4 x 100 Gbps Superspine • Spine • Leaf • Access • Peer • Unused • Generic Superspine • Spine • Leaf • Access • Peer • Unused • Generic							
2 x 40 G Superspine • Spine • Leaf • Access	DDS ● Peer ● Unused ● Generic						
INTERFACES Click on interface to toggle							
1 3 5 7 9 11 13 15 2 4 6 8 10 12 14 16	17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 58						
 Unused interfaces (1) 							

NOTE: In this case, the single leaf and ESI server leaf pairs both have the same device profile, but due to differences in how the physical ports on the switches are connected towards the servers and the spine, two different logical devices were designed.

Border Leaf Switches Logical Device and Interface Maps

The border leaf logical device is a representation of the QFX5130-32CD switches used in this design. The physical cabling determines the ports allocated for the interface Maps.

Figure 19: Border Leaf Switches Logical Device

Logical Device Preview

Name		
MUST-QFX5130-32CD		
PANEL#1		
TOTAL	PORT GROUPS	Connected to -
32 ports	4 x 10 Gbps8 x 40 Gbps20 x 100 GbpsSuperspine • Spine • Leaf • Access • Peer • Unused • GenericSuperspine • Spine • Leaf • Peer • Unused • GenericSuperspine • Spine • Leaf • Peer • Unused • Generic	
1 3 5 7 9 11	13 15 17 19 21 23 25 27 29 31	
2 4 6 8 10 12	14 16 18 20 22 24 26 28 30 32	

Figure 20: Border Leaf Switches Interface Map

Interface Map Preview

Name	MUST-DC1-Border-Qfx5130		
Logical device	MUST-Border-jvdv2		
Device profile	Juniper_QFX5130-32CD		
Interface map preview			
SUMMARY			Connected to -
SUMMARY 64 x 10 Superspine • Spine • Leaf • Acce	Gbps ss • Peer • Unused • Generic	8 x 40 Gbps Superspine • Spine • Leaf • Peer • Unused • Generic	Connected to -
64 x 10 Superspine • Spine • Leaf • Acce 8 x 100 Gbps Superspine • Spine • Leaf	Gbps ss • Peer • Unused • Generic	8 x 40 Gbps Superspine • Spine • Leaf • Peer • Unused • Generic	Connected to •

The rest of the Logical Devices are described below. The interface maps are optional and can be omitted.

Generic Servers Logical Device

Generic servers define the network interface connections from the servers connected to the leaf switches (border and single).

Logical devices for the servers used are already pre-defined within Apstra. A similar generic system can be used for DCI; however, DCI will be covered in a separate JVD Extension document.

External Routers

External routers are connected to the border leaf switches.

Apstra does not manage external routers such as MX Series devices; hence, the MX Series router is classified as an external generic server with the relevant port and speed configuration.

NOTE: A generic external system is added to the blueprint after a blueprint is created. An interface map is not needed for generic servers or external routers. The connectivity and features of external routers is beyond the scope of this document.

Apstra Web UI: Racks, Templates, and Blueprints–Create Racks

After defining the logical devices and Interface maps, the next step is to create racks to place the logical devices in rack formation. The default design for this solution is two spines, five server leaf switches, and two border leaf switches. Any rack design can be created and used any number of times, so long as the spine switches have enough ports to support it.

In Apstra, create racks under **Design** > **Rack Types**. For this solution, there are four racks. One rack for border leaf switches and three racks for server leaf switches. For more information on creating racks, refer to the Juniper Apstra User Guide.

For this design, the L3 Clos rack structure is as follows:

Server Leaf Switch (Single Leaf)

Figure 21: Single Leaf Rack Without ESI

☆ 谷 → Design → Rack Typ	pes → dc1_must_singlev2		
← back to list			
Evolution Compact View			
Expanded view Compact view			
Topology Preview			
	Leafs, (1) Access switches	single_leaf	
	Generic Systems, (3) esi	single_generic nsxt_overlay	generic-000000001
Summary			
	Display Name	dc1_must_singlev2	
	Description	jvdv2 3-stage	
	Fabric Connectivity Design	L3 Clos	
Leafs			
	single_leaf		
	1 x 100 Gbps Links per spine		
	MUST-single-QFX5120-54x10-2x100-2x40		
			No ESI Lag configured for Single Leaf
Generic Systems			NO ESI Lag configured for Single Lean
			• •
	esi single generic	1 generic system	generic-000000001 1 generic system
	1 x 10 Gbps single_leaf_gen single-homed at si	ingle_leaf	1 x 10 Gbps link-000000001 single-homed at single_leaf
	AOS-4x10-1		A05-2x10-1
	1 x 10 Gbps nsxt_overlay single-homed at sing	1 generic system	
	AOS-4x10-1		
			

Server Leaf Switches (Two Leaf Switches)



Figure 22: Server Leaf Switches with ESI Lag for Multihomed

Border Leaf Switches

Figure 23: Border Leaf Switches Rack

☆ ♣ → Design → Rack Types →	dc1_borde_qfx5130			
Topology Preview				
	Leafs. (2) Access switches Generic Systems, (4) generic-O	border_leaf_1 E bo	nder, Just, 2 generic-00000002 nsst, everlay, 2	
Summary				
	Display Name Description Fabric Connectivity Design	dc1,borde,qh5130 Jodv2-3-stage L3 Gos		
Leafs				
Generic Systems	border_leaf 1 x 100 Gbps Links per spine MUST-Border-Ivel/2	53		
	generic-000000001 1 x 10 Gbps End-00000001 single-homed at bond Lid Made No Lid	1 generic system r_leaf (irst poor)	generic-000000002 1 x 10 Gbps Biolcoccoccos single-homed at border, leaf (second peer) UKS Made the UKS	1 generic system
	A05-2x10-1		A05-2x10-1	
	nsxt_overlay_1 1 × 10 Gbps Enk-00000001 single-homed at bed Loc Netex No Loc 1 × 10 Gbps Enk-00000002 single-homed at bed Loc Netex No C 1 × 10 Gbps Internet work with homed at bertler_les Loc Netex LACP (Active) AC5-2x10-1,x2	1 generic system er_leaf (first peer) er_leaf (second peer) ef	nsxt_overlay_2 1 x 10 Gbps Lick Mode: No. UKC 1 x 10 Gbps Biel-000000001 single-homed at border_leaf (incl peer) Lick Mode: No. UKC 1 x 10 Gbps Inst_overlag due homed at border_leaf Lick Mode: No. UKC 1 x 10 Gbps Inst_overlag due homed at border_leaf UKC Mode: No.UKC 1 x 10 Gbps Vick Mode: NO.Piketweit	1 powerk system

NOTE: Once the blueprint is created and functional, if you need to perform any changes to the racks follow this KB article: https://supportportal.juniper.net/s/article/Juniper-Apstra-How-to-change-Leaf-Access-Switch-of-existing-rack-after-Day2-operations?language=en_US. During validation, the border leaf rack was modified to validate all devices listed in Table 5 on page 81.

Create Templates

Templates define the structure and the intent of the network. After creating the racks, the spine links need to be connected to each of the racks. In this design, the rack-based templates are used to define the racks to connect as top-of-rack (ToR) switches (or pairs of ToR switches).

As described in the spine logical devices section, there are 100G links assigned to each server leaf and border leaf. The spine logical device is assigned in the template. Since there are no super spines in this design, this is left out of the templates. For more information on templates, refer to the Juniper Apstra User Guide.

NOTE: Templates are used as a base for creating the blueprints, which are covered in the next section. Templates are used only once in the lifetime of a blueprint. Hence changing the template doesn't modify the blueprint.

Figure 24: DC Rack-Based Template with Rack Assignment and Link Speed

Structure

Add racks

Rack Types *			
dc1_must_single (1x40 Gbps links to spines)	×	1	×
dc1_must_esi (1x40 Gbps links to spines)	×	1	×
dc1_must_border (1x40 Gbps links to spines)	×	1	×

	Name	dc1-3stage			
	Туре	冒 RACK BASED			
Topology Preview					
	Selected Rack Al - Expand Nodes? Show Links?	spine1 spine2 61_boxt, gfs130 61_most_sby2 border_leaf_1 border_leaf_2 generic-000000001_1 esl_generic_1_1 generic-000000002_1 esl_generic_2_1			
		nsxt_overlay_1 esi_generic_3_1 nsxt_overlay_1 nsxt_overlay_2 esi_generic_3_2 generic-000000001_1 generic-1_1 generic-1_2 nsxt_overlay_1			
Structure					
	Spines	2 of MUST_QFX5220-12x100G_10x10G_5x40_5x400_2x25G			
	Tags				
	Rack Types	1 of dc1_borde_qt/s130 (4 generic systems) 1 of dc1_must_esiv2 (8 generic systems) 1 of dc1_must_singlev2 (3 generic systems)			
Policies					
	Overlay Control Protocol	MP-EBGP EVPN			

Figure 25: Figure Rack-Based Template Structure

Blueprint

Each blueprint represents a data center. Templates are created under the **Design > Templates** section and will be available in the global catalog for the blueprints. Once the template is defined, it can be used to create a blueprint for the data center.

To create a blueprint, click on **Blueprints > Create Blueprint**. For more information on creating the blueprint, see the Juniper Apstra User Guide.

Figure 26: Create Blueprint with Dual Stack

Create Blueprint				
Blueprint parameters				
	Name*			
	must_blueprint_dc1			
	Reference Design "			
	O Datacenter			
	Freeform			
	Filter Templates All RACK BASED POD BASED COLLAPSED			
	Template *			
	dc1-3stage			
	Spine to Leaf Links Underlay Type IPv4 IPv6 RFC-5549 IPv4-IPv6 Dual Stack Spine to Superspine Links IPv4 IPv6 RFC-5549 IPv4-IPv6 Dual Stack			
Intent preview				
Expanded View Compact View				
Template Parameters	Name	dc1-3stage		
Topology Preview	Туре	RACK BASED		
Structure				

Navigate to **Blueprint** > **Staged**. The topology shown can be expanded to view all connections. From here, the blueprint can be provisioned under Staged.

Dashboard MA Analytics Staged The Uncommitted ((D)) Active	🕄 Time	Voyager		
Search_ Search_ Poiscal X ⁴ Virtual O Policies DCI Catalog E Tasks A: Connectivity Templates	්ම Fabric Sr	Q.		
Topology Nodes Links Interfaces Racks Pods	Layer	Uncommitted Changes	×	Selection Build
Q. Nodes Q. Links			Has Uncommitted Changes	
Selected Rack Selected Node			Topology Label	
Al ·			Name *	▲ 0/2 ASNs - Spines
🕑 Expand Nodes? 🕑 Show Lisls?				ASNs - Leafs
				A 0/2 Loopback IPs - Spines
spinet		spine2		▲ 0/5 Loopback IPs - Leafs
dc1_sode_gds130_001 dc1_borde_gfx5130_001_leaf1dc1_borde_gfx5130_001_leaf2dc1_must_esiv2_001_leaf1dc1_must_esiv2_00	01_leaf2	dc1_must_singlev2_001	lev2_001_leaf1	▲ 0/20 Link IPs (IPv6) - Spines<>Leafs
dc1_borde dc1_borde dc1_borde dc1_borde dc1_must_e dc	1_must_e	dc1_must_singlev2 dc1_must	singlev2 dc1_must_singlev2	▲ 0/20 Link IPs - Spines<>Leafs
dc1_must_e dc1_must_e dc1_must_e dc1_must_e	1_must_e			

Figure 27: Blueprint Created and Not Provisioned

As shown above, the blueprint is created but not provisioned. The topology can be inspected for any discrepancies, and if so, then the blueprint can be recreated after fixing the template or the rack. Alternatively, navigate to **Staged > Racks** to edit the rack by following the steps mentioned in this article.

×
Apstra Web UI: Provisioning and Defining the Network

Once the blueprint is created, it means that the blueprint is ready to be staged. Review the tabs under the blueprint created.

To start provisioning, click on the **Staged tab > Physical** and then click Build from the right-hand side panel. For more information, refer to the Juniper Apstra User Guide.

Das 4. 3 0 Resources Q. Link 5. Edit and Status Assign turns green 0 0 41 N dc1 must e dc1 must e... dc1_must_e_. dc1 must singlev2. dc1 b de1 n dc1 m 2 ick IPs - Spine dc1_must_e... dc1_must_e...

Figure 28: Blueprint Assign Resources Under Build

Assign Resources

The first step is assigning IPs created in this **Resources** section. For this design, below are the resource values used:

- 1. Click Staged > Physical > Build > Resources and update as below:
 - a. DC1 ASNs-Spines & leaf switches: 64512 64999
 - b. Loopback IPs-Spines & Leaf switches: 192.168.255.0/24
 - c. Link IPs-Spines <> Leaf switches: MUST-FABRIC-Interface-IPs DC1-10.0.1.0/24

ASN Pool Preview		
Name	DC1	
Status	🗢 IN USE	
Total Usage	1.43N	
Range Usage	143N	64512 - 64999
IP Pool Preview		
Name	MUST-FABRIC-Loopbacks DC1	
Status	🗢 IN USE	
Total Usage	2.73%	
Per Subnet Usage	2.73%	192.168.255.0/24
IP Pool Preview		
Name	MUST-FABRIC-Interface-IPs DC1	
Status	🗢 IN USE	
Total Usage	7.81%	
Per Subnet Usage	7.81%	10.0.1.0/24

Figure 29: ASN, Loopback IPs and Fabric Link IP Pools

Figure 30: Resources Assigned

Dashboard Analytics Staged G Uncommitted (D) Active	") Time Voyager		
O Search C O	ৃত্য ates ্য® Fabric Settings	٩	Assign Resources
Topology Nodes Links Interfaces Racks Pods	Layer Uncommitted Changes	× Has Uncommitted Changes	
Selected Rack Selected Node		Topology Label Name *	 ■ 2/2 ASNs - Spines
Capand Nodes?			5/5 ASNs - Leafs 2/2 Loopback IPs - Spines
spire1 e1_web_ab109_001 e1_borde_ch5130_001_keb1 e1_borde_ch5130_001_keb1 d1_borde_ch5130_001_keb1	spine2 et.mat.ingle2.001 esiv2.001 jeat2	singlev2,001,leaf1	 5/5 Loopback IPs - Leafs 20/20 Link IPs (IPv6) - Spines<>Leafs
del borde del borde del borde del borde del most e del	dc1_must_edc1_must_singlev2dc1_u	must_singlev2 dc1_must_singlev2	€ 20/20 Link IPs - Spines⇔Leafs

Assign Interface Maps to Switches

From the blueprint, navigate to **Staged > Physical > Build > Device Profiles**.

Next, assign devices to interface maps created in the section "Apstra Web UI: Identify and Create Logical Devices, Interface Maps with Device Profiles" on page 18 of this document.

Figure 31: Blueprint Assign Interface Maps in Device Profiles Under Build

Dashboard Analytics Analytics Staged G) Active	S Time Voyager		
0 Searth.		Q,	3
Physical 2-5 Virtual © Policies @ DCI Catalog Tasks Connectivity Templates	è Fabric Settings	4. Device Profile and Interface Maps	1
Topology Nodes Links Interfaces Racks Pods	Layer Build: Device Profile	s - MUST-QFX5120-52x10-4x100-2x40 N	Selection Build
Q, Nodes Q, Livits		Assigned Not Assigned	
Selected Rock Selected Node		Topology Label	Manage Interface Maps 🕈
🕑 Ergand Nodes? 🥥 Brow Links?		5. Assign	2/2 MUST_QFX5220-12x1006_10x106_5x40_5x400_2 MUST_single-QFX5120-54x10-2x100-2x40
	spine2 det.must.singlev2.001		6. Status
dc1_borde_qtx5130_001_ieaf1 dc1_borde_qtx5130_001_leaf2 dc1_must_en/v2_001_leaf1 dc1_must_en/v2_001	1.Jea/2	dc1_must_singlev2_001_leaf1	Node Name : turns green
del_borde del_borde del_borde del_borde del_mast.e del_mast.e del_mast.e del_mast.e del	must_e dc1_must_singlev2	dc1_must_singlev2 dc1_must_singlev2	dc1,mast,esh2,001,had1 Juniper,QFX5120-48Y dc1,mast,esh2,001,had2 Juniper,QFX5120-48Y
del_mat_a_ del_mat_a_ del_mat_a_ del_	.must_e		A 0/2 MUST-Border-jvdv2

Figure 32: Interface Maps Assigned

Dashboard Analytics Staged G. Uncommitted (CS) Active S Time Voyager		
Seach Q Seach Q Physical X ⁶ Virtual O Policies DCI Catalog E Tasks A: Connectivity Templates d Catalog E Tasks A: Connectivity Templates d DCI DCI	Assign Interface Maps	Find by t
Q. Nodes Q. Links		
Selected Ruck Selected Ruck Tepdage Label	Manage Interface Maps 🕈	
	@ 2/2 MUST_QFX5220-12x1006_10x106_5x40_5x	x400_2x25G
Copand Noden? Show Links?	0 1/1 MUST-single-QFX5120-54x10-2x100-2x40	
section and sectio	2/2 MUST-QFX5120-52x10-4x100-2x40	
105 (respect) (area, tab.	© 2/2 MUST-Border-jvdv2	
41, berde gl/5130,007 hart dt. borde, gl/5130,001 hart dt. borde, gl/5130,001 hart dt. mast ski/2,001 hart	g 1.5	2 of 2
del borde_ del borde_ del borde_ del borde_ del borde_ del mate del mate del mate del mate del mate del mate.	Node Name 0 Device Profile 0	
dc1.must_e dc1.must_e dc1.must_e	dc1_borde_gfx5130_001_leaf1 Juniper_QFX5130	0-32CD
	dc1,borde,qfx5130,001,leaf2 Aniper,QFX5130	0-32CD
	AOS-4x10-1 (optional)	
	▲ 0/2 AO5-2x10-1_v2 (optional)	

NOTE: The assignment of interface maps to generic systems or servers is optional. The status of these parameters will be marked RED and they are also marked as optional.

Assign the System IDs and the Correct Management IPs

From the blueprint, navigate to **Staged > Physical > Build > Devices** and click on Assigned System IDs. The system IDs are the devices serial numbers.



Figure 33: Blueprint Staged Assign System IDs Under Build

NOTE: The device hostname and the display name (on Apstra) for each node or device is different these can be changed using Apstra.No system IDs are assigned to generic servers and external routers, as these are not managed by Apstra.

Ensure all the devices are added to Apstra under **Devices** > **Managed Devices** before assigning system IDs (serial numbers of the devices).

Review Cabling

Apstra automatically assigns cabling ports on devices that may not be the same as physical cabling. However, the cabling assigned by Apstra can be overridden and changed to depict the actual cabling. This can be achieved by accessing the blueprint, navigating to **Staged > Physical > Links**, and clicking the **Edit Cabling Map** button. For more information, refer to the Juniper Apstra User Guide.

Figure 34: Review and Edit Cabling

Œ) Dashboard	Analytics	4	🖬 Sta	ged	≅ _© Uncomm	nitted	(m)	Active	S Time	Voyager				
@ s	Physical 3- Virtual	Ø Policies	5	DCI	2 E Catalo	g ⊟ Tasks		Connectivity	y Templates	্ ⊄⊛ Fabric Se	ettings				
Topo Q, Ni Selected	ology Nodes Links I odes Q, Links Rack	interfaces	Racks	Pods Mod mat	ify cablir tch physi cabling	ng to ical		L	iyer Un	committed Changes	Has Un	committed (X Changes	Selection	Build
 Filter se	± t t t t = t	e unsel	ected or	ıly				Alter disco to	nativel overed assign	y use fetch LLDP data cabling.	< 1	2 3	> »	© 7/7 A	SNs - Leafs SNs - Generics
0	Name 6	Role 0	Speed	Ture 0		Endpo	int 1 0				Endp	oint 2 0		© 2/2 L	oopback IPs - Spines
selected	runne v	Hole U	•	ings ¥	Name ©	Role ©	0	IPv4 0	IPv6 0	Name 0	Role ©	Interface 0	IPv4 0	© 2/2 L	oopback IPs (IPv6) - Spine
	DCI-DC2 BL1<-	То				Remote								Ø 7/7 L	oopback IPs - Leafs

It is best practice to review the switch names, including the generic servers, to ensure the naming is consistent. To review and modify the names of the devices, navigate to **Staged > Physical > Nodes** and click on the name of any of the devices listed to present a screen with the topology and connections to the device along with the panel on the right that shows the device properties, tags, and so on, as shown in Figure 35 on page 40.

Figure 35: Review Device Links, Properties



Configlet and Property Sets

Configlets are configuration templates defined in the global catalog under **Design > Configlets**. Configlets are not managed by Apstra's intent-based functionality, and these are to be managed manually. For more information on when not to use configlet refer to the Juniper Apstra User Guide. Configlets should not be used to replace reference design configurations. Configlets can be declared as a Jinja template of the configuration snippet, such as Junos configuration JSON style or Junos set-based configuration. For more information on designing a configlet, refer to the Apstra Configlets user guide.

NOTE: Improperly configured configlets may not raise warnings or restrictions. It is recommended that configlets are tested and validated on a separate dedicated service to ensure that the configlet performs exactly as intended.Passwords and other secret keys are not encrypted in configlets.

Property sets are data sets that define device properties. They work in conjunction with configlets and analytics probes. Property sets are defined in the global catalog under **Design > Property Sets**.

NOTE: Configuration templates in Freeform blueprints also use property sets, but they're not related to property sets in the design catalog.

Configlets and property sets defined in the global catalogue need to be imported into the required blueprint and if the configlet is modified then the same needs to be reimported into the blueprint, as is the case with property sets too. The following figure shows configlets and property sets located on a blueprint.

Figure 36: Import Configlet into Blueprint

☆ ★ Blueprints → must_b	lueprint_dc1 > Staged > Physical	Build > Configlets			
① Dashboard	Analytics	d ^B B Uncomm	itted ((B)) Active	Time Voyager	
Ø Search		1.		٩	4. Configlet
Physical 25. Virtual	⊘ Policies @ DCI	Catalog 🗄 Tasks	🚆 Connectivity Templates	∜⊚ Fabric Settings	3.
Z. Topology Nodes Links	Interfaces Racks Pods		Layer Uncommi	tted Changes	× Selection Build
Q, Nodes Q, Links				Has Uncommitte	ed Changes
Selected Rack	Selected Node			configlet	Topology Car
AI ·	AI •			Name	I/1 Condition 1
Expand Nodes? Show Links?					♦ 1/1 Condition 2
	DCI-DC2_BL2		DCI-DC2_BL1		● 1/1 Condition 3
		mx-router1			© 3/3 Condition 4

Figure 37: Import Property Set into Blueprint

☆ ☆ > Blueprints + must_blueprint_dcl + Staged + Catalog + Property ① Dashboard △ ▲ ▲ ▲ ▲ ▲ ▲	Prty Sets	>_ 疾
Search Search	1. Q	T Find by tags
Logical Devices Interface Maps Property Sets Configlets AAA	Servers Tags	Import Property Set

During 3-stage validation, several configlets were applied either as part of the general configuration for setup and management purposes (such as nameservers, NTP, and so on).

Fabric Setting

Fabric policy

This option allows for fabric-wide setting of various parameters such as MTU, IPv6 application support, and route options. For this JVD, the following parameters were used: View and modify these settings within the blueprint **Staged > Fabric Settings > Fabric Policy** within the Apstra UI.

Figure 38: Fabric Policy Settings

A → Blueprints → must_blueprint_dc1 → Staged → Fabric Settings	Fabric Policy	
) Dashboard 🖉 Analytics 🔹 Staged	ن المحمد (۵) Active المحمد المحم المحمد المحمد	Voyager
	MTU Settings	
	Fabric MTU [®]	Default
	IP Links to Generic Systems $\text{MTU}^{\textcircled{0}}$	9100
	Default SVI L3 MTU [®]	9000
	Fabric Design	
	IPv6 Applications [®]	Enabled
	Routing Zone Footprint Optimization ${}^{\textstyle \Theta}$	Disabled
	Route Options	
	Max External Routes Count [®]	Unlimited
	Max MLAG Routes Count®	Unlimited
	Max EVPN Routes Count®	Unlimited
	Max Fabric Routes ${\rm Count}^{\it O}$	Unlimited
	Generate EVPN host routes from ARP/IPV6 ND $\text{ARP}^{\textcircled{O}}$	Disabled
	Vendor Specific	
	Junos EVPN routing instance mode $^{\Theta}$	MAC-VRF
	Junos EVPN Next-hop and Interface count maximums $^{\ensuremath{\Theta}}$	Enabled
	Junos Graceful Restart [®]	Disabled
	Junos EX-Series Overlay ECMP®	Disabled
	Anti Affinity	
	Mode	Disabled

1. To simulate moderate traffic in datacenter, traffic scale testing was performed refer Table 6 on page 84 for more details. The scale testing was performed on QFX5120-48Y switches.

The setting **Junos EVPN Next-hop and Interface count maximums** was also enabled, which allows Apstra to apply the relevant configuration to optimize the maximum number of allowed EVPN overlay next-hops and physical interfaces on leaf switches to an appropriate number for the data center fabric. Along with this the configlet is also used to set a balanced memory allocation for Layer 2 and Layer 3 entries as shown in Figure 39 on page 44.

For more information on these features, refer to:

• https://www.juniper.net/documentation/us/en/software/junos/multicast-l2/topics/topic-map/ layer-2-forwarding-tables.html

- https://www.juniper.net/documentation/us/en/software/junos/evpn-vxlan/Other/interface-num-edit-forwarding-options.html
- https://www.juniper.net/documentation/us/en/software/junos/cli-reference/topics/ref/ statement/next-hop-edit-forwarding-options-vxlan-routing.html

For QFX5120 leaf switches configuration:

<pre>{master:0}</pre>
root@dc1-esi-001-leaf1> show configuration forwarding-options display set
set forwarding-options vxlan-routing next-hop 45056
set forwarding-options vxlan-routing interface-num 8192
set forwarding-options vxlan-routing overlay-ecmp
<pre>{master:0}</pre>
<pre>root@dc1-esi-001-leaf1> show configuration chassis forwarding-options display set</pre>
set chassis forwarding-options 12-profile-three

Figure 39: Configlet on Leaf Switches for Balanced Memory

Configlet Preview

Expanded View	Compact View		
Configlet Param	neters		
	Name	Forwarding-Options Scale Settings	
	Node Condition	hostname in ["dc1-esi-001-leaf1", "dc1-single-leaf1", "dc1-esi-001-leaf2"]	
	Application Scope	dc1_esi_001_leaf1, dc1_esi_001_leaf2, dc1_single_leaf1	
Junos: SET BASI	ED SYSTEM		
	Config	Configlets that require device context may be rendered incorrectly here. Please refer to the rendered config of each device for more details.	
		set chassis forwarding-options l2-profile-three	

2. For the non-EVO leaf switches, the setting Junos EVPN routing instance mode was also enabled, as this is the default setting Apstra applies to all new blueprints from Apstra 4.2. For any blueprint created prior to Apstra 4.2, post-Apstra upgrade of the default switch for non-EVO switches is allowed. However, it is recommended that MAC-VRF normalize the configuration in a mixed setup of Junos OS and Junos OS Evolved. A VLAN-aware routing instance 'evpn-1' for MAC-VRF is created for only non-EVO Junos devices. This option doesn't affect Junos OS Evolved devices as Junos OS Evolved can only support MAC-VRF, and the same is already implemented by default.

NOTE: If the blueprint is live and running in a production network, it is recommended to perform the above setting changes to MAC-VRF routing instance mode during a maintenance window as it is disruptive and requires a 'reboot' of non-EVO Junos leaf switches, in this case the QFX5120s.

For QFX5120 Leaf switches configuration:

```
{master:0}
root@dc1-esi-001-leaf1> show configuration forwarding-options | display set
set forwarding-options evpn-vxlan shared-tunnels
{master:0}
root@dc1-esi-001-leaf1> show configuration routing-instances evpn-1 | display set
set routing-instances evpn-1 instance-type mac-vrf
set routing-instances evpn-1 protocols evpn encapsulation vxlan
set routing-instances evpn-1 protocols evpn default-gateway do-not-advertise
set routing-instances evpn-1 protocols evpn duplicate-mac-detection auto-recovery-time 9
set routing-instances evpn-1 protocols evpn vni-options vni 10050 vrf-target target:10050:1
set routing-instances evpn-1 protocols evpn vni-options vni 10108 vrf-target target:10108:1
set routing-instances evpn-1 protocols evpn vni-options vni 10400 vrf-target target:10400:1
```

Anomalies for "Device reboot required" will be raised for non-EVO leaf switches when the MAC-VRF routing instance mode is enabled. To fix these anomalies, reboot the leaf switches affected by the above change from the CLI.

Figure : Anomalies Raised by Apstra for QFX5120 Device Reboot After Change to MAC-VRF

Juniper Shared Runnel Reboot Detector	Device reboots required		modal metric system_id	Amper_QPIS120-48Y shared_ternii_mode xx13710060097	Anomalous value: Disabled, reboot is required Actual value: Disabled, reboot is required
Juriper Shared Turned Reboot Detector	Device rebots required		model metric system_id	Jumper, CBVS520-48Y shared, turnel, mode xxxx7190000062	Anomolecul value: Disabled, reboet is required Actual value: Disabled, reboet is required
Juriper Shared Turnel Reboot Detector	Device reboots required		model metric system_id	Aniper, QPK5120-48Y shared, turnel, mode Xx43719000120	Anomalous value: Disabled, relocat is required Actual value: Disabled, relocat is required
Probe *	Stagu ‡	Tags ¢	Properties		Values

Commit the Configuration

Once the cabling has been verified, the fabric is ready to be committed. This means that the control plane is set up, and all the leaf switches are able to advertise routes through BGP. Review changes and commit by navigating from the blueprint to **Blueprint > <Blueprint-name> Uncommitted**.

As of Apstra 4.2, a new feature is to perform a commit check before committing, which is introduced to check for semantic errors or omissions, especially if any configlets are involved.

Note that if there are build errors, those need to be fixed. Otherwise, Apstra will not commit any changes until the errors are resolved.

For more information, refer to the Juniper Apstra User Guide.

☆ 希→ Blueprints→ must_blu ② Dashboard	eprint_dc1 > Un Analytics	committed > Commit Check	O Uncommitted ((D)) Active	③ Time Voyager				📽 Commit	
Commit Check Commi									
System name \$	Role ‡	Hostname \$	Device profile \$	Serial number \$	Status 🕆	Last commit check result \$	Result Validity [®] \$	Actions	
dc1_border_leaf1	leaf	dc1-border-leaf1	Juniper_QFX5130-32CD	YR3622410005	SUCCESS	Success	FRESH	Ø 8 =	
dc1_border_leaf2	leaf	dc1-border-leaf2	Juniper_QFX5130-32CD	YR3622410026	SUCCESS	Success	FRESH	Ø 0 =	
dc1_esi_001_leaf1	leaf	dc1-esi-001-leaf1	Juniper_QFX5120-48Y	XH3719090062	SUCCESS	Success	FRESH	0 B =	
dc1-spine1	spine	dc1-spine1	Juniper_QFX5220-32CD	XC3622260031	SUCCESS	Success	FRESH	0 0 ×	
dc1_single_leaf1	leaf	dc1-single-leaf1	Juniper_QFX5120-48Y	XH3719090097	SUCCESS	Success	FRESH	0 B =	
dc1_esi_001_leaf2	leaf	dc1-esi-001-leaf2	Juniper_QFX5120-48Y	XH3719030120	SUCCESS	📀 Success	FRESH	08:	
dc1_spine2	spine	dc1-spine2	Juniper_QFX5220-32CD	XC3623120013	SUCCESS	Success	FRESH	0 B :	

Figure 40: Blueprint Committed

Apstra Fabric Configuration Verification

After reviewing the changes and committing them to the devices, a functional fabric should be created.

D	 Analytics Bulagrints + must_blueprint_dc1 + Staged > Physical + Build > Resources Dashboard ✓ Analytics ✓ Staged ✓ Uncommitted ((5)) Active ✓ Time Voyager 										
Se	SearchQ										
3 F	ク hysical ガーVi	irtual	⊘ ⊘ Pol	licies 🖤	DCI 🖾 Cat	o talog ⊟ Tasks 🕌	Connectivity Temp	olates	්ම Fabric Setting	2 5	
ol	ogy Nodes Lie	nks	Interfaces	Racks	Pods				Layer Ur	committed Changes	
No	des Q Links										Has Uncommit
ed R	1 Rack ▼ ■ ♥ ● ■ 1-7 of 7 ≪ <										
50	Name #	Tags	Role ‡	External? \$	Deploy Mode ≑	Device Profile \$	Hostname \$	ASN \$	Loopback IPv4 0	Loopback IPvó 🌣	Port Channel ID Range \$
ted	dc1-spine1	÷	Spine	N/A	Deploy	Juniper_QFX5220-32CD	dc1-spine1	64512	192.168.255.0/32	fdf6:ed70:1fac:f2d1::1000/128	n/a
i.	dc1_spine2		Spine	N/A	Deploy	Juniper_QFX5220-32CD	dc1-spine2	64513	192.168.255.1/32	fdf6:ed70:1fac:f2d1::1001/128	n/a
L	dc1_border_leaf1		Leaf	N/A	Deploy	Juniper_QFX5130-32CD	dc1-border-leaf1	64514	192.168.255.2/32	fdf6:ed70:1fac:f2d1::1002/128	n/a
	dc1_border_leaf2		Leaf	N/A	Deploy	Juniper_QFX5130-32CD	dc1-border-leaf2	64515	192.168.255.3/32	fdf6:ed70:1fac:f2d1::1003/128	n/a
	dc1_esi_001_leaf1		Leaf	N/A	Deploy	Juniper_QFX5120-48Y	dc1-esi-001-leaf1	64516	192.168.255.4/32	fdf6:ed70:1fac:f2d1::1004/128	n/a
	dc1_esi_001_leaf2		Leaf	N/A	Deploy	Juniper_QFX5120-48Y	dc1-esi-001-leaf2	64517	192.168.255.5/32	fdf6:ed70:1fac:f2d1::1005/128	n/a
	dc1_single_leaf1		Leaf	N/A	Deploy	Juniper_QFX5120-48Y	dc1-single-leaf1	64518	192.168.255.6/32	fdf6:ed70:1fac:f2d1::1006/128	n/a

Figure 41: Blueprint Nodes Deployed and IPv4 and IPv6 Loopback Assigned by Apstra

The blueprint for the data center should indicate that no anomalies are present to show that everything is working. To view any anomalies with respect to blueprint deployment, navigate to **Blueprint** > *<Blueprint-name>* > Active to view anomalies raised with respect to BGP, cabling, interface down events, routes missing, and so on. For more information, refer to the Apstra User Guide.

A A Discontrate a must be experient and a Anthree Discontrate Charles	
Ymax > Blueprint_ocl > Active > Mission Image: Imag	
Dashboard 🗠 Analytics a Staged 🖆 Uncommitted (D) Active S Time Voyager	
֎ Search	
• • • • • • • • • • • • • • • • • • • •	
😫 Physical 🌮 Virtual 🗇 Policies 🐨 DCI 🔲 Catalog Q, Query 🍄 Anomalies 🌲 Connectivity Templates 🕫 Fabric Settings	
	0
Topology Nodes Links Interfaces Racks Pods Layer Anomalies: All Services	* Selection Status
Q. Nodes Q. Links No Anomalies Anomalies Press	ent
interface states and the states of the state	Anomalies: All Services
Al + Al + Al +	Anomalies: BGP
	o Anomalies: Cabling
Equard Nodes?	0 Anomalies: Config
	Anomalies: Hostname
DC-DC2,8L2 DC-DC2,8L1	An and the second second
mx-router1	o Anomalies: Interface
dc1-spine1 dc1_spine2	Anomalies: LAG
	o Anomalies: Liveness
Act, border, dc1, border, ,, dc1, ed, 00 dc1, ed, 00 dc1, single Jeal1	Anomalies: MLAG
	Anomalies: Probes
DCT-10-02-2811 BigWin-01 E-541_1 DCT-02-B12 BigWin-02 Mia-set interfine1	Anomalies: Route
ESKI1_3 ESKI1_2	22/0/0/2 Deploy Mode
ESXI1_4 bia-1	Deployment Statur: Discover
lxia-2	Deployment status. Discover
Ixia-esi-singlellok 1	0/0/0 Deployment Status: Drain
bria-esi-singlefinite.2	7/0/0 Deployment Status: Service
must-deit-dertuzz	0 Traffic Heat

Figure 42: Blueprint Deployed Shows the Active Tab with No Anomalies

Figure 43: Data Center Blueprint Summary

must_blueprint_dc1 Datacenter	
Physical Structure:	1 pod, 3 racks 2 spines, 5 leaves, 17 generic systems
Virtual Structure:	8 routing zones, 535 virtual networks
Analytics	
Deployment Status	2
Service Anomalies	0
Probe Anomalies	0
Root Causes:	0
Version 2947 Total lines of config 63647	Last modified 19 minutes ago

To verify that the fabric is functional and the changes are configured, log into the console or CLI of each of the spine switches. From the shell of each of the spine switches, enter the following Junos OS CLI command:

show bgp summary | no-more

The output of this command should resemble the output below. It shows that BGP is established from each spine to each of the seven leaf switches for loopback and fabric link IPs.

On Spine 1:

root@dc1-spine1> sh	root@dc1-spine1> show bgp summary no-more								
Warning: License ke	y missing;	requires '	ogp' licen	se					
Threading mode: BGP	9 I/O								
Default eBGP mode:	advertise -	· accept, re	eceive - a	ccept					
Groups: 2 Peers: 14 Down peers: 0									
Table Tot	Paths Act	Paths Supp	ressed I	History D	amp State	Pending			
inet.0									
	49	42	0	0	0	0			
bgp.evpn.0									
	8263	8263	0	0	0	0			
Peer	AS	InPkt	OutPkt	OutQ	Flaps Last	Up/Dwn State #Active/			
Received/Accepted/D	amped								
10.0.1.5	64520	100256	98745	0	12 4w3d	15:44:08 Establ			
inet.0: 2/3/3/0									
10.0.1.7	64518	100736	99371	0	31 4w3d	19:24:11 Establ			
inet.0: 2/3/3/0									
10.0.1.9	64514	17957	17900	0	73 5d 18	:19:16 Establ			
inet.0: 16/17/17/	0								
10.0.1.11	64515	17943	17889	0	34 5d 18	:13:02 Establ			
inet.0: 16/17/17/	0								
10.0.1.13	64516	100735	99370	0	30 4w3d	19:23:45 Establ			
inet.0: 2/3/3/0									
10.0.1.15	64517	100736	99373	0	34 4w3d	19:24:21 Establ			
inet.0: 2/3/3/0									
10.0.1.27	64519	100255	98745	0	18 4w3d	15:44:09 Establ			
inet.0: 2/3/3/0									
192.168.255.2	64514	21707	40706	0	92 5d 18	:18:25 Establ			
bgp.evpn.0: 1149/	1149/1149/0)							
192.168.255.3	64515	18907	43483	0	31 5d 18	:12:36 Establ			
bgp.evpn.0: 1147/	bgp.evpn.0: 1147/1147/1147/0								
192.168.255.4	64516	124001	244758	0	30 4w3d	19:23:43 Establ			

bgp.evpn.0:	1216/1216/1216/0				
192.168.255.5	64517	238893	138433	0	34 4w3d 19:24:20 Establ
bgp.evpn.0:	1216/1216/1216/0				
192.168.255.6	64518	102398	265528	0	31 4w3d 19:23:58 Establ
bgp.evpn.0:	1137/1137/1137/0				
192.168.255.7	64519	101447	217804	0	16 4w3d 15:43:55 Establ
bgp.evpn.0:	1199/1199/1199/0				
192.168.255.8	64520	101419	217814	0	12 4w3d 15:44:00 Establ
bgp.evpn.0:	1199/1199/1199/0				

On Spine 2:

root@dc1-spine2> show bgp summary | no-more Warning: License key missing; requires 'bgp' license Threading mode: BGP I/O Default eBGP mode: advertise - accept, receive - accept Groups: 3 Peers: 14 Down peers: 0 Table Tot Paths Act Paths Suppressed History Damp State Pending inet.0 0 49 42 0 0 0 bgp.evpn.0 8263 8263 0 0 0 0 Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn State|#Active/ Received/Accepted/Damped... 10.0.1.1 64520 100269 98778 0 15 4w3d 15:49:41 Establ inet.0: 2/3/3/0 10.0.1.3 64519 100267 98778 0 20 4w3d 15:49:40 Establ inet.0: 2/3/3/0 10.0.1.17 64518 100749 99375 0 35 4w3d 19:29:47 Establ inet.0: 2/3/3/0 10.0.1.19 64514 17968 17915 92 5d 18:24:04 Establ 0 inet.0: 16/17/17/0 10.0.1.21 64515 17953 17903 0 36 5d 18:17:34 Establ inet.0: 16/17/17/0 10.0.1.23 36 4w3d 19:29:25 Establ 64516 100748 99374 0 inet.0: 2/3/3/0 10.0.1.25 64517 40 4w3d 19:29:58 Establ 100749 99378 0 inet.0: 2/3/3/0 192.168.255.2 64514 93 5d 18:23:41 Establ 21711 40714 0 bgp.evpn.0: 1149/1149/1149/0 28 5d 18:16:29 Establ 192.168.255.3 18902 43498 0 64515 bgp.evpn.0: 1147/1147/1147/0

192.168.255.4	64516	124014	243943	0	35 4w3d 19:29:20 Establ
bgp.evpn.0:	1216/1216/1216/0				
192.168.255.5	64517	238899	137577	0	39 4w3d 19:29:53 Establ
bgp.evpn.0:	1216/1216/1216/0				
192.168.255.6	64518	102416	264691	0	34 4w3d 19:29:44 Establ
bgp.evpn.0:	1137/1137/1137/0				
192.168.255.7	64519	101454	217761	0	21 4w3d 15:49:28 Establ
bgp.evpn.0:	1199/1199/1199/0				
192.168.255.8	64520	101424	217769	0	13 4w3d 15:49:32 Establ
bgp.evpn.0:	1199/1199/1199/0				

If the output of the *show bgp summary | no-more* command resembles the screenshot above, a barebones network fabric is now complete. However, it is not yet ready for production use as the overlay network with VRFs, VLANs, and VNIs still must be applied.

If the output of the *show bgp summary | no-more* command does not resemble the screenshot, it is essential to remedy any configuration errors before proceeding further.

Configure Overlay Network

Configure Routing Zone (VRF) for Red and Blue Tenants, and Specify a Virtual Network Identifier (VNI)

- 1. From Blueprints > Staged -> Virtual > Routing Zones.
- 2. Click Create Routing Zone and provide the following information:
 - a. VRF Name: blue
 - b. VLAN ID: 3
 - c. VNI: 20002
 - d. Routing Policies: Default immutable
- 3. Create another routing zone with the following information:
 - a. VRF Name: red
 - b. VLAN ID:2
 - **c.** VNI: 20001
 - d. Routing Policies: Default immutable

Figure 44: Red and Blue Routing Zone

② Dashboard Analytic	ics 🗳 🕃 Staged 👻 😁 Uncommitted	(D) Active	Dashboard	🗠 Analytic		Staged	B Uncommitted	(i5) Active	3 Time Voyager
Parameters			Parameters						
	VRF Name	blue			VRF Name			red	
	Туре	EVPN			Type			EVPN	
	VLAN ID [®]	3		VLAN	VLAN ID [®]			2	
	VNE	20002			VNI Route Target [®] Junos EVPN IRB Mode [®]			20001	
	Route Target ^Q	20002:1						20001:1 Asymmetric	
	Junos EVPN IRB Mode [®]	Asymmetric							
	DHCP Servers	DHCP Relay not configured			DHCP Serve	rs		DHCP Relay not configured	
Routing Policy			Routing Policy						
	Name	Default_immutable		Name			Default_immutable		
	Description				Description				
	Import Policy [®]	All			Import Polic	y ⁰		Al	
	Extra Import Routes®	Not provided			Extra Import	t Routes®		Not provided	
	Spine Leaf Links [®]	no			Spine Leaf Links [®] Spine Superspine Links [®]			no no yes	
	Spine Superspine Links [®]	no							
	L3 Edge Server Links [®]	yes			L3 Edge Server Links [®]				
	L2 Edge Subnets [®]	yes			L2 Edge Sub	nets [®]		yes	
	Loopbacks [®]	yes			Loopbacks ⁶			yes	
	Static routes [©]	no			Static routes	se		no	
	Extra Export Routes®	Not provided			Extra Export Routes®			Not provided	
	Aggregate Prefixes [®]	Not provided			Aggregate P	vefoxes [®]		Not provided	
	Expect Default IPv4 Route®	yes			Expect Defa	ult IPv4 Route®		yes	
	Expect Default IPv6 Route®	yes			Expect Defa	ult IPv6 Route®		yes	
Route Target Policies			Route Target Policies						
	Import Route Targets	Not provided			Import Rout	e Targets		Not provided	
	Export Route Targets	Not provided			Export Rout	e Targets		Not provided	

Assign EVPN Loopback to Routing

After creating the routing zones, assign the EVPN loopback below to both the Red and Blue routing zones. Navigate to **Blueprint > Staged > Routing Zone** and assign resources from the right-hand side panel.

Resources	Range
MUST-EVPN-Loopbacks-DC1	192.168.11.0/24

Figure : Red and Blue Loopback Assigned

	⊘ 7,	red: Leaf Loopback IPs
	ľ	0
		Pool Name
	×	MUST-EVPN-Loopbacks DC1
	By Re	source Groups
ļ	o 7/	7 EVPN L3 VNIs
Ú	⊘ 7/	7 blue: Leaf Loopback IPs
	ß	0
		Pool Name
	~	MUST-EVPN-Loopbacks DC1

Create Virtual Networks in Red and Blue Routing Zones

Virtual networks should be associated with routing zones (VRF). Create the virtual networks (VNIs) and associate these Virtual Networks with the routing zone (VRF) created earlier. Optionally, create any additional routing zones and virtual networks for production environments based on individual requirements.

Below are the networks created and assigned to appropriate leaf switches in the fabric. The input fields are as follows:

For Blue Network:

- 1. Click Create Virtual Networks.
- 2. Set type of network VXLAN.
- 3. Provide name: dc1_vn1_blue and dc1_vn2_blue.
- 4. Select the **Blue** security zone for both networks:
- 5. Provide VNI:
 - **a.** 12001 for dc1_vn1_blue.
 - **b.** 12002 for dc1_vn2_blue.

- 6. IPv4 Connectivity set enabled.
- 7. Create Connectivity Template for: Tagged.
- 8. Provide IPv4 Subnet and Virtual IP Gateway:
 - a. 10.12.1.0/24, 10.12.1.1 for dc1_vn1_blue
 - **b.** 10.12.2.0/24, 10.12.2.1 for dc1_vn2_blue
- 9. Assign to leaf switches.

For Red Network:

- 1. Click Create Virtual Networks.
- 2. Set type of network VXLAN.
- 3. Provide name: dc1_vn1_red and dc1_vn2_red.
- 4. Select the Red security zone for both networks:
- 5. Provide VNI:
 - a. 11001 for dc1_vn1_red
 - **b.** 11002 for dc1_vn2_red
- 6. IPv4 Connectivity set enabled.
- 7. Create Connectivity Template for: Tagged.
- 8. Provide IPv4 Subnet and Virtual IP Gateway:
 - a. 10.11.1.0/24, 10.11.1.1 for dc1_vn1_red
 - b. 10.11.2.0/24, 10.11.2.1 for dc1_vn2_red
- 9. Assign to leaf switches.

Figure 45: Virtual Networks Created

A > Blueprints	★ > Blueprints > must_blueprint_dc1 > Staged > Virtual > Virtual Networks										
Dashboard	Analytics	Stage	d	[≅] ⊖ Uncommitte	Active	Time Voyager					
Search		0 0						٩			
Physical 🗴	Physical 🎽 Virtual 📀 Policies 🖤 DCI 🔄 Catalog 🗮 Tasks 🏦 Connectivity Templates 👎 Fabric Settings										
tual Networks	Routing Zones Floating IP	s Static Routes	Protocol Sessi	ons Virtual Infr	a Statistics						
									• Create Virt	ual Networks	
Q .5 € selected by ○ all											
Name \$	Routing Zone \$	Type ‡	VN ID \$	L3 MTU \$	Assigned to	IPv4 Connectivity \$	IPv4 Subnet \$	IPv6 Connectivity \$	IPvő Subnet 🕈	Actions	
dc1_vm1_blue	blue	VXLAN	12001		4 modes 4.1.5erder;501,kef.peir1 4c1,mi,002,kef.peir1 4c1,mi,002,kef.peir1 4c1,mi,0c,001,kef.peir1 4c1,kepir,1eef1	Enabled	10.12.1.0/24	Disabled	N/A	C A 8	
dc1_vm1_red	red	VXLAN	11001		• <u>2 nodes</u> dc1_mi_001_leaf_pair1 dc1_mi_ox_001_leaf_pair1	Enabled	10.11.1.0/24	Disabled	N/A	C 🔺 🕯	
dc1_vn2_blue	blue.	VXLAN	12002		4 modes 4-1 binder, 001, binf, pinf1 4-1, binder, 001, binf, pinf1 4-1, bin, 001, binf, pinf1 4-1, bingle, binf1	Enabled	10.12.2.0/24	Disabled	N/A	6 A	
dc1_vn2_red	red	VXLAN	11002		 <u>2 nodes</u> dc1_mi_001_imf_pair1 dc1_mi_m_001_imf_pair1 	Enabled	10.11.2.0/24	Disabled	N/A	8 1	

IRB Network is created, and a connectivity template is added and assigned to leaf switches as shown in Figure 47 on page 56. For more information on connectivity templates, see the Juniper Apstra User Guide.

While creating a virtual network, if the create connectivity template is selected above as tagged, Apstra creates a connectivity template, which is generated automatically for the virtual network.

Navigate to **Blueprint > Staged > Connectivity Templates** to view the templates and assign them to leaf switches. When assigned to leaf switches, a tagged aggregated ethernet interface is created connecting the servers.

Figure 46: Apstra Generated Connectivity Templates

Tagged VxLAN 'dc1_vn1_blue'	Automatically created by AOS at VN creation time	Virtual Network (Single)	Assigned on 1 endpoint(s)	ø	ß	
Tagged VxLAN 'dc1_vn1_red'	Automatically created by AOS at VN creation time	Virtual Network (Single)	Assigned on 1 endpoint(s)	ø	ß	Î
Tagged VxLAN 'dc1_vn2_blue'	Automatically created by AOS at VN creation time	Virtual Network (Single)	Assigned on 1 endpoint(s)	ø	ß	Ĩ
Tagged VxLAN 'dc1_vn2_red'	Automatically created by AOS at VN creation time	Virtual Network (Single)	Assigned on 1 endpoint(s)	Ø	ß	I

Figure 47: Assign Connectivity Template for Each Network to Leaf Switches

Assign Tagged VxLAN 'dc1_vn1_blue'		
Table view		
٩		All bulk actions (\$) will be applied only to the loa
Fabric	Tags	Tagged VxLAN 'dc1_vn1_blue'
* podi (Pod)		0
dc1_border_001 (Rack)		•
 dc1_border_leaf1 (Leaf) 		•
et-0/0/0:0 -> ESXI1_4 (Interface)	rsxt-edge-left NSXT-overlay	
et-0/0/0:3 -> DCI-TO-DC2-BL1 (Interface)		
et-0/0/1:0 -> mx-router1 (Interface)	ext, router	
et-0/0/1:1 -> ESXi1_3 (Interface)		
		•
ae1 -> ESX01_4 (Interface)	NSXT-overlay	
ae2 -> ESXI1_3 (Interface)		

Then, navigate to **Blueprint > Uncommitted** to review the uncommitted changes and commit the overlay configuration. Alternatively, also review the configuration generated for each leaf switch to which the overlay network is created by navigating to **Blueprint > Staged > Physical > Nodes** and check the configuration.

Verify Overlay Connectivity for Blue and Red Network

Having committed changes in the Apstra UI, these changes are now applied to the switches.

To begin verifying the fabric's configuration, log in to the console of each of the leaf switches.

From the CLI of the leaf switches, enter the following commands:

```
!
//begin QFX leaf switch commands//
show interfaces irb terse
show vlans instance evpn-1 vn1101
show vlans instance evpn-1 vn1201
show vlans instance evpn-1 vn1202
!
```

This output displays multiple IRB interfaces and the configured routing instances for the Blue and Red networks.

Red Network IRB on one of the leaf switches:

```
{master:0}
root@dc1-esi-001-leaf1> show interfaces irb terse | match 10.11.*.1/24
```

irb.1101	up	up	inet	10.11.1.1/24
irb.1102	up	up	inet	10.11.2.1/24

Blue Network IRB on one of the leaf switches:

{master:0}					
root@dc1-esi-001-leaf1>	show	inter	faces	irb	terse match 10.12.*.1/24
irb.1201	up	up	inet		10.12.1.1/24
irb.1202	up	up	inet		10.12.2.1/24

Since Apstra now, by default, uses MAC-VRF routing mode, the same can be seen from the below command output for all the Red and Blue network VLANs.

{master:0}			
root@dc1-esi-001-leaf1>	show vlans instance	e evpn-1 vn1101	
Routing instance	VLAN name	Tag	Interfaces
evpn-1	vn1101	1101	
			vtep-15.32772*
			xe-0/0/50:0.0*
{master:0}			
root@dc1-esi-001-leaf1>	show vlans instance	e evpn-1 vn1102	
Routing instance	VLAN name	Tag	Interfaces
evpn-1	vn1102	1102	
			vtep-15.32772*
{master:0}			
root@dc1-esi-001-leaf1>	show vlans instance	e evpn-1 vn1201	
Routing instance	VLAN name	Tag	Interfaces
evpn-1	vn1201	1201	
			ae1.0
			ae3.0*
			et-0/0/52.0*
			et-0/0/53.0*
			vtep-15.32771*
			vtep-15.32772*
			vtep-15.32776*
			vtep-15.32777*
			xe-0/0/50:0.0*
{master:0}			
root@dc1-esi-001-leaf1>	show vlans instance	e evpn-1 vn1202	
Routing instance	VLAN name	Tag	Interfaces
evpn-1	vn1202	1202	

ae2.0*
et-0/0/52.0*
et-0/0/53.0*
vtep-15.32771*
vtep-15.32772*
vtep-15.32776*
vtep-15.32777*

Verify that ERB is Configured on Leaf Switches

Within the CLI of the leaf switches, enter the following commands:

```
!
//begin QFX CLI commands//
show evpn database | match irb.110
show evpn database | match irb.120
!
```

The output of this command displays the distributed gateways on all switches.

The gateways display 10.11.1.1, 10.11.2.1 for the Red network, and 10.12.1.1, 10.12.2.1 for the Blue network. These IRB configurations apply only to devices assigned in the connectivity templates. No other fabric switches have this IRB configured unless assigned through the connectivity template.

<pre>root@dc1-esi-001-leaf1> show evpn database match irb.110</pre>							
00:1c:73:00:00:01	irb.1101	Feb 28 11:33:36	10.11.1.1				
00:1c:73:00:00:01	irb.1102	Feb 28 11:33:36	10.11.2.1				
-leaf1> show evpn	database match irb.120						
00:1c:73:00:00:01	irb.1201	Feb 28 11:33:22	10.12.1.1				
00:1c:73:00:00:01	irb.1202	Feb 28 11:33:22	10.12.2.1				
	-leaf1> show evpn 00:1c:73:00:00:01 00:1c:73:00:00:01 -leaf1> show evpn 00:1c:73:00:00:01 00:1c:73:00:00:01	-leaf1> show evpn database match irb.110 00:1c:73:00:00:01 irb.1101 00:1c:73:00:00:01 irb.1102 -leaf1> show evpn database match irb.120 00:1c:73:00:00:01 irb.1201 00:1c:73:00:00:01 irb.1202	-leaf1> show evpn database match irb.110 20:1c:73:00:00:01 irb.1101 Feb 28 11:33:36 20:1c:73:00:00:01 irb.1102 Feb 28 11:33:36 -leaf1> show evpn database match irb.120 20:1c:73:00:00:01 irb.1201 Feb 28 11:33:22 20:1c:73:00:00:01 irb.1202 Feb 28 11:33:22				

Verify the Leaf Switch Routing Table

Within the CLI of the leaf switches, enter the following commands:

```
!
//begin QFX CLI commands//
show route table red.inet.0 10.11.1.0/24
show route table red.inet.0 10.11.2.0/24
show route table blue.inet.0 10.12.1.0/24
show route table blue.inet.0 10.12.2.0/24
!
```

The output of this command displays the routes for the VRFs Red network for one of the leaf switches.

```
{master:0}
root@dc1-esi-001-leaf1> show route table red.inet.0 10.11.1.0/24
red.inet.0: 1811 destinations, 3372 routes (1811 active, 0 holddown, 0 hidden)
@ = Routing Use Only, # = Forwarding Use Only
+ = Active Route, - = Last Active, * = Both
10.11.1.0/24
                  *[Direct/0] 06:21:33
                    > via irb.1101
                    [EVPN/170] 06:14:27
                    > to 10.0.1.12 via et-0/0/48.0
                       to 10.0.1.22 via et-0/0/49.0
                    [EVPN/170] 00:15:18
                    > to 10.0.1.12 via et-0/0/48.0
                       to 10.0.1.22 via et-0/0/49.0
                    [EVPN/170] 00:17:59
                    > to 10.0.1.12 via et-0/0/48.0
                       to 10.0.1.22 via et-0/0/49.0
10.11.1.1/32
                   *[Local/0] 06:21:33
                       Local via irb.1101
{master:0}
root@dc1-esi-001-leaf1> show route table red.inet.0 10.11.2.0/24
red.inet.0: 3061 destinations, 4622 routes (3061 active, 0 holddown, 0 hidden)
@ = Routing Use Only, # = Forwarding Use Only
+ = Active Route, - = Last Active, * = Both
10.11.2.0/24
                  *[Direct/0] 06:23:38
                   > via irb.1102
                    [EVPN/170] 06:16:32
```

	> to 10.0.1.12 via et-0/0/48.0
	to 10.0.1.22 via et-0/0/49.0
	[EVPN/170] 00:17:43
	> to 10.0.1.12 via et-0/0/48.0
	to 10.0.1.22 via et-0/0/49.0
	[EVPN/170] 00:19:44
	> to 10.0.1.12 via et-0/0/48.0
	to 10.0.1.22 via et-0/0/49.0
10.11.2.1/32	*[Local/0] 06:23:38
	Local via irb.1102

The output of this command displays the routes for the VRFs Blue network for one of the leaf switches.

```
{master:0}
root@dc1-esi-001-leaf1> show route table blue.inet.0 10.12.1.0/24
blue.inet.0: 3087 destinations, 4650 routes (3087 active, 0 holddown, 0 hidden)
@ = Routing Use Only, # = Forwarding Use Only
+ = Active Route, - = Last Active, * = Both
10.12.1.0/24
                  *[Direct/0] 06:26:21
                   > via irb.1201
                    [EVPN/170] 06:26:02
                    > to 10.0.1.12 via et-0/0/48.0
                       to 10.0.1.22 via et-0/0/49.0
                    [EVPN/170] 06:26:04
                    > to 10.0.1.12 via et-0/0/48.0
                       to 10.0.1.22 via et-0/0/49.0
                    [EVPN/170] 06:19:10
                    > to 10.0.1.12 via et-0/0/48.0
                       to 10.0.1.22 via et-0/0/49.0
                    [EVPN/170] 05:58:16
                    > to 10.0.1.12 via et-0/0/48.0
                       to 10.0.1.22 via et-0/0/49.0
                    [EVPN/170] 00:19:51
                    > to 10.0.1.12 via et-0/0/48.0
                       to 10.0.1.22 via et-0/0/49.0
                    [EVPN/170] 00:22:32
                    > to 10.0.1.12 via et-0/0/48.0
                       to 10.0.1.22 via et-0/0/49.0
10.12.1.1/32
                  *[Local/0] 06:26:21
                       Local via irb.1201
{master:0}
```

root@dc1-esi-001-leaf1> show route table blue.inet.0 10.12.2.0/24

```
blue.inet.0: 3087 destinations, 4650 routes (3087 active, 0 holddown, 0 hidden)
@ = Routing Use Only, # = Forwarding Use Only
+ = Active Route, - = Last Active, * = Both
10.12.2.0/24
                  *[Direct/0] 06:26:26
                   > via irb.1202
                   [EVPN/170] 06:26:07
                   > to 10.0.1.12 via et-0/0/48.0
                       to 10.0.1.22 via et-0/0/49.0
                    [EVPN/170] 06:26:09
                   > to 10.0.1.12 via et-0/0/48.0
                      to 10.0.1.22 via et-0/0/49.0
                    [EVPN/170] 06:19:15
                   > to 10.0.1.12 via et-0/0/48.0
                       to 10.0.1.22 via et-0/0/49.0
                    [EVPN/170] 06:20:38
                   > to 10.0.1.12 via et-0/0/48.0
                       to 10.0.1.22 via et-0/0/49.0
                    [EVPN/170] 00:20:16
                   > to 10.0.1.12 via et-0/0/48.0
                       to 10.0.1.22 via et-0/0/49.0
                    [EVPN/170] 00:22:17
                   > to 10.0.1.12 via et-0/0/48.0
                       to 10.0.1.22 via et-0/0/49.0
                  *[Local/0] 06:26:26
10.12.2.1/32
                      Local via irb.1202
```

The following command shows the ESI leaf switches overlay. It shows that the remote leaf VNIs are exchanged between the ESI leaf switches.

{master:0}							
root@dc1-esi-001	-leaf1> show	ethernet-swit	ching vxl	an-tunnel-e	nd-point re	emote	
Logical System N	ame Id	SVTEP-IP	IFL	L3-Idx	SVTEP-Mode	e ELP-SVT	EP-IP
<default></default>	0	192.168.255.	4 lo0.	0 0			
RVTEP-IP	L2-RTT		IFL-Idx	Interface	NH-Id	RVTEP-Mode	ELP-
IP Flags							
192.168.255.5	evpn-1		67108864	2 vtep-15.3	2772 7000	RNVE	
VNID	MC-Group-IP						
11001	0.0.0.0						
11002	0.0.0.0						
12001	0.0.0.0						
12002	0.0.0.0						

{master:0}								
root@dc1-esi-0	01-leaf2> s	show ether	rnet-switchi	ng vxla	n-tunne.	l-end-pc	oint remote	
Logical System	Name	Id SVTE	EP-IP	IFL	L3-Idx	SVTE	P-Mode EL	P-SVTEP-IP
<default></default>		0 192.	.168.255.5	100.0	0			
RVTEP-IP	IFL-Idx	Interfa	ace NH-Id	RVTE	-Mode	ELP-IP	Flags	
192.168.254.2	1388	vtep.32	2771 4989	RNVE				
192.168.255.2	1391	vtep.32	2774 4995	RNVE				
192.168.254.3	1389	vtep.32	2772 4991	RNVE				
192.168.255.3	1390	vtep.32	2773 4994	RNVE				
192.168.255.4	1393	vtep.32	2776 5392	RNVE				
192.168.255.6	1392	vtep.32	2775 4996	RNVE				
192.168.255.7	1381	vtep.32	2777 5811	RNVE				
192.168.255.8	1394	vtep.32	2770 5845	RNVE				
L2-RTT		IFL-Idx	Interface	NH-I	d RVTI	EP-Mode	ELP-IP	Flags
192.168.255.4	evpn-1		67	1088646	vtep-1	5.32776	5392 RNVE	
VNID	MC-Grou	ıp-IP						
12002	0.0.0.0)						
11002	0.0.0.0)						
11001	0.0.0.0)						
12001	0.0.0.0)						

Configure External Router and Inter-VRF Routing

For this JVD, an MX204 router is used as an external router to perform external routing and also for inter-VRF route leaking between the Red and Blue networks. Configuring an external router is similar to adding a generic server. The MX204 router is connected to the border leaf switches, which act as an external gateway to the data center fabric.

To add the MX router as an external router, navigate to Apstra UI, **Blueprint > Staged > Topology**, and click on the border leaf switch to add an external generic system and the connections to the external generic system, as shown in Figure 50 on page 63.

On the following graphic, select the interface for border leaf1 and the MX204 device and its interface and click **Add Link**.



Figure 48: Adding MX204 as External Generic System

Next navigate to **Stage > Policies > Routing Policies** and create an external routing policy to export the route to the external router. This policy is then applied to the connectivity template to allow for exporting Red and Blue network routes as is covered in the next steps.

Figure 49: External Router Policy

Bluepri	nts > must_blue	eprint_dc1 > Stage	d ► Policies ► Ro	uting Policies	▶ external_rou	ter_mx		_	
Dashboard	• <u>~</u>	Analytics	Staged		🔒 Uncommitt	ted	((=)) Active		3 Time Voyager
irch									
nysical	ల సం Virtual	 Policies 	PCI E	Catalog	∏ Tasks	E Conne	ectivity Templat	es 🔊	 Fabric Settings
nts Secu	urity Policies	Interface Policies	Routing Policies	Routing Zor	e Constraints				
to list									
		Name			externa	al_router_m	1		
		Description							
		Import Policy®			Defaul	+			

Name	external_router_mx
Description	
Import Policy®	Default
Extra Import Routes®	Not provided
Spine Leaf Links [®]	no
Spine Superspine Links [®]	no
L3 Edge Server Links®	yes
L2 Edge Subnets®	yes
Loopbacks [®]	yes
Static routes [©]	no
Extra Export Routes®	0.0.0.0/0 GE mask: 1, LE mask: 32, Permit ::/0 GE mask: 1, LE mask: 128, Permit
Aggregate Prefixes [®]	Not provided
Expect Default IPv4 Route [®]	yes
Expect Default IPv6 Route	yes
Associated Routing Zones	No items

Next, navigate to the connectivity template on the blueprint and add the below connectivity template to add IP links, BGP peering, and routing policy with MX204 (external router). In the case of this JVD, the Red and Blue networks are routed towards the MX204, where inter-VRF routing is performed. VLAN 299 is used for the Red network and VLAN 399 for the Blue network.

Figure : IP Links for Red and Blue VRF

Edit Connectivity Template

Parameters	Primitives	User-defined	Pre-defined
 Il_ex_router_ Type: IP Link 	dc1_red_sz		
Routing Zone *			
red			×
Interface Type Tagged Untagged VLAN ID •• 299	•		
L3 MTU 🖲			
IPv4 Addressing None Numbered IPv6 Addressing None Numbered Link local	g Type * g Type *		

Edit Connectivity Template



Figure : BGP Peering to MX for Red and Blue VRF

Edit Connectivity Template

Parameters	Primitives	User-defined	Pre-defined
 bgp_ex_rout: Type: BGP Peering IPv4 A IPv6 A IPv6 A TTL*0 2 OFF Enable Password Keep Alive Time 	er_dc1_red_sz (Generic System) VFI * VFI * P BFD * •		
Hold Time Time	r (sec)		
IPv4 Addressing None Addressed	; Type *		
IPv6 Addressing None Addressed Link local Local ASN • 64497	g Туре [*]		
Neighbor ASN T Static Dynamic Peer From Loopback Interface Peer To Loopback Interface/IP	Type * Endpoint ared IP Endpoint	:	

Edit Connectivity Template

Parameters	Primitives	User-defined	Pre-defined
 bgp_ex_rout Type: BGP Peering 	er_dc1_blue_sz ; (Generic System)		
IPv4 #	AFI *		
	AFI		
TTL *0			
2			
OFF Enable	e BFD 🔭		
Password			
Keep Alive Time	er (sec)		
Hold Time Time	er (sec)		
			÷
IPv4 Addressing	g Type *		
 Addressed 			
IPv6 Addressin	g Type *		
O None			
Addressed			
Local ASN [®]			
64497			
Neighbor ASN	Type *		
 Static 			
 Dynamic 			
Peer From			
 Interface 			
Peer To *®			
Loopback			
Interface/IP	Endpoint		
Interface/St	hared IP Endpoint		

Figure : Routing Policy for Red and Blue VRF

 rp_ex_router_dc1_red_sz Type: Routing Policy Routing Policy 	
external_router_mx	×
 rp_ex_router_dc1_blue_sz Type: Routing Policy 	
Routing Policy	
external_router_mx	×

Then navigate to **Staged > Virtual > Routing Zone**, click on **Red VRF Network**, and scroll below to add IP interface links from both border leaf switches. The same is performed for Blue VRF networks.

Figure 50: Adding IP Interface Links for Red Network

Interfaces 2																			
	··· Filter	selected	ьу 🔾	all 🔵 selected	only	unselecte	d only											1-2 of 2	
					End	point 1		Interface 1					En	dpoint 2		Interface 2			
	0 selected	Routing Zone 0	VLAN ID 0	Name \$	Role ©	Interface \$	L3 MTU	IPv4 Address 0	IPv4 Address Type ©	IPv6 Address 🗢	IPv6 Address Type ©	Name 0	Role \$	Interface 0	L3 MTU	IPv4 Address 0	IPv4 Address Type 0	IPv6 Address \$	IPv6 Addre Type 0
		red	299	dc1_border_leaf1	Leaf	et- 0/0/1:0.299	Not provided	10.200.0.4/31	Numbered	2001:db8:dc1:10:200::4/127	Numbered	mx- router1	Generic System	n/a	Not provided	10.200.0.5/31	Numbered	2001:db8:dc1:10:200::5/127	Number
		red	299	dc1_border_leaf2	Leaf	et- 0/0/1:0.299	Not provided	10.200.0.6/31	Numbered	2001:db8:dc1:10:200::6/127	Numbered	mx- router1	Generic System	n/a	Not provided	10.200.0.7/31	Numbered	2001:db8:dc1:10:200::7/127	Number

Figure 51: Adding IP Interface Links for Blue Network

Interfaces 2																			
	 Filter	selected	by 🔾	all 🔵 selected	only	unselecte	d only											1-2 of 2	
	Endpoint 1 Interface 1 Endpoint 2 Interf												Interface 2	erface 2					
	0 selected	Routing Zone ‡	VLAN ID ‡	Name 0	Role ©	Interface 0	L3 MTU	IPv4 Address 0	IPv4 Address Type ©	IPvő Address 0	IPv6 Address Type \$	Name ¢	Role 0	Interface ©	L3 MTU	IPv4 Address 0	IPv4 Address Type ©	IPvő Address 0	IPv6 Adi Type ≎
		blue	399	dc1_border_leaf2	Leaf	et- 0/0/1:0.399	Not provided	10.200.0.10/31	Numbered	2001:db8:dc1:10:200::a/127	Numbered	mx- router1	Generic System	n/a	Not provided	10.200.0.11/31	Numbered	2001:db8:dc1:10:200:b/127	Numb
		blue	399	dc1_border_leaf1	Leaf	et- 0/0/1:0.399	Not provided	10.200.0.8/31	Numbered	2001:db8:dc1:10:200::8/127	Numbered	mx- router1	Generic System	n/a	Not provided	10.200.0.9/31	Numbered	2001:db8:dc1:10:200::9/127	Numb

Commit the blueprint to push configs to the two border leaf switches. Note that the external router needs to be configured manually, as Apstra does not manage the MX204. For the configuration MX204 router, the interfaces are configured using the IPs used above in Figure 50 on page 69 and Figure 56 on page 69.

MX204 configuration snippet for the Red and Blue networks:

```
xe-0/0/2:0 {
    vlan-tagging;
    unit 0 {
        vlan-id 0;
        family inet;
    }
    unit 299 {
        vlan-id 299;
        family inet {
            address 10.200.0.5/31;
        }
        family inet6 {
            address 2001:db8:dc1:10:200::5/127;
        }
    }
    unit 399 {
        vlan-id 399;
        family inet {
            address 10.200.0.9/31;
        }
        family inet6 {
            address 2001:db8:dc1:10:200::9/127;
        }
    }
}
xe-0/0/2:1 {
    vlan-tagging;
    unit 0 {
        vlan-id 0;
        family inet;
    }
    unit 299 {
        vlan-id 299;
        family inet {
            address 10.200.0.7/31;
        }
        family inet6 {
            address 2001:db8:dc1:10:200::7/127;
        }
    }
```
```
unit 399 {
    vlan-id 399;
    family inet {
        address 10.200.0.11/31;
    }
    family inet6 {
        address 2001:db8:dc1:10:200::b/127;
    }
  }
}
```

For inter-VRF routing, a policy is configured on the MX as below to enable inter-VRF routing between the Red and Blue VRF networks. Both VRFs are configured on the border leaf switches to BGP peer with the MX204 (external router). The MX204 uses a BGP routing policy to exchange inter-VRF routes.

NOTE: Apstra can also configure inter-VRF routing between the Red and Blue networks without needing an external router. Refer to the Apstra guide for more information. It is recommended that any changes made to any settings be thoroughly tested. For this JVD, the "Route Target Overlaps Allow internal route-target policies" setting was not used. If this setting is set to 'No Warning', then each of the routing zones, such as Red and Blue, can be changed to allow for route target exchange using import and export route target policies within Apstra.

MX204 configuration snippet for inter-VRF:

```
root@must-mx204-1> show configuration policy-options policy-statement RoutesToFabric
term 1 {
    from interface lo0.0;
    then accept;
}
term 2 {
    from {
        protocol [ static bgp ];
        route-filter 0.0.0.0/0 exact;
    }
    then accept;
}
term 3 {
    from {
        protocol [ static bgp ];
        rib inet6.0;
        route-filter::/0 exact;
```

72

```
}
    then accept;
}
term 4 {
    then reject;
}
root@must-mx204-1> show configuration protocols bgp
group fabric {
    type external;
    multihop {
        ttl 1;
    }
    multipath {
        multiple-as;
    }
    neighbor 10.200.0.4 {
        export RoutesToFabric;
        peer-as 64514;
    }
    neighbor 2001:db8:dc1:10:200::4 {
        export RoutesToFabric;
        peer-as 64514;
    }
    neighbor 10.200.0.8 {
        export RoutesToFabric;
        peer-as 64514;
    }
    neighbor 2001:db8:dc1:10:200::8 {
        export RoutesToFabric;
        peer-as 64514;
    }
    neighbor 10.200.0.6 {
        export RoutesToFabric;
        peer-as 64515;
    }
    neighbor 2001:db8:dc1:10:200::6 {
        export RoutesToFabric;
        peer-as 64515;
    }
    neighbor 10.200.0.10 {
        export RoutesToFabric;
        peer-as 64515;
```

}

```
neighbor 2001:db8:dc1:10:200::a {
    export RoutesToFabric;
    peer-as 64515;
}
```

Apstra UI: Blueprint Dashboard, Analytics, probes, Anomalies

The managed switches generate vast amounts of data about switch health and network health. To analyze these with respect to the data center network, Apstra uses Intent-Based Analytics that combines the intent from the graph¹ with switch-generated data to provide the data center network view using the Apstra Dashboard.

NOTE: Apstra uses a graph model to represent data center infrastructure, policies, and so on. All information about the network is modeled as nodes and relationships between them. The graph model can be queried for data and used for analysis and automation. For more information on Apstra graph model and queries refer to the Apstra user Guide.

Analytics Dashboard, Anomalies, Probes and Reports

Apstra also provides predefined dashboards that collect data from devices. With the help of IBA probes, Apstra combines intent with data to provide real-time insight into the network, which can be inspected using Apstra GUI or Rest API. The IBA probes can be configured to raise anomalies based on the thresholds. It recommended to analyze the amount of data generated by probes to ensure the disk space of Apstra server is able accommodate IBA operation. By adjusting the log rotation setting, the disk usage can be reduced.

Apstra allows the creation of custom dashboards; refer to the Apstra User Guide for more information. From the blueprint, navigate to **Analytics > Dashboards** to view the analytics dashboard.

Figure 52: Analytics Dashboard

☆ 希 → Blueprints → must_blueprint_dc1 → Analytics → Dashboards	0 0	-
Dashboard Analytics Staged Chrommit	ted ((D)) Active D Time Voyager	
	Root Causes <u>lat</u> Flow Data	
		Configure Auto-Enabled Dashboards
Display mode *** Expanded * NO ANOMALIES		1-9 of 9 < >
Device Environmental Health Summary 🔹 System 7 months ago Default 💽		2 8 1
The dashboard shows device environmental data		
Systems missing power supplies	Systems missing fans	Switch temperature alarm
Value®	Value®	No anomalies!
		View stage
o view stage	view stage	Power supply temperature alarm
Systems with inoperative power supplies	Systems with inoperative fans	No anomalies!
No anomalies!	No anomalies!	View stage
View stage	View stage	
Systems with faulty power supply fans	Airflow direction mismatch	
No anomalies!	No anomalies!	
View stage	View stage	

The analytics dashboard displays the status of all device health statuses. In case of anomalies, click on the anomalies tab to view anomalies. The blueprint anomalies tab displays a "No Anomalies!" message in case no anomalies are detected by the IBA probes. For more information, refer to the Apstra User Guide.

Figure 53: Blueprint Anomalies

☆ 谷 → Blueprints → must_blueprint_	lc1 → Analytics → Anomalies				
Dashboard Analytic	ics 🖹 Staged	≝ _⊖ Uncommitted ((6	B)) Active	3 Time Voyager	
Dashboards Anomalies	15 Widgets @ Probes	Reports Root Causes	년 Flow Data		
No anomalies!					

To view the probes configured, navigate to **Blueprint > Analytics > Probes**. Here, actions can be performed to edit, clone, or delete probes. For instance, if a probe anomaly needs to be suppressed, the same can be performed by editing the probe.

Figure 54: Apstra Predefined Probes

습 🏠	> Blueprints + must_blueprint_dc1 + Analytics + Probes						1
(1)	Dashboard 🖉 Analytics 🝙 Staged 🚔 Uncommitted	((B)) Active	Time Voyager				
@ Da	shboards 🔅 Anomalies 🗵 Widgets 🖤 Probes 💿 Reports 🔅 Ro	ot Causes 네 Flow Data					
							Create Probe -
	9						1-24 of 24
0 selected	Name 🔺	Anomalies \$	State \$	Updated By \$	Tags ≎	Enabled \$	Actions
	BGP Monitoring	No anomalies	Operational	🚨 admin a year ago			8 6 1
	Device Environmental Checks	No anomalies	Operational	2 bhakti 5 months ago			C 6 1
	Device System Health	No anomalies	Operational	🙎 bhakti a month ago			2 8 1
	Device Telemetry Health	No anomalies	III Disabled	2 bhakti 3 months ago		099	2.6
	Device Traffic	No anomalies	Operational	System a year ago			2.6
	Drain Traffic Anomaly	No anomalies	Operational	System 6 months ago			C 8 1
	ECMP Imbalance (Fabric Interfaces)	No anomalies	Operational	Sausalito 4 months ago			C 6 1
	ESI Imbalance	No anomalies	Operational	🚨 admin a year ago			C 6 1
	ESI Imbalance 🕋	O No anomalies	Operational	🚨 admin a year ago			C 6 1
	EVPN Host Flapping	No anomalies	Operational	🙎 admin a year ago			C 6 1
	Hypervisor & Fabric LAG Config Mismatch	No anomalies	III Disabled	🚨 admin a year ago		CEL	6 6 1
	Hypervisor & Fabric LAG Config Mismatch_MUST	No anomalies	III Disabled	🚨 admin a year ago			C 6 1
	Hypervisor & Fabric VLAN Config Mismatch	No anomalies	II Disabled	🚨 admin a year ago		Out	8 8 T

To raise or suppress an anomaly, mark or unmark the **Raise Anomaly** check box.

Figure : Configure Probe Anomaly

ú	? A → Blueprints → must_blueprint_dc1 → Analytics → Probes		n	
	Dashboard Analytics Staged		Uncommitted ((D) Active 'S Time Voyager	
Des	cription		Enabled	
Т	his probe shows BGP session statuses for all switches and raises anomalie	s for flapping	BGP sessions. Cor Disabled probes don't produce data and don't raise anomalies.	
		>	Processor: Sustained BGP Session Flapping Time in State	
	BGP Session		Inputs	
	BGP Session	99	Input Stage Input Mame Stage Name	Column Name
Ļ	BGP Session Flapping		in • BGP Session Flapping *	value
	BGP Session Flapping			
Ļ	Sustained BGP Session Flapping		Properties	
	Sustained BGP Session Flapping	84	Graph Query	
	Add Processor		Add Cape Query Add Cape Query Concernment queries on the graph for probe parametrization. Results of the queries can be accessed using the "query_result" variable with the appropriate index. For example, if querying prope Time Window S Minutes How long to monitor state. State Range* ""true"	ty set nodes under m
			More than or equal to - 120	
			Map state value to its allowed time range in seconds.	
			Raise Anomaly Whether to raise an anomaly Anomaly Metric Logging Erable metric logging for anomalies	
			Anomaly MetricLog Retention Duration	
			1 Day Retain anomaly metric data in MetricDb for specified time period Anomaly MetricLog Retention Size	
			1073741824	
			Maximum allowed size, in bytes of anomaly metric data to store in MetricDB Enable Streaming	

To generate reports, navigate to **Blueprints > Analytics > Reports**. Here, reports can be downloaded to analyze health, device traffic, and so on.

Figure 55: Generate Health Report

☆ 希 → Blueprints → must_blueprint_dc1 → Ana ② Dashboard Analytics	ytics · Reports		-
Dashboards Anomalies Io Wid	gets II Probes D Reports D Root Causes Id Pow Data		
Reports are accessible when: All associated probes have been instantiat Run time data has been correctly populate Historical data has been stored on disk via	nd d in the probe stages Metric Logging		
Name	Description	Required Predefined Probes	Actions
Device Health	Analyze device health	Device System Health ⊘ Device Telemetry Health ⊗	۲
Optical XCVR	Analyze spitical transceivers telemetry patterns and trends	 Optical Transcrivers ⊚ Device Traffic ⊘ 	۲
Traffic	Analyze device traffic patterns and trends	 Device Traffic ∅ Device System Health ∅ 	۲

Root Cause Identification (RCI) is a technology integrated into Apstra software that automatically determines the root causes of complex network issues. RCI leverages the Apstra datastore for real-time network status and automatically correlates telemetry with each active blueprint intent. Root cause use cases include, for instance, link down, link miscabled, Interface down, link disconnect, and so on.

Figure 56: Enable Root Cause Analysis

Dashboard		Malytics	Staged	Uncon	imitted ((色))	Active	3 Time Voyage
@ Dashboards	1. 公 Anomali	es 🔯 Widgets	(Probes	Reports	 Root Causes 	@ Flow Data	
				01			
				2.*		Enable Root 0	Cause Analysis
•				2.*	3.	Enable Root 0	Cause Analysis
-				2.*	3.	C Enable Root (Cause Anal

Figure 57: Root Cause Enabled for Connectivity

☆ 😚 > Blueprints > must_blueprint_dc1 > Analytics > Root Causes > connectivity					
 Dashboard Analytics 	Staged Encommitted	(四) Active	Time Voyager		
④ Dashboards 🏠 Anomalies 🗉 Widgets	(I) Probes	Causes @ Flow Data			
Back to list					
Configuration					
	Model Name	connecti	vity		
	State	OPERAT	TIONAL		
	Trigger Period	30s			
	Config Updated	a few sec	conds ago		
	States Updated	a few sec	conds ago		
Root Causes					
	No Root C	auses Found			

Validation Framework

IN THIS SECTION

- Test Bed | 79
- VRF Characteristics: | 79
- Platforms / Devices Under Test (DUT) | 81
- Test Bed Configuration | 81

Extensive testing of best practice architectures is key to the Juniper Validated Design (JVD) program. JVDs qualify and quantify these best practice architectures, allowing you to know exactly what you're buying and to spend your time deploying and managing your network instead of designing it.

JVDs employ a layered testing approach to deliver reliability and repeatability. Individual features receive functional testing. Multifunction testing builds on this functional testing to see if multiple features work together. Product delivery testing builds upon multifunctional testing to validate that these features combined perform as expected for tested use cases, and JVD testing builds upon product delivery testing by testing multiple products together (including third-party integrations where appropriate) to ensure that all these products combined make an industry-leading solution.

Figure 58: Validation Framework



Testing with real-world applications and traffic provides more accurate data regarding performance and response to different configurations. The standardized nature of JVDs ensures the same network

architecture is deployed in multiple testing environments, and the use of JVDs by multiple customers allows for any lessons learned in production deployments to rapidly benefit all JVD customers. The more JVDs that are deployed worldwide, the greater the value they provide to all.

Test Bed

The test bed environment consists of a 3-stage EVPN/VXLAN fabric managed by Juniper Apstra, with four ESI server leaf switches configured as two redundant pairs, one single (non-redundant) server leaf (non-ESI), and two redundant border leaf switches connected to two spines. An external router is also connected to the border leaf switches. A traffic generator is connected to the test ports on the external router and the ESXi servers.

To ensure all the platforms specified in the Supported Devices and Positioning Table 1 on page 7 are validated, two data center topologies connected using DCI were used. Since there were multiple devices for each role, the devices were swapped, and the tests were repeated with each combination. For instance, border leaf switches were swapped with QFX5130-32CD, PTX10001-36MR, ACX7100-32CD, and so on.

Figure 59: 3-Stage Lab Topology



VRF Characteristics:

RED VRF

- VLANs 400-649 with IRB v4/v6:
 - on DC1-SNGL-LEAF1 single access port
 - on DC1-ESI-LEAF1 single access port, AE1 and AE2
 - on DC1-ESI1-LEAF2 single access port, AE1 and AE2
 - on DC1-BRDR-LEAF1 to distribute routes to external-router
 - on DC1-BRDR-LEAF2 to distribute routes to external-router
- VLANs 400-649 on each test port with 10 unique MAC/IP per VLAN
- DHCP client on TP3
- External DHCP server on TP17

Blue VRF

- VLANs 3500-3749 with IRB v4/v6:
 - on DC1-SNGL-LEAF1 single access port
 - on DC1-ESI-LEAF1 single access port, AE1 and AE2
 - on DC1-ESI1-LEAF2 single access port, AE1 and AE2
 - on DC1-BRDR-LEAF1 to distribute routes to external-router
 - on DC1-BRDR-LEAF2 to distribute routes to external-router
- VLANs 3500-3749 on each test port with 10 unique MAC/IP per VLAN
- DHCP client on TP3, TP4, TP5
- External DHCP server on TP2

Platforms / Devices Under Test (DUT)

Table 5: Devices Under Test (Validated Devices)

Devices Under Test (Validated Devices)						
Solution	Server Leaf Switches	Border Leaf Switches	Spine			
3-stage EVPN/VXLAN (ERB)	QFX5120-48Y-8C	QFX5130-32CD	QFX5220-32CD			
	QFX5110-48S	QFX5700	QFX5120-32C			
	EX4400-24MP	ACX7100-48L				
		ACX7100-32C				
		PTX10001-36MR				
		QFX10002-36Q				

Test Bed Configuration

Contact your Juniper Networks representative to obtain the full archive of the test bed configuration used for this JVD.

Test Objectives

IN THIS SECTION

- Test Goals | 82
- Test Non-Goals | 82

The primary objective of this JVD testing is the qualification testing of the 3-stage fabric with Juniper Apstra. The design is based on an ERB (Type 2 and Type 5) EVPN/VXLAN fabric with the spine, server leaf, and border leaf switches. The goal is to ensure the design is well-documented and will produce a reliable, predictable deployment for the customer. The qualification objectives include validation of blueprint deployment, device upgrade, incremental configuration pushes/provisioning, Telemetry/ Analytics checking, failure mode analysis, and verification of host traffic.

Test Goals

The 3-Stage Fabric with Juniper Apstra JVD testing uses the following flow:

- Initial design and blueprint deployment through Apstra
- Validation of fabric operation and monitoring through Apstra analytics and telemetry dashboard
- Scale testing
- Validation of end-to-end traffic flow
- System health, ARP, ND, MAC, BGP (route, next hop), interface traffic counters, and so on
- Test for anomalies
- In order to pass validation, the 3-stage fabric with Juniper Apstra must also pass the following scenarios:
 - Node Reboot simulated real-world switch outage.
 - Field scenarios like interface down/up and Laser on/off impact to the fabric and check anomalies reporting in Apstra.
 - Traffic recovery was validated after all failure scenarios.

Refer to the test report for more information.

Test Non-Goals

Test non-goals for this JVD were to test the following switches are tested for non-baseline use in the following roles:

QFX10002-36Q as a border leaf

Other features tested:

- DCI Interconnectivity between data centers
- Interoperability with NSX-T Edge Gateway
- Host connectivity between fabric-connected hosts created by Apstra towards NSX-managed hosts

Results Summary and Analysis

For the 3-stage JVD, comprehensive functional testing was performed on devices listed in Devices Under Test (Validated Devices) Table 5 on page 81 to validate the Junos OS Release 23.4R2-S3 and Apstra 4.2.1:

- Baseline System Test:
 - Enabling devices for Apstra, applying pristine configuration, and designing logical devices and interface maps.
 - Apstra Provisioning of the entire 3-stage using the Data Center Reference Architecture feature of Apstra, involving racks, Templates, and blueprints, assigning interface maps and resources to switches, and cabling switches.
 - Modifying Apstra blueprints to swap border leaf switches during testing.
 - Apstra commits to deploy configurations to devices.
 - Provisioning virtual networks and routing zones, assigning EVPN loopbacks for VRFs, and IRB interfaces through Apstra.
- Operational and Trigger Tests:
 - Operational testing of switches was carried out for the following:
 - Device upgrade to 23.4R2-S3 release
 - Reboot devices cause no issues when the devices boot up
 - Process Restarts—I2ald, interface-control, rpd—aim to minimize packet loss and fully restore the control and data plane.
 - Move 4 MAC hosts from one port to another without connectivity issues.
 - BFD failover tests by deactivating BGP on the leaf switches with ESI configured to allow for traffic convergence.
 - Reset DHCP Bindings to ensure fabric forwards the DHCP requests and address assignment should be released and reassigned.

- Extended negative tests (process restart, deactivate BGP, link failures) in an 8-hour cycle to ensure switches restore to baseline state and resume normal traffic forwarding.
- Connectivity tests for the following items were carried out:
 - Service leaf link failure.
 - Multihomed link failure.
 - Leaf-to-spine link failure.
- Resiliency tests for overlay connectivity testing for the below scenarios:
 - Intra-VLAN.
 - Inter-VLAN to every host.
 - Traffic to external route.
 - DHCP client/server flows.

Scale Testing numbers are as follows:

Table 6: Multi-dimensiona	I Scale Number	s Tested
---------------------------	----------------	----------

Features	Tested Scale Numbers	Tested Scale Numbers with EX4400* ESI Leaf Pair
VLANs	500	500
V4 host entries (MAC-IP)	35500	17500
V6 host entries (NDP)	1400	1400
VNI	500	500
VTEP	6	6
ESI	4	4
IRB	500	500
BGP Routing Table	343000	148900
EVPN Table	35500	17500

The scale numbers above are not device maximums; they only reference the scale at which these multidimensional test cases are performed.

NOTE: The maximum VLANs per aggregated Ethernet (AE) interface is 2,000 on the QFX5120 and 1,000 on the EX. Attempting to define more VLANs than this on these platforms will cause a commit warning of too many VLAN IDs on an untagged interface.

Overall, the JVD validation testing didn't detect any issues, and all performance parameters were within the threshold and performed as expected. Traffic profiles tested on all server leaf switches for intra-VRF, inter-VRF, and external routes were 1000 pps with a random packet size of 256-1024 bytes.

Recommendations

The 3-Stage EVPN/VXLAN Fabric with Juniper Apstra JVD follows an industry-standard ERB design. It simplifies the data center provisioning process. Not only does it help in managing the data center for Day-0 and Day-1 operations, but it also simplifies Day-2 operations by enabling customers to upgrade devices, manage devices, and monitor device telemetry. As an inherently multi-vendor management platform, Apstra also provides customers the ability to choose vendors, something that is especially valuable today, as data center technology is evolving rapidly with the advent of AI technology.

Junos OS Release 23.4R2-S3 is the minimum recommended software version for this JVD.

The Juniper hardware listed in the Devices Under Test (Validated Devices) Table 5 on page 81 are the best-suited switch platforms in terms of features, performance, and the roles that are specified in this JVD.

Tested Optics

Table 7: Optics used during testing

Part number	Optics Name	Device Role	Device Model
740-032986	QSFP+-40G-SR4	External Router	MX204
740-070749	JPSU-650W-AC-AO	External Router	MX204

740-061405	QSFP-100GBASE-SR4	Spine	QFX5220-32CD
740-065630	QSFP28-100G-AOC-1M	Spine	QFX5220-32CD
740-065631	QSFP28-100G-AOC-3M	Spine	QFX5220-32CD
740-032986	QSFP+-40G-SR4	Spine	QFX5220-32CD
740-065632	QSFP28-100G-AOC-5M	Spine	QFX5220-32CD
740-065463	SFP+-10G-AOC3M	Server Leaf	QFX5120-48Y-8C
740-021308	SFP+-10G-SR	Server Leaf	QFX5120-48Y-8C
740-038624	QSFP+-40G-CU3M	Server Leaf	QFX5120-48Y-8C
740-061405	QSFP-100GBASE-SR4	Server Leaf	QFX5120-48Y-8C
740-031980	SFP+-10G-SR	Server Leaf	QFX5120-48Y-8C
740-054053	QSFP+-4X10G-SR	Server Leaf	QFX5120-48Y-8C
740-032986	QSFP+-40G-SR4	Server Leaf	QFX5120-48Y-8C
650-114386	2x100G QSFP28	Server Leaf	EX4400-24MP
740-065631	QSFP28-100G-AOC-3M	Server Leaf	EX4400-24MP
740-021308	SFP+-10G-SR	Server Leaf	EX4400-24MP
740-031980	SFP+-10G-SR	Server Leaf	EX4400-24MP
740-065630	QSFP28-100G-AOC-1M	Border Leaf	QFX5130-32CD
740-032986	QSFP+-40G-SR4	Border Leaf	QFX5130-32CD
740-065632	QSFP28-100G-AOC-5M	Border Leaf	QFX5130-32CD
740-032986	QSFP+-40G-SR4	Border Leaf	QFX5700

740-065631	QSFP28-100G-AOC-3M	Border Leaf	QFX5700
740-021308	SFP+-10G-SR	Border Leaf	QFX5700
740-061405	QSFP-100GBASE-SR4	Border Leaf	QFX5700
740-031980	SFP+-10G-SR	Border Leaf	QFX5700
740-030658	SFP+-10G-USR	Border Leaf	QFX5700
740-021308	SFP+-10G-SR	Border Leaf	ACX7100-48L
740-030658	SFP+-10G-USR	Border Leaf	ACX7100-48L
740-031980	SFP+-10G-SR	Border Leaf	ACX7100-48L
740-065463	SFP+-10G-AOC3M	Border Leaf	ACX7100-48L
740-030076	SFP+-10G-CU1M	Border Leaf	ACX7100-48L
740-061405	QSFP-100GBASE-SR4	Border Leaf	ACX7100-48L
740-065631	QSFP28-100G-AOC-3M	Border Leaf	ACX7100-48L
740-065630	QSFP28-100G-AOC-1M	Border Leaf	ACX7100-48L
740-058734	QSFP-100GBASE-SR4	Border Leaf	ACX7100-32C
740-061405	QSFP-100GBASE-SR4	Border Leaf	ACX7100-32C
740-032986	QSFP+-40G-SR4	Border Leaf	ACX7100-32C
740-065632	QSFP28-100G-AOC-5M	Border Leaf	ACX7100-32C
740-065630	QSFP28-100G-AOC-1M	Border Leaf	PTX10001-36MR
740-061405	QSFP-100GBASE-SR4-T2	Border Leaf	PTX10001-36MR
740-032986	QSFP+-40G-SR4	Border Leaf	PTX10001-36MR

Revision History

Table 53: Revision History

Date	Version	Description
December 2024	JVD- DCFABRIC-3STAGE -02-01	Recommended Junos version updated to 23.4R2-S3 from 22.2R3-S3

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners. Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice. Copyright © 2025 Juniper Networks, Inc. All rights reserved.