Extreme Fabric Automation, 2.2.0

Administration Guide

9036659-00 Rev AD
September 2020

Legal Notice

Extreme Networks, Inc. reserves the right to make changes in specifications and other information contained in this document and its website without prior notice. The reader should in all cases consult representatives of Extreme Networks to determine whether any such changes have been made.

The hardware, firmware, software or any specifications described or referred to in this document are subject to change without notice.

Trademarks

Extreme Networks and the Extreme Networks logo are trademarks or registered trademarks of Extreme Networks, Inc. in the United States and/or other countries.

All other names (including any product names) mentioned in this document are the property of their respective owners and may be trademarks or registered trademarks of their respective companies/owners.

For additional information on Extreme Networks trademarks, see: www.extremenetworks.com/company/legal/trademarks

Open Source Declarations

Some software files have been licensed under certain open source or third-party licenses. End-user license agreements and open source declarations can be found at: https://www.extremenetworks.com/support/policies/open-source-declaration/

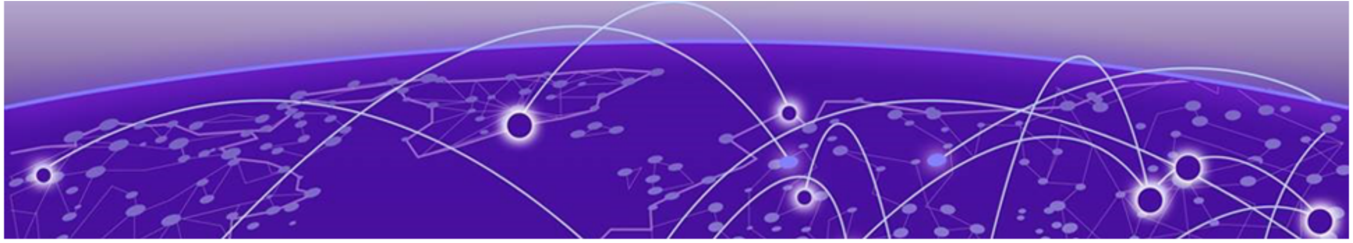
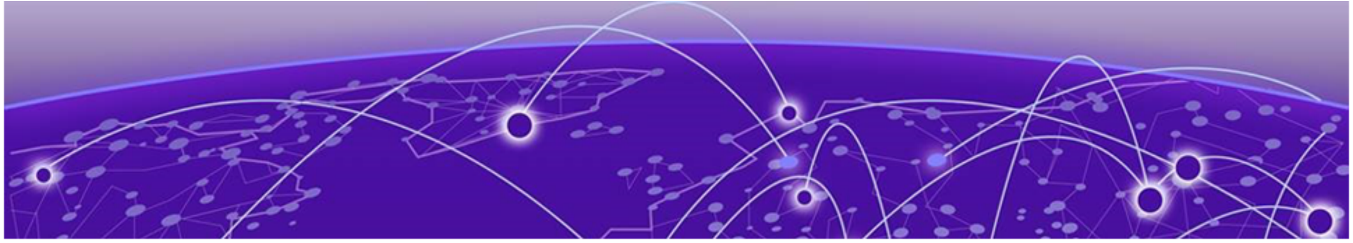


Table of Contents



Preface

This section describes the text conventions used in this document, where you can find additional information, and how you can provide feedback to us.

Text Conventions

Unless otherwise noted, information in this document applies to all supported environments for the products in question. Exceptions, like command keywords associated with a specific software version, are identified in the text.

When a feature, function, or operation pertains to a specific hardware product, the product name is used. When features, functions, and operations are the same across an entire product family, such as ExtremeSwitching switches or SLX routers, the product is referred to as *the switch* or *the router*.

Table 1: Notes and warnings

Icon	Notice type	Alerts you to...
💡	Tip	Helpful tips and notices for using the product.
🗒	Note	Useful information or instructions.
➡	Important	Important features or instructions.

Table 1: Notes and warnings (continued)

Icon	Notice type	Alerts you to...
⚠	Caution	Risk of personal injury, system damage, or loss of data.
⚠	Warning	Risk of severe personal injury.

Table 2: Text

Convention	Description
`screen displays`	This typeface indicates command syntax, or represents information as it appears on the screen.
The words *enter* and *type*	When you see the word *enter* in this guide, you must type something, and then press the Return or Enter key. Do not press the Return or Enter key when an instruction simply says *type*.
Key names	Key names are written in boldface, for example **Ctrl** or **Esc**. If you must press two or more keys simultaneously, the key names are linked with a plus sign (+). Example: Press **Ctrl**+**Alt**+**Del**
Words in italicized type	Italics emphasize a point or denote new terms at the place where they are defined in the text. Italics are also used when referring to publication titles.
NEW!	New information. In a PDF, this is searchable text.

Table 3: Command syntax

Convention	Description
bold text	Bold text indicates command names, keywords, and command options.
italic text	Italic text indicates variable content.
[ ]	Syntax components displayed within square brackets are optional. Default responses to system prompts are enclosed in square brackets.
{ **x** | **y** | **z** }	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options.
x | **y**	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, such as passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, *member*[*member*...].
\	In command examples, the backslash indicates a “soft” line break. When a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Documentation and Training

Find Extreme Networks product information at the following locations:

Current Product Documentation

Release Notes

Hardware and software compatibility for Extreme Networks products

Extreme Optics Compatibility

Other resources such as white papers, data sheets, and case studies

Extreme Networks offers product training courses, both online and in person, as well as specialized certifications. For details, visit www.extremenetworks.com/education/.

Getting Help

If you require assistance, contact Extreme Networks using one of the following methods:

Extreme Portal

Search the GTAC (Global Technical Assistance Center) knowledge base; manage support cases and service contracts; download software; and obtain product licensing, training, and certifications.

The Hub

A forum for Extreme Networks customers to connect with one another, answer questions, and share ideas and feedback. This community is monitored by Extreme Networks employees, but is not intended to replace specific guidance from GTAC.

Call GTAC

For immediate support: (800) 998 2408 (toll-free in U.S. and Canada) or 1 (408) 579 2826. For the support phone number in your country, visit: www.extremenetworks.com/support/contact

Before contacting Extreme Networks for technical support, have the following information ready:

- Your Extreme Networks service contract number, or serial numbers for all involved Extreme Networks products
- A description of the failure
- A description of any actions already taken to resolve the problem
- A description of your network environment (such as layout, cable type, other relevant environmental information)
- Network load at the time of trouble (if known)
- The device history (for example, if you have returned the device before, or if this is a recurring problem)
- Any related RMA (Return Material Authorization) numbers

Subscribe to Service Notifications

You can subscribe to email notifications for product and software release announcements, Vulnerability Notices, and Service Notifications.

1. Go to www.extremenetworks.com/support/service-notification-form.
2. Complete the form (all fields are required).

3. Select the products for which you would like to receive notifications.

   Note
   You can modify your product selections or unsubscribe at any time.

4. Select **Submit**.

Providing Feedback

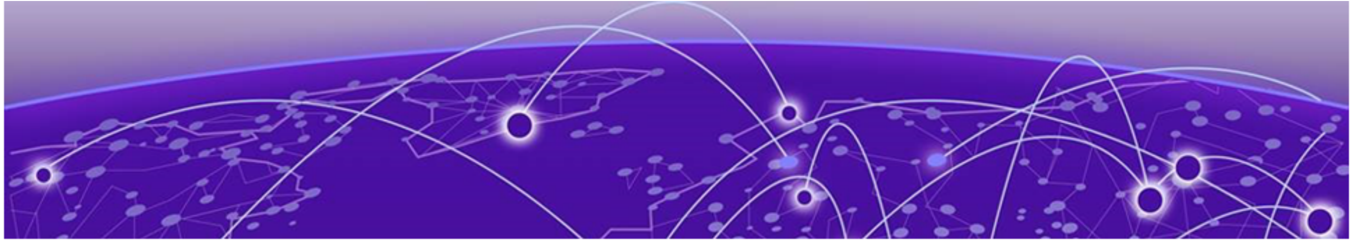
The Information Development team at Extreme Networks has made every effort to ensure the accuracy and completeness of this document. We are always striving to improve our documentation and help you work better, so we want to hear from you. We welcome all feedback, but we especially want to know about:

- Content errors, or confusing or conflicting information.
- Improvements that would help you find relevant information in the document.
- Broken links or usability issues.

If you would like to provide feedback, you can do so in three ways:

- In a web browser, select the feedback icon and complete the online feedback form.
- Access the feedback form at https://www.extremenetworks.com/documentation-feedback/.
- Email us at documentation@extremenetworks.com.

Provide the publication title, part number, and as much detail as possible, including the topic heading and page number if applicable, as well as your suggestions for improvement.



About this Document

What's New in this Document

The following table describes information added to this guide for the Extreme Fabric Automation 2.2.0 software release.

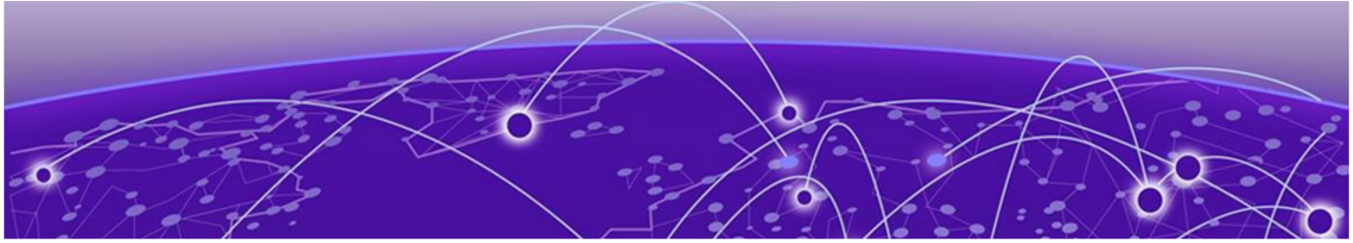
Table 4: Summary of changes

Feature	Description	Link
Data Consistency	This feature ensures that SLX devices have the correct configuration before allowing traffic.	Data Consistency on page 49
Redundant Management Network	This feature provides fault tolerance for the management path.	Redundant Management Network on page 31
RASlog Service	The RASlog service is aware of all devices that are registered with EFA services and processes events only from those devices.	RASlog Service on page 160
Supported Platform Matrix	New tables in this topic describe deployment models and supported TPVM versions.	Supported Platform Matrix on page 21
L3 Tenant Network Services	This feature supports IPv6 Anycast gateways; VRF backup routing, static routing, BFD static routing, local ASNs, and graceful restart; CEP reload delay, LACP timeout for port channels; centralized and distributed routing; and sharing resources across tenants.	Layer 3 Network Services on page 94
Security	New topics for user management (including RBAC enforcement), certificate management, and audit trail logging.	Audit Trail Logging on page 46 EFA Certificate Management on page 43 EFA User Management on page 38

Table 4: Summary of changes (continued)

Feature	Description	Link
SLX device configuration	New commands and topics to configure the SLX devices that EFA manages.	Configure MTU for Physical Ports on page 159 Configure Breakout Ports on page 158 Configure Physical Port Speed on page 158 Change a Physical Port State on page 159 Enable Maintenance Mode on SLX Devices on page 157
Cluster tracking	New functionality to enable a CEP interface to track the state of MCT clusters.	Enable Cluster Tracking on CEP Interfaces on page 103
Running-configs	New command to display a list of current EFA running-configs.	Display EFA Running Configurations on page 45
Deployment	Updated installation and upgrade topics.	Install EFA in a Single-Node Deployment on page 24 Install EFA on TPVM on page 25 Upgrade EFA on TPVM on page 27 Deploy the OVA for EFA on page 28 Upgrade EFA on a Server on page 29 Upgrade EFA on an OVA on page 30
NTP requirements	Improved the list of requirements	EFA Requirements on page 23

For more information about this software release, see the *Extreme Fabric Automation Release Notes*.



Extreme Fabric Automation

Introduction to Extreme Fabric Automation

Extreme Fabric Automation (EFA) is a micro-services-based scalable automation application.

EFA orchestrates the following:

- The life cycle of
  - Small Data Center (small DC) fabrics based on Non-Clos topology
  - 3-stage or 5-stage IP Clos Fabric
- Tenant-aware Layer 2 and Layer 3 networks
- Integration with ecosystems that support HCI (Hyper Convergence Infrastructure) Service
  - VMware vCenter
  - OpenStack
  - Microsoft Hyper-V

The key tenets of this orchestration are as follows:

- Conformance to the EVD (Extreme Validated Design) for IP Fabrics: https://www.extremenetworks.com/resources/extreme-validated-design/extreme-ip-Fabric-architecture/
- Speed of provisioning
- Seamless installation and deployment mechanism
- High in performance, low in resource utilization, with minimal touch points
- Programmable containerized services, through an industry-standard Open API (https://www.openapis.org/)-based programmable interface
- Easy-to-use CLI commands to manage devices in an IP Fabric and tenant networks

EFA consists of core containerized services that interact with each other and with other infrastructure services to provide the core functions of Fabric and tenant network automation.

Asset Service	Provides the secure credential store and deep discovery of physical and logical assets of the managed devices, and publishes the asset refresh or change events to other services.
Fabric Service	Helps orchestrate and visualize BGP-EVPN-based 3-stage IP Clos, 5-stage IP Clos and Non-CLOS fabrics
Tenant Service	Helps manage the Tenants, Tenant Networks, and end points, fully leveraging the knowledge of assets and the underlying fabric.
Inventory Service	Acts as an inventory of all the necessary physical and logical assets of the fabric devices. All other EFA services rely on inventory service asset data for their respective configuration automation.
System Service	Provides EFA system utilities such as supportsave, backup, and restore.
Notification Service	Sends events, alerts, and task updates to external entities.
Authentication Service	Enforces a security boundary between northbound clients and downstream operations between EFA and SLX.
Authorization Service	Validates users and their credentials.
vCenter Service	The vCenter integration provides connectivity between EFA and vCenter using a REST API as documented in the VI SDK. EFA does not connect to individual ESXi servers. All integration is done through vCenter.
OpenStack Service	OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a data center.
Hyper-V Service	The Hyper-V integration supports networking configuration for Hyper-V servers in a data center, manual and automated configuration updates when VMs move, and visibility into the VMs and networking resources that are deployed in the Hyper-V setup.

The following figure illustrates the application functionality in provisioning and discovery.

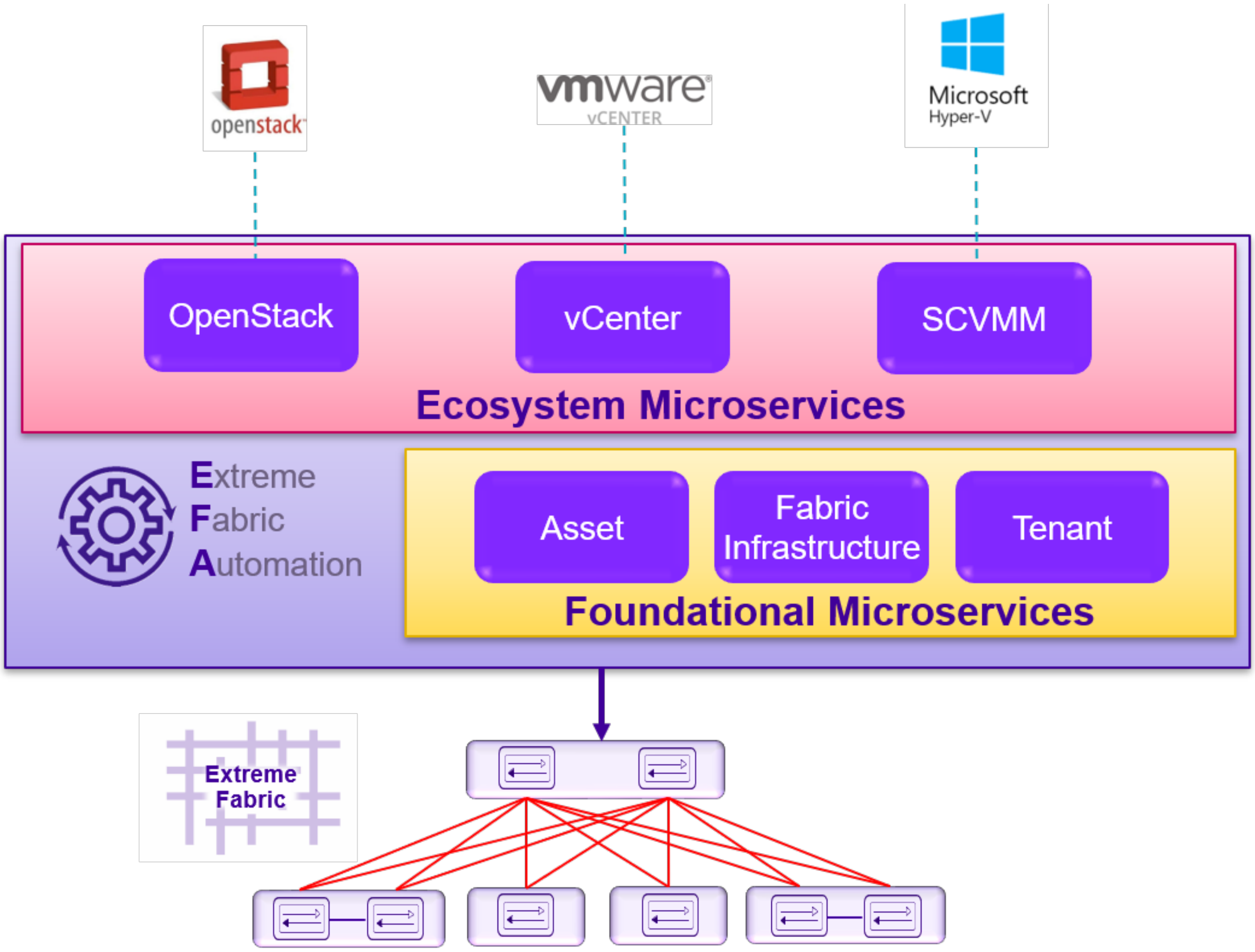


Figure 1: Fabric Automation Microservices

EFA Feature Overview

EFA features allows automation of the fabric (Clos / Non-Clos) life-cycle management, tenant L2/L3 networks life-cycle management on top of the fabric infrastructure and ecosystem (vCenter / Openstack / HyperV) integration.

Fabric Service

Fabric Service is responsible for automating the Fabric BGP underlay and EVPN overlay. By default, the EVPN overlay is enabled but you can disable it before provisioning if necessary. The Fabric Service exposes the CLI and REST API to clients for automating the Fabric underlay and overlay configuration.

Fabric Service features include:

- Small Data Center Topology (non-Clos support)
- Support for 3-stage and 5-stage Clos data center Fabrics
- Support for MCT configuration
- Support for ecosystem integration: OpenStack, VMware vCenter, and Microsoft Hyper-V

Underlay automation includes interface configurations (IP numbered), BGP underlay for spine and leaf, BFD, and MCT configurations. Overlay automation includes EVPN and overlay gateway configuration. The Fabric Service is deployed along with the Inventory and Tenant services.

Tenant Service

The Tenant Service exposes the CLI and REST API for automating the tenant network configuration on the Clos and non-Clos overlay Fabric. Tenant network configuration includes VLAN, BD, VE, EVPN, VTEP, VRF, and router BGP configuration on the necessary Fabric devices to provide Layer 2-Extension, Layer 3-Extension across the Fabric, Layer 2-Handoff, and Layer 3-Handoff at the edge of the Fabric.

Inventory Service

The Inventory Service is a REST layer on top of device inventory details, with the capability to filter data based on certain fields. The Inventory Service securely stores the credentials of devices in encrypted form and makes those credentials available to different components such as the Fabric and Tenant services.

The Inventory Service supports the **`execute-cli`** option for pushing configuration and exec commands that are not included in the EFA CLI to devices. Examples include configuring SNMP parameters or OSPF configurations. This supports lets you use EFA for any SLX-OS command and push the same configuration to multiple devices.

Asset Service

The Asset Service provides the secure credential store and deep discovery of physical and logical assets of the managed devices. The service publishes the Asset refresh and change events to other services.

Notification Service

The Notification Service sends events, alerts, and tasks to external entities. Notifications sent from EFA are derived from the syslog events received from the devices that EFA manages. Alerts are notifications that services in EFA send for unexpected conditions. Tasks are user-driven operations or timer-based tasks such as device registration or Fabric creation.

RASlog Service

The RASlog Service acts as a syslog server to process syslog messages from devices. The service also acts as an SNMP trap receiver to process traps from devices.

Authentication and Authorization Service

Authentication and authorization enforce a security boundary between northbound clients and the downstream operations between EFA and SLX devices.

EFA Deployment on an External VM

You can deploy EFA on an external Virtual Machine to support more than 24 devices or based on where tools are deployed within the Data Center. Running EFA on the TPVM or an external VM provides added deployment flexibility.

EFA Ecosystem Integration

Administrators can use EFA to integrate with several orchestration ecosystems.

EFA provides one-touch integration points with these ecosystems, providing deep insight into VMs, vSwitches, port groups, and hosts, and the translation of these into IP Fabric networking constructs.

VMware vCenter

- Registration of 1 or more vCenter servers in EFA
- Updates for vCenter asset details
- Lists of information about vCenter servers
- Delete or unregister vCenter servers
- Inventory integration
- Tenant Service integration Dynamic updates from vCenter and from EFA services

Microsoft Hyper-V

- SCVMM (System Center Virtual Machine Manager) server discovery
- SCVMM server update
- Periodic polling of registered SCVMM servers
- SCVMM server list
- SCVMM server delete and deregister
- Network event handling

OpenStack

The OpenStack plugin package for ML2 and ML3 includes the following.

ML2 Plugin	• CRUD operations on Network and Port • LAG Support • Provider Network (default, PT)
Trunking (VLAN)	Trunking using virtio ports
SRIOV-VF	Network Operations using SRIOV-VF Passthrough – Intel, Mellanox
SRIOV-PF	Network Operations using SRIOV-VF Passthrough – Intel, Mellanox
Layer 3 (E-W)	East West Traffic using virtio ports (Neutron Router, Router Interface, Subnet CRUD operations)

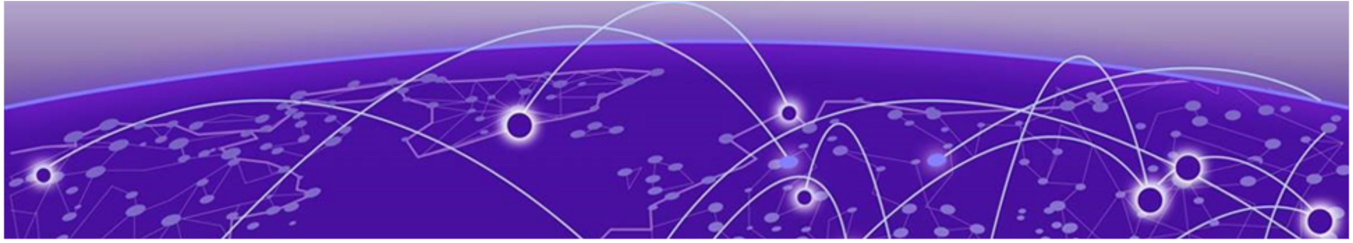
VMotion	Virtual Machine Migration
BD Support	Support for BD-enabled in Tenant Service
Multi VIM Support	Multiple Tenants managed from OSS
Multi-Segment Support	Multiple segments using SRIOV (PF/VF)+ Virtio (DHCP)
CEP Support	Support for single-homed connections to the edge port

Rest APIs for EFA

When EFA is installed, the REST API guide is available as an HTML reference: `http://<host_ip>/docs`.

The API guide is a good reference to help integrate with other automation tools. The REST API is specified by OpenAPI and Swagger. The API guide is not available with TPVM installations.

For more information, see the selection of API guides on the Extreme Networks website. Select **Extreme Fabric Automation** here: https://www.extremenetworks.com/support/documentation/product-type/software/



EFA Deployment

Supported Platform Matrix

EFA provides seamless support for upgrade and downgrade of SLX devices across pre-20.1.x and 20.1.x images to keep the device and application configuration in sync.

Extreme Fabric Automation: Deployment Models

Table 5: EFA Deployment on an External Server

Version	Deployment	Managed SLX Devices	Multi-Fabric Support	Ubuntu Version	Server Requirement
EFA 2.1.0	External server (Bare metal or OVA)	More than 24	Yes	16.04	CPU: 4 cores Storage: 50 GB RAM: 8 GB
EFA 2.2.0	External server (Bare metal)	More than 24	Yes	16.04, 18.04	
EFA 2.2.0	External server (OVA)	More than 24	Yes	18.04	

Table 6: EFA Deployment on a TPVM

Version	Deployment	Managed SLX Devices	Multi-Fabric Support	Ubuntu Version	Minimum SLX-OS version
EFA 2.1.0	SLX 9150, SLX 9250, or SLX 9640 TPVM	Up to 24	Yes	16.04	20.1.1
EFA 2.2.0	SLX 9150, SLX 9250, or SLX 9640 TPVM	Up to 24	Yes	18.04	20.1.2

Table 7: TPVM Versions

TPVM Version	SLX-OS 20.1.1	SLX-OS 20.1.2	Ubuntu Version	EFA Version
3.0	Yes	Yes	16.04	2.1.0
4.0	No	Yes	18.04	2.2.0

Extreme IP Fabric: Topologies and PINs

Table 8: Extreme IP Fabric: Topologies and PINs

Platforms	SLX-OS Release	Leaf	Spine	Super Spine	Border Leaf	Small DC Fabric
SLX 9150	20.1.x	✔				✔
SLX 9250	20.1.x	✔	✔			✔

Table 8: Extreme IP Fabric: Topologies and PINs (continued)

Platforms	SLX-OS Release	Leaf	Spine	Super Spine	Border Leaf	Small DC Fabric
SLX 9540	18r.1.00aa, 18r.1.00b,18r. 1.00c, 18r. 1.00cc				✔	
SLX 9540	20.1.x	✔			✔	
SLX 9640	20.1.x				✔	
SLX 9140	18s.1.01, 18s. 1.01a, 18s. 1.01c, 18s.1.03	✔				✔
SLX 9240	18s.1.01, 18s. 1.01a, 18s. 1.01c, 18s.1.03	✔	✔	✔		
SLX 9030	18x.1.00, 18x. 1.00a, 18x. 1.00b	✔				
SLX 9850	18r.1.00aa, 18r.1.00b, 18r. 1.00c		✔	✔		

EFA Port Requirements

The following tables identify ports that must be available and not used by other services. EFA installation fails if a required port is not available.

Table 9: Port requirements

Port	Service
80	EFA HTTP requests
443	EFA HTTPs requests
514	Syslog service
6443	K3S
8079	Host authentication
10010	Containerd
30085	OpenStack service
30432	Postgres database
30500	Logstash (non-TPVM)
30601	Kibana (non-TPVM)
30672	Rabbitmq
30920	Elasticsearch (non-TPVM)

Table 9: Port requirements (continued)

Port	Service
30930	Elasticsearch (non-TPVM)
31672	Rabbitmq management

EFA Requirements

Review this topic for requirements for host names, NTP, user privileges, DNS configuration, passwordless SSH, and IP addresses.

- Host names: Host names must be alphanumeric. Hyphens are the only special characters allowed. No other special characters are allowed by Kubernetes for cluster formation or by the K3s service.
- NTP: The server on which EFA is installed must use NTP or be synchronized to the correct time and timezone. Having the correct time and timezone ensures the following:
  - Self-signed certificates have valid start and expiration times.
  - EFA logs have the correct time stamp.
  - The K3s service starts without errors.

  You can edit `etc/systemd/timesyncd.conf` to select NTP servers in the `[Time]` section of the configuration file. The `NTP=` option takes a space-separated list of host names or IP addresses. NTP suggests selecting as many servers as is feasible, but at least 3. Select from the pool of publicly available servers or your company's internal NTP servers. For example:

  ```
  [Time]
  NTP=0.pool.ntp.org 1.pool.ntp.org 2.pool.ntp.org 3.pool.ntp.org
  ```

  You can use the following commands to access `timesyncd.conf` and to synchronize your changes.

  ```
  # sudo vim /etc/systemd/timesyncd.conf
  # sudo service systemd-timesyncd restart
  # systemctl status systemd-timesyncd
  # sudo timedatectl set-timezone <your_time_zone>
  ```
- NTP: All devices that EFA manages must use NTP to ensure easy audit trails and logging from EFA.
- User privileges: The user who installs EFA must be a root user or have `sudoers` privileges to ensure components are installed correctly. Installation fails if this requirement is not met.
- DNS: To ensure that DNS resolution of Kubernetes functions correctly, DNS configuration on the nodes must be valid or the `/etc/resolv.conf` file must be empty.
  - Ensure that `nslookup` returns the correct host name based on the IP address. For example, `nslookup node1`.
  - Ensure that the DNS servers listed in the `/etc/resolv.conf` file can resolve to the addresses of all the nodes. For example, `dig <node_hostname> +short` should return the correct IP addresses assigned to the hosts.
- TPVM: With the 4.0.x release of TPVM, you can configure DNS, NTP, and LDAP as part of deploying TPVM. For more information, see "Guest OS for TPVM" in the *Extreme SLX-OS Management Configuration Guide*.

EFA Installation Modes

You can install EFA in secure mode or non-secure (standard) mode.

You can choose one of these modes when you install EFA:

- Secure mode: Traffic to EFA uses the HTTPS protocol. All non-HTTP requests are redirected to the secure port. Traffic out of EFA (toward northbound interfaces) uses TLS.
- Standard mode: Traffic to EFA uses the HTTP protocol. Traffic toward northbound interfaces also uses HTTP.

You cannot change a secure installation to a standard installation. Nor can you change a standard installation to a secure installation.

Install EFA in a Single-Node Deployment

You can install EFA in a single-node, non-TPVM deployment.

Verify the following prerequisites:

- CPU: 4 cores
- Storage: 50 GB
- RAM: 8 GB
- OS: Ubuntu 16.04 or 18.04

Ensure you have configured NTP according to EFA Requirements on page 23.

To install EFA, you must be a root user or have `sudoers` privileges.

1. Download the image (*.tar.gz).
2. Untar the image.

   ```
   $ tar -xzf efa-v2.x.x.tar.gz
   ```
3. Verify the PGP signature as described in article 48172 on the Extreme Portal.

   The process is also described in the *Extreme Fabric Automation Release Notes*.
4. Change to the EFA directory.

   ```
   $ cd efa
   ```
5. Run the deployment script.

   When prompted, ensure that you select single-node and either secure or standard mode. For more information, see EFA Installation Modes on page 24.

This example shows the flow for installing EFA in a single-node deployment.

```
$ source deployment.sh

Step 1: Checking for EFA Stack ...
1) Single-node deployment
2) Multi-node deployment
3) Quit
Enter your choice (1/2/3): 1
Single-node Deployment
1) Enable secure mode (https with http redirection)
2) Standard mode (http)
3) Quit
Enter your choice (1/2/3): 1
Secure mode: yes
Step 2: Installing EFA 2.2.0-14...
```

```
Started checking system configuration...
Checking required ports availability...
Verifying if k3s services are running ...
Unpacking images of EFA...
Verifying if hostauth service are running ...
Generating Unix user for EFA use...
Saving EFA user information for this node (sam-kub-master)...
Deploying EFA application...
* Deploying core services
* Deploying ecosystem services
Extreme Fabric Automation Stack is now Deployed and Ready!
```

Note
If the installation only partially succeeds, you can remove it using the **`source deployment.sh -o undeploy`** script from the `/efa` directory on the node where you installed EFA.

EFA Installation and Deployment on TPVM

TPVM (Third-Party Virtual Machine) is a general server that resides on the Extreme SLX 9150 and the SLX 9250. When EFA is deployed on a TPVM, no other applications can be run on that TPVM.

In a TPVM deployment, EFA is a microservice-based Fabric automation engine that leverages the K3S Kubernetes cluster as an underlying infrastructure for the EFA services deployment. You can install or upgrade the EFA application on a TPVM with one SLX command. The EFA application binary is shipped with the SLX 9150 and SLX 9250, along with the binaries for SLX-OS and the TPVM. Decoupling EFA from SLX-OS allows for upgrades to EFA without a need to upgrade SLX-OS or the TPVM. EFA can be deployed on one of the SLX devices in the Fabric to manage the Fabric.

The EFA package can be found under the `/efaboot` folder on the SLX device. Additionally, for an incremental EFA image upgrade, you can copy the EFA tar file to the `/efaboot` directory on the SLX device before the deployment.

Requirements

TPVM must be installed and running on the SLX device. You can accomplish these tasks by running the **`tpvm deploy`** command on the SLX device.

```
tpvm deploy mgmt [dhcp/ipaddr] [gw] admin-pwd allow-pwdless confirm-pwd
```

See the *Extreme SLX-OS Command Reference* for specific information about using this command.

Install EFA on TPVM

You can install EFA on a TPVM in a single-node deployment.

The EFA tar must be available on the `/efaboot` partition of the SLX device. You need root access to the device.

EFA on TPVM is supported only on the SLX 9150 and SLX 9250 platforms.

1. Verify that the TPVM is set up for an EFA deployment.
   a. Validate that the TPVM is running version 4.0.0 and that SLX-OS is running with at least the minimum requirements.

      ```
      # show tpvm status
      # show version
      # lsb_release -a
      ```

   b. Validate that the TPVM has an assigned IP address.

      ```
      # show tpvm ip-address
      ```

   c. Validate that the SSH keys are uploaded.

      ```
      # show tpvm status
      ```

   d. Validate that passwordless access is configured.

      ```
      # show tpvm status
      ```

   e. Confirm NTP on the TPVM.

      ```
      # tpvm config ntp add server <ip>
      ```

   f. Validate that NTP is synchronized.

      ```
      # show tpvm config ntp
      ```

   g. If necessary, log in to TPVM and configure the NTP time zone.

      ```
      # sudo timedatectl set-timezone your_time_zone
      ```

2. Enter SLX Linux mode.

   ```
   # start-shell
   # cd /efaboot
   ```

3. Copy the EFA tar file to the SLX device.

   ```
   # scp <efa-bundle>
   ```

4. Deploy EFA on TPVM from the SLX shell.

   ```
   device# efa deploy
   ```

   When prompted, ensure that you select single-node and either secure or standard mode. For more information, see EFA Installation Modes on page 24.

5. Verify the status of the installation.

   ```
   # show efa status
   ```

This example shows the flow for installing EFA on a TPVM in a single-node deployment, which takes approximately 15 minutes.

```
device# efa deploy
Starting "efa deploy", please DO NOT hit CTRL+C
Step 1: Checking if TPVM is deployed ...
Step 2: Get IP Addressed assigned to TPVM to deploy EFA
IP Address of the TPVM 10.24.95.69
Step 3: Checking for EFA packages in /efaboot directory
Step 4: Deploying EFA package efa-2.2.0-15.tar.gz on 10.24.95.69
Step 5: Checking for EFA Stack...
1) Single-node deployment
2) Multi-node deployment
3) Quit
Please enter your choice (1/2/3): 1
Single-node Deployment
1) Enable secure mode (https with http redirection)
2) Standard mode (http)
```

```
3) Quit
Please enter your choice(1/2/3): 1
Secure mode: yes
Step 6: Installing EFA 2.2.0-14...
Started checking system configuration ...
Checking required ports availability ...
Verifying if k3s services are running...
Unpacking images of EFA...
Generating Unix user for EFA use...
Saving EFA user information for this node (tpvm)...
Deploying EFA application...
* Deploying core services
* Deploying ecosystem services
Extreme Fabric Automation Stack is now Deployed and Ready!
```

Upgrade EFA on TPVM

If a new version of the EFA bundle is required or if no EFA bundle is present, the EFA .tar file must be copied to the `/efaboot` directory.

1. Verify that the TPVM is set up for an EFA deployment.
   a. Validate that the TPVM is running version 4.0.0 and that SLX-OS is running with at least the minimum requirements.

   ```
   # show tpvm status
   # show version
   # lsb_release -a
   ```

   b. Validate that the TPVM has an assigned IP address.

   ```
   # show tpvm ip-address
   ```

   c. Validate that the SSH keys are uploaded.

   ```
   # show tpvm status
   ```

   d. Validate that passwordless access is configured.

   ```
   # show tpvm status
   ```

   e. Confirm NTP on the TPVM.

   ```
   # tpvm config ntp add server <ip>
   ```

   f. Validate that NTP is synchronized.

   ```
   # show tpvm config ntp
   ```

   g. If necessary, log in to TPVM and configure the NTP time zone.

   ```
   # sudo timedatectl set-timezone your_time_zone
   ```

2. Determine whether more than one EFA version is available in the SLX `/efaboot` directory.
   - If no version is available, the installer stops.
   - If more than one version is available, you have the option to pick a version.
   - If only one version is available, the installer picks up that version.
3. Determine whether the TPVM already has a version of EFA installed.
   - If the same version is already installed, the installer stops.
   - If no EFA is installed, the installer continues with installation.
   - If a different version is detected, the upgrade or downgrade continues, depending on the detected version.

4. Copy the EFA tar file to the SLX device.

   ```
   # start-shell
   # cd /efaboot
   # scp <efa-bundle>
   ```

5. Deploy EFA on the TPVM from the SLX device.

   ```
   # efa deploy
   ```

6. Validate the upgrade.

   a. Verify the status of EFA.

      ```
      # show efa status
      ```

   b. Verify the status of the Fabric.

      ```
      # efa fabric show
      ```

   c. Verify the status of the tenant.

      ```
      # efa tenant show
      ```

   d. Verify the status of the EPG (endpoint group).

      ```
      # efa tenant epg show
      ```

   e. If the output of the commands indicates that any network is in **cfg-refreshed** state, run the following commands.

      - Run this command for each device in the tenant: `efa tenant debug device drift --device-ip <ip-address> --reconcile`
      - `efa inventory device update --fabric <fabric-name>`

Deploy the OVA for EFA

Open Virtual Appliance (OVA) is an OVF file packaged with a base image Ubuntu image and installed with EFA.

Prerequisites:

- The virtual machine (VM) on which you deploy the OVA requires a network adapter with a valid IP address and DNS. The IP address is required to configure the SLX devices to forward syslog entries back to the VM. The VM needs DNS configuration to resolve the URL during setup and forwarding of events to the notification subscriber.
- The VM must be able to access switches and the notification subscriber.
- For networks without DHCP, you must assign valid, static IP addresses and DNS. Then reboot the VM. Ensure that all services are up and running before running commands.

OVA is compatible with VMware ESXi servers and can be deployed with VMware products. For more information about supported Ubuntu versions, see Supported Platform Matrix on page 21.

Use the OVA image for new installations only.

Warning

- Do not change the host name of the .ova VM after deploying the .ova image. Doing so prevents EFA PODs from coming online.
- The EFA 2.2.0 OVA is not supported for Oracle VirtualBox. The syslog service requires port forwarding for port 514 on UDP, but the source IP address of the syslog message will be changed from the SLX device to the host IP, which the syslog service ignores.

1. Download the `EFA_v2.x.x_<build_number>.ova` file.
2. Run the OVA.
3. Start the VM.

   The credentials for the OVA installation are:

   - Admin/Password: admin/password
   - Root/Password: ubuntu/ubuntu

   When the VM starts, a start-up script checks whether the IP address of the primary interface eth0 has changed since it was last configured. If the IP address has changed, the script updates the EFA profile and configuration files appropriately and reapplies the k3s application deployment template. This operation takes a few minutes to complete. On subsequent VM reboots, if the IP address has not changed, no operation is performed by the start-up script. The logs are located under `/var/log/efa/installer`.
4. Sign in to the VM as the admin user and then use **`sudo`** to run commands (such as **`sudo efa supportsave`**, **`sudo efa backup`**, and **`sudo efa restore`**).

   The new admin user is added in the build.

Upgrade EFA on a Server

You can upgrade EFA.

1. Download the image (*.tar.gz) to a new sub-folder.
2. Untar the image.

   ```
   # tar -xfz efa-v2.x.x.tar.gz
   ```
3. Verify the PGP signature as described in article 48172 on the Extreme Portal.

   The process is also described in the *Extreme Fabric Automation Release Notes*.
4. Change to the EFA directory.

   ```
   # cd efa
   ```
5. Configure NTP with Ubuntu commands.

   ```
   # sudo vim /etc/systemd/timesyncd.conf
   # sudo service systemd-timesyncd restart
   # systemctl status systemd-timesyncd
   # sudo timedatectl set-timezone your_time_zone
   ```

6. Run the deployment script.

   ```
   # source deployment.sh
   ```

   If the previous deployment stack is running, the deployment script presents the following options:

   - Remove the current stack. Select this option to remove the entire stack.
   - Upgrade or Redeploy. If you are running the deployment script from the new tar ball, select this option to upgrade without erasing the current database. You have this same option if the script is run from the same folder, in which case the stack is redeployed.
   - Quit. Select this option if you do not want to change the current stack.

7. Validate the upgrade.

   a. Verify the status of EFA.

   ```
   # show efa status
   ```

   b. Verify the status of the Fabric.

   ```
   # efa fabric show
   ```

   c. Verify the status of the tenant.

   ```
   # efa tenant show
   ```

   d. Verify the status of the EPG (endpoint group).

   ```
   # efa tenant epg show
   ```

   e. If the output of the commands indicates that any network is in cfg-refreshed state, run the following commands.

   - Run this command for each device in the tenant: `efa tenant debug device drift --device-ip <ip-address> --reconcile`
   - `efa inventory device update --fabric <fabric-name>`

Upgrade EFA on an OVA

Open Virtual Appliance (OVA) is an OVF file packaged with a base image Ubuntu image and installed with EFA.

See Deploy the OVA for EFA on page 28 for a list of prerequisites.

1. Log in to the OVA as admin.
2. Change to the root.

   ```
   # sudo su
   ```

3. Copy the new tar file to `/opt/godcapp/`.
4. Extract the tar.

   ```
   # tar -xvf efa-2.2.0.tar.gz
   # cd efa
   ```

5. Run the deployment script.

   ```
   # source deployment.sh
   ```

6. Select the `Upgrade` option.
7. When the upgrade is complete, set the environment variable.

   ```
   # source /etc/profile.d/efa_env.sh
   ```

   At this point, EFA commands can be run only by the root user. A user must be assigned the SystemAdmin role to run commands as an admin user.

8. To assign the SystemAdmin role to a user, take the following steps.
   a. Log in to the OVA as admin.
   b. Change to the root.

   ```
   # sudo su
   ```

   c. Assign the SystemAdmin role to an admin user.

   ```
   # efa auth rolemapping add  --name admin --type user
   --role SystemAdmin
   ```

   d. Validate the role.

   ```
   # efa auth rolemapping show
   ```

Uninstall EFA

Use the **no efa deploy** command to uninstall EFA.

Take the following step to uninstall the current instance of EFA, including uninstalling EFA from a TPVM.

Run the **no efa deploy** command.

```
# no efa deploy
```

Redundant Management Network

Redundant Management Network provides fault tolerance for the management path. This is done using Linux bonding by pairing the physical Management port of the chassis with any one of the physical front panel User Ports.

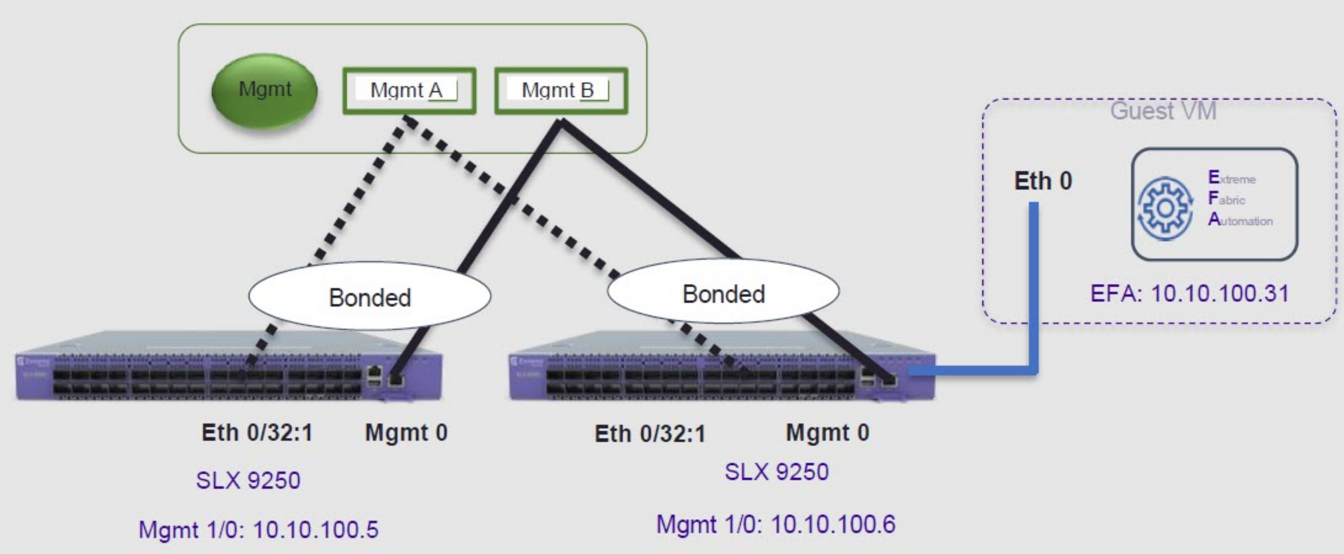


Figure 2: Redundant Management Network Overview

Linux Bonding

The `redundant-management enable` command can be used to pair one of the front panel ports with the conventional `Mgmt 0` port to form a Linux Bonding interface, `bond0` at SLX Linux OS.

- The Linux bond will be in Active/Standby mode. The Physical Management port is the primary and active port. The configured front panel port will be in Standby mode.
  - `mode 1` supported by Linux Bonding with `Mgmt 0` (`eth0`) is the primary port.
  - The front panel port allows traffic through it only if `Mgmt 0` is down. `Mgmt 0` takes over Active port as soon as it recovers.
- If the active primary `Mgmt 0` path experiences failure, SLX OS and TPVM OS can be reached through Standby path.

Supported Ports

Any SLX front panel port can be used at native speed and property for Linux Bonding.

Note

- SLX 9640 and SLX 9150 - Preferred ports are 10G/1G port in 1G mode.
- SLX 9640 - Avoid Insight port 0/24.
- SLX 9250 - Breakout mode 4x1G ports are available to allow the Mellanox adapter with 1G transceiver. As the adapter occupies whole cage, only the first member port (:1) can be used as redundant management interface.

No Redundancy Period

Redundancy is not supported if the device is reloaded or in ZTP mode.

- After reloading a device, use the `redundant-management enable` command or startup config replay to enable Linux Bonding or redundancy.
- Upon factory arrival, across first power cycle, or due to `write erase` CLI, ZTP mode is set in with factory default configuration.
- Breakout mode 1G ports are not supported in factory default configuration.

Standby Port Rate Throughput

Since internal path for Standby traffic is Control Plane traffic on PCIe Channel between ASIC and CPU, its function of internal CPU load is totally unrelated and independent of front panel physical port limit and capability.

Datapath

SLX Linux, boots with `bond0` interface with `Primary Active Slave eth3` (Physical Management 0 Interface). Interface `bond0` serves as slave to vBridge (`eth0`) which serves as Management 0 interface to SLX Linux and all applications on it. This `eth0` is connected though Linux Tap to `TPVM eth0`. `TPVM eth0` contains a separate MAC. IPv4 address is assigned to `eth0` through DHCP or static.

At SLX Linux, logical proxy interface `Eth0.15` or Say `Eth0.32.1` is created to represent the front panel port as Standby member for `bond0`.

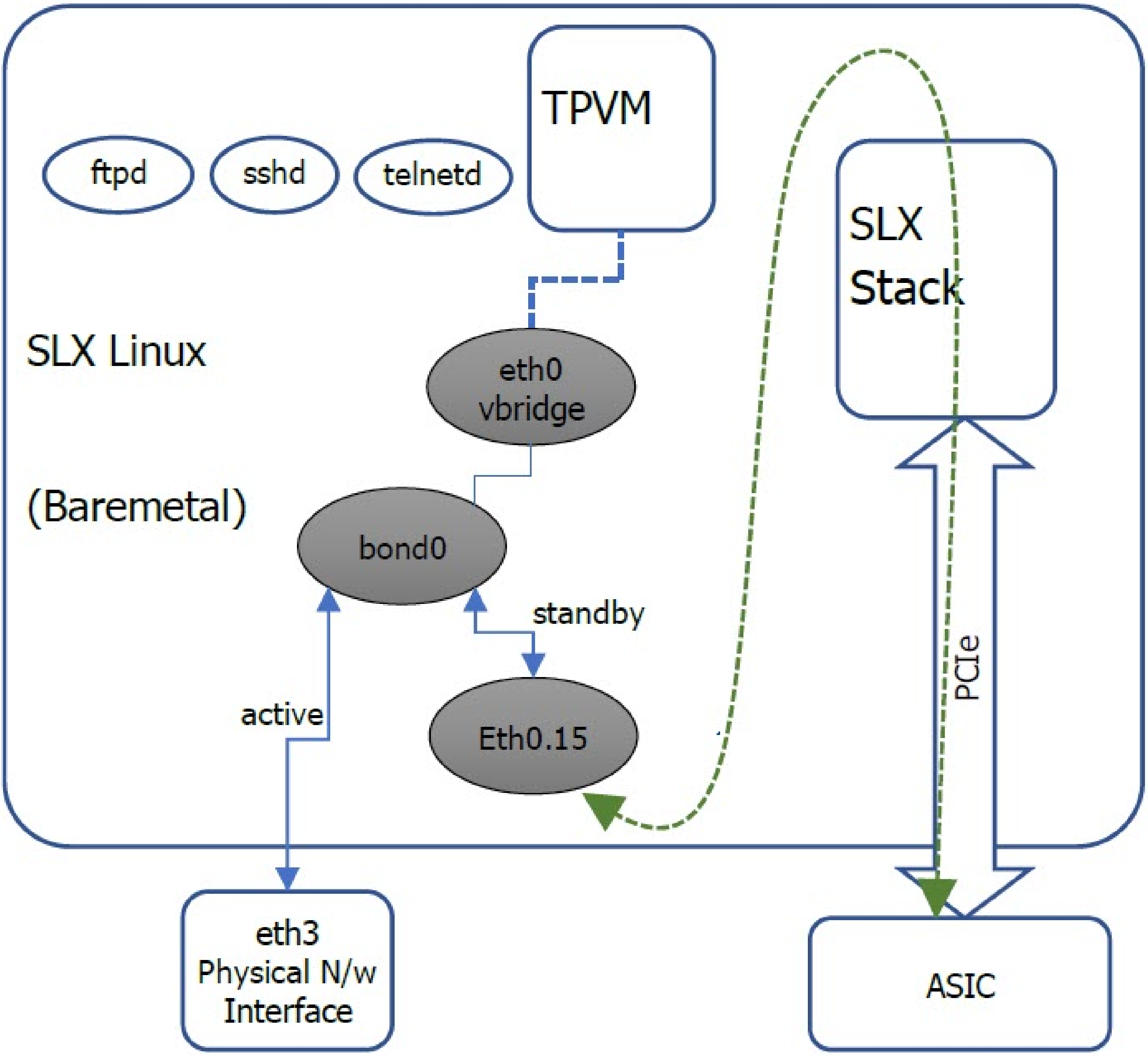


Figure 3: Datapath overview

Enable Redundant Management

1. Enter global configuration mode.

```
SLX # configure terminal
```

2. Enter interface configuration mode.

```
SLX(config)# interface ethernet 0/15
```

3. Enable Redundant Management.

   ```
   SLX(conf-if-eth-0/15)# redundant-management enable
   ```

Port 0/15 at 10G speed:

```
SLX # config
SLX(config)# interface ethernet 0/15
SLX(conf-if-eth-0/15)# redundant-management enable
SLX(conf-if-eth-0/15)# no shut
```

Port 0/15 at 1G speed:

```
SLX # config
SLX(config)# interface ethernet 0/15
SLX(conf-if-eth-0/15)# speed 1000
SLX(conf-if-eth-0/15)# redundant-management enable
SLX(conf-if-eth-0/15)# no shut
```

Port 0/15 on SLX-9250 only with Mellanox Adapter at 1G speed:

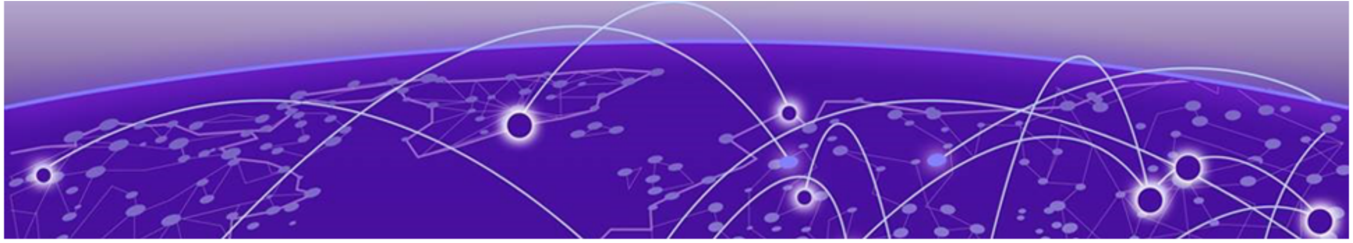
```
SLX# conf t
SLX(config)# hardware
SLX(config-hardware)# connector 0/15
SLX(config-connector-0/49)# breakout mode 4x1G
SLX(config-connector-0/49)# end
SLX# conf t
SLX(config)# interface Ethernet 0/15:1
SLX(conf-if-eth-0/49:1)# redundant-management enable
SLX(conf-if-eth-0/49:1)# no shut
```

```
SLX# show interface Management 0
interface Management 0
line-speed actual "1000baseT, Duplex: Full"
oper-status up
ip address "static 10.24.12.89/22"
ip gateway-address 10.24.12.1
ipv6 ipv6-address [ ]
ipv6 ipv6-gateways [ ]
redundant management port 0/15

SLX# show ip interface brief
Flags: I - Insight Enabled U - Unnumbered interface M - Redundant management port
Interface          IP-Address     Vrf            Status                 Protocol
================   ===========    =========     =====================   ========
Ethernet 0/1       unassigned     default-vrf   administratively down   down
Ethernet 0/2       unassigned     default-vrf   administratively down   down
...
Ethernet 0/15 (M)  unassigned     mgmt-vrf      administratively down   down
...

SLX# show interface ethernet 0/15
Ethernet 0/15 is admin down, line protocol is down (admin down)
Redundant management mode is enabled
Hardware is Ethernet, address is 609c.9f5a.a35f
Current address is 609c.9f5a.a35f
Pluggable media not present
Description: Insight port
Interface index (ifindex) is 202350592 (0xc0fa000)
MTU 9216 bytes
Maximum Speed : 10G
LineSpeed Actual : Nil
LineSpeed Configured : Auto, Duplex: Full
Priority Tag disable
Forward LACP PDU: Disable
Route Only: Disabled
```

```
Tag-type: 0x8100
Last clearing of show interface counters: 00:01:13
Queueing strategy: fifo
FEC Mode - Disabled
Receive Statistics:
0 packets, 0 bytes
Unicasts: 0, Multicasts: 0, Broadcasts: 0
64-byte pkts: 0, Over 64-byte pkts: 0, Over 127-byte pkts: 0
Over 255-byte pkts: 0, Over 511-byte pkts: 0, Over 1023-byte pkts: 0
Over 1518-byte pkts(Jumbo): 0
Runts: 0, Jabbers: 0, CRC: 0, Overruns: 0
Errors: 0, Discards: 0
Transmit Statistics:
0 packets, 0 bytes
Unicasts: 0, Multicasts: 0, Broadcasts: 0
Underruns: 0
Errors: 0, Discards: 0
Rate info:
Input 0.000000 Mbits/sec, 0 packets/sec, 0.00% of line-rate
Output 0.000000 Mbits/sec, 0 packets/sec, 0.00% of line-rate
Route-Only Packets Dropped: 0
Time since last interface status change: 00:01:13
```



EFA System Management

Verify the Running System and Services

After any of the following scenarios, wait 10 minutes for EFA micro-services to be operational before you run EFA commands.

- Powering on the OVA
- Rebooting the OVA
- Rebooting the TPVM
- Rebooting the SLX (which also reboots the TPVM)
- Rebooting the server on which the EFA is installed

You can use various commands and scripts to verify the status of the EFA system, to help troubleshoot, and to view details of EFA nodes, PODs, and services.

1. Verify the K3s installation in a TPVM.
   a. Run the **show efa status** command from the SLX command prompt.

```
device# show efa status
       NAME   STATUS   ROLES    AGE     VERSION
    TPVM   Ready    master   6m59s   v1.14.5-k3s.1
    admin@10.24.51.226's password:
    NAME                            READY      STATUS    RESTARTS   AGE
    pod/godb-service-wk57h          1/1         Running  0          6m11s
    pod/gofabric-service-8v8b2      1/1         Running  3          6m12s
    pod/goinventory-service-4kggf   1/1         Running  3          6m12s
    pod/gotenant-service-xcqf6      1/1         Running  3          6m12s
    pod/rabbitmq-0                  1/1         Running  0          6m12s
    pod/rabbitmq-1                  1/1         Running   0         4m51s
```

   Output varies by type of deployment and the services that are installed.

2. View details of EFA nodes, PODs, and services.

   a. Run the **efactl status** script.

   ```
   root@node1:/home/ubuntu/efa# efactl status
   NAME       STATUS     ROLES     AGE     VERSION
   node1      Ready      Master    22m     v1.17.3+k3s1
   node2      Ready      Master    22m     v1.17.3+k3s1
   NAME                                     Ready   Status      Restarts  Age
   pod/efa-api-docs-55c97cbdf-qnqg4          1/1     Running     0         16m
   pod/rabbitmq-6qkmp                        1/1     Running     0         16m
   pod/godb-service-6c7f7d865b-q2rhv         1/1     Running     0         16m
   pod/rabbitmq-4671j                        1/1     Running     0         16m
   ```

   This example shows only a few of all possible rows of detail.

3. Verify that all PODs are in a running state.

   a. Run the **k3s kubectl get pods -n efa** command.

   ```
   # k3s kubectl get pods -n efa

   NAME                                     READY   STATUS    RESTARTS   AGE
   goswitch-service-958fcfb4f-qddnw         1/1     Running   4          72d
   godb-service-57bd99747-f4cxb             1/1     Running   4          83d
   efa-api-docs-6bb5dbcc74-br485            1/1     Running   4          72d
   filebeat-service-86ddd654b6-z9zhr        1/1     Running   4          72d
   goopenstack-service-554c57548f-bjwtb     1/1     Running   8          72d
   logstash-service-6c49f8dd85-mngd4        1/1     Running   4          72d
   rabbitmq-0                               1/1     Running   7          72d
   govcenter-service-f6b49d9b9-s24wk        1/1     Running   19         72d
   gohyperv-service-854654f6b9-m9mv8        1/1     Running   20         72d
   goinventory-service-59d9b798d8-s9wn6     1/1     Running   20         72d
   gotenant-service-55fd8889d8-g8rgb        1/1     Running   19         72d
   gofabric-service-69d8995fc6-swnqw        1/1     Running   19         72d
   elasticsearch-service-5cdc874b5d-f6rjh   1/1     Running   4          72d
   kibana-service-7748b6db9c-lbm9w          1/1     Running   6          72d
   metricbeat-service-76c4874887-mbm7h      1/1     Running   32         72d
   ```

4. Verify the status of the Authentication service.

   a. Run the **systemctl status hostauth.service** script.

   ```
   $ systemctl status hostauth.service
   hostauth.service - OS Auth Service
   Loaded: loaded (/lib/systemd/system/hostauth.service; enabled; vendor preset:
   enabled)
   Active: active (running) since Thu 2020-04-23 07:56:20 UTC; 23 h ago
   Main PID: 23839 (hostauth)
   Tasks: 5
   CGroup: /system.slice/hostauth.service
           23839 /apps/bin/hostauth

   Apr 23 07:56:20 tpvm2 systemd[1]: Started OS Auth Service
   ```

5. Restart a service with the **efactl restart-service gotenant-service** script.

6. Identify the active node that serves as the database for Kubenetes clusters.

   a. Run the **ip addr show** command from all nodes.

   b. Verify that on one of the Ethernet interfaces, the virtual IP address shows up as the secondary IP address.

Log in to EFA

Use of the EFA command line requires a valid, logged-in user.

1. Verify the status of the EFA deployment using one of the following methods.
   - Run the SLX **show efa status** command.
   - Run the EFA **efactl status** script.

   For more information, see Verify the Running System and Services on page 36.
2. Log in to EFA.

   ```
   $ efa login --username <username>
   Password: <password>
   ```

   The <username> variable is optional. If you do not provide a user name, log-in defaults to the current (Unix) user.

   With a successful log-in, the command prompt shows the logged-in user in green text. If the log-in is not successful, the command prompt is displayed in red text.
3. To log out of EFA, run the **efa logout** command.

EFA User Management

EFA users are validated with Unix authentication and LDAP and managed with Role-based Access Control (RBAC).

EFA validates users and their credentials with the following mechanisms:

- Unix authentication (local and remote) on the host where EFA is installed. Host credentials are the default validation method if LDAP validation fails.
- External LDAP server. Users configured in LDAP use their LDAP credentials to log in to EFA.

After EFA is deployed, the installing user has the role of SystemAdmin and has complete access to EFA functionality. For installation on TPVM, this user has the user name of ‘extreme’.

By default, no other host OS users can access EFA unless the SystemAdmin assigns the appropriate roles.

LDAP supports three modes for fetching the roles assigned to a user.

- The role is available as an attribute in the user Distinguished Name (DN) entry. Group attribute definition is not needed.
- The user has a "memberOf" attribute or any appropriate group DN attribute to identify the groups assigned to the user. Assign the corresponding LDAP group to a role in EFA.
- LDAP groups have user entries in their group definitions. Assign the LDAP groups to roles in EFA.

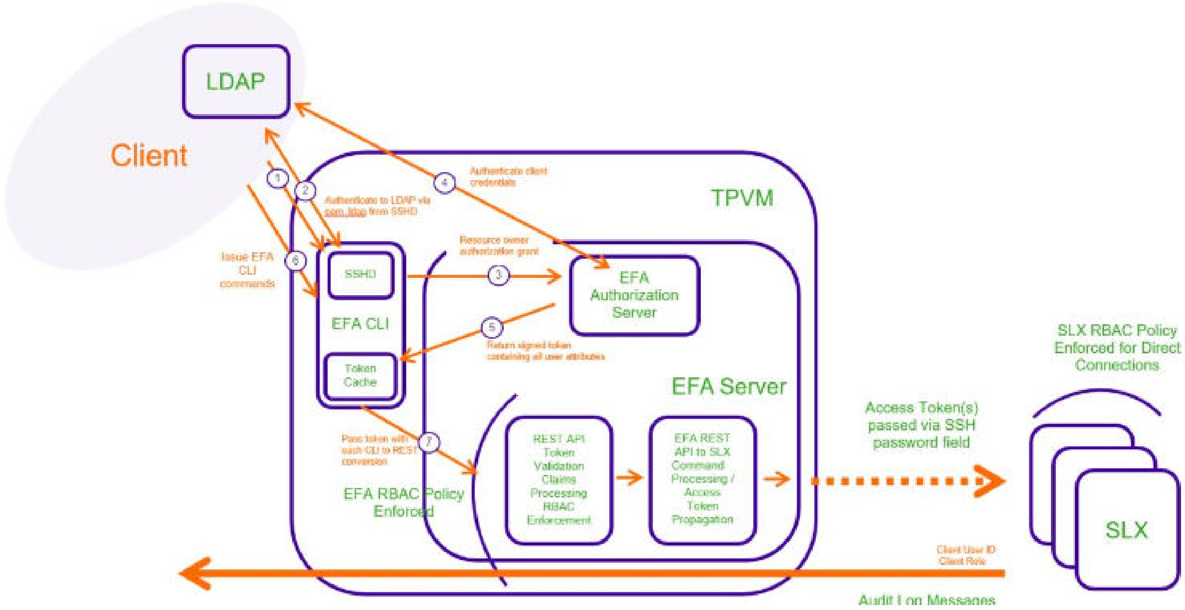

LDAP
Client
1
2
Authenticate to LDAP via pam_ldap from SSHD
4
Authenticate client credentials
TPVM
Issue EFA CLI commands
6
SSHD
EFA CLI
Token Cache
Resource owner authorization grant
3
EFA Authorization Server
5
Return signed token containing all user attributes
EFA Server
Pass token with each CLI to REST conversion
7
EFA RBAC Policy Enforced
REST API Token Validation Claims Processing RBAC Enforcement
EFA REST API to SLX Command Processing / Access Token Propagation
Access Token(s) passed via SSH password field
SLX RBAC Policy Enforced for Direct Connections
SLX
Client User ID
Client Role
Audit Log Messages

Figure 4: EFA LDAP Workflow

For more information about assigning roles, see Assign and View EFA Roles on page 42. For more information about supported roles, see EFA RBAC Policy Enforcement on page 39.

EFA RBAC Policy Enforcement

EFA implements an RBAC (Role-based Access Control) policy governing access to northbound REST APIs.

The RBAC policy is enforced at the northbound interface, immediately after validation of the access token. An error message is returned if an RBAC permissions check fails.

RBAC and REST URI matrix

The RBAC policy is expressed in a permissions matrix indexed by RBAC role and REST URI, in which each matrix element enumerates the permitted HTTP methods.

Table 10: RBAC and REST matrix

	Role A	Role B	Role C
REST URI 1	GET	GET	GET, POST, PUT, PATCH, DELETE
REST URI 2	GET, POST	GET, POST, PUT	GET, POST, PUT, PATCH, DELETE
REST URI 3	GET, POST	GET, POST	GET, POST, PUT, PATCH, DELETE

RBAC roles

Roles can be populated into the upstream LDAP instance.

Table 11: Role definitions

Role	Description
FabricAdmin	• Registers devices to the Fabric • Configures Fabric parameters • Validates all devices in the Fabric • Configures switches for IP Fabric with overlay and without overlay • Creates tenants • Creates networks inside tenants, such as VRF, EPG, and PO • Performs Fabric debug activities • Has privileges for OpenStack, Hyper-V, and vCenter operations
SecurityAdmin	Performs user management, PKI, and key management operations
NetworkOperator	• Has view-only privileges for Fabric configurations, information for tenants and inventory, and all ecosystem information • Cannot make changes in the system
SystemDebugger	• Has privileges to perform supportsave and system backup, and to view the running system configurations • Has privileges to perform Fabric debug operations • Sets debug levels for services • Has privileges to collect execution logs from services
SystemAdmin	Has complete privileges to all operations in the system
<Tenant>Admin * Created dynamically per tenant	Performs tenant administration within the assigned tenant, such as the following: • Adding networks to the tenant • Configuring network parameters • Configuring switches with tenant-specific information Cannot perform actions for any other tenant

* Tenant Administrator roles are added dynamically to the system when a tenant is created. The name of the role is of the format `<Tenant-name>Admin`. For example, if a tenant with the name “RegionOne” is created, the role created for the Tenant Administrator is “RegionOneAdmin”.

Note
You cannot create custom roles.

Role permissions

Allowed Privileges	System Admin	Fabric Admin	Tenant Admin	Network Operator	Security Admin	System Debugger
Create/Clone/Delete Fabric in the system	✔	✔				
Register/Unregister devices in Fabric, Configure IP Fabric on the device	✔	✔				
Show IP Fabric physical/underlay/overlay topology, IP fabric configs and devices in IP Fabric	✔	✔		✔		
Debug Fabric operations	✔	✔				✔
Inventory/Asset service operations	✔	✔				
Execute CLI access on the device	✔	✔				
Create/Delete/Update tenants	✔	✔				
Create/Delete EPG, PO, VRFs inside tenant	✔	✔	✔			
Add/Remove Port/Port Channels to/from EPG	✔	✔	✔			
Add/Remove Network Policies to EPG	✔	✔	✔			
Detach Network from EPG	✔	✔	✔			
Identify drift in device configuration	✔	✔				
Set tenant debug level	✔	✔	✔			✔
Show OpenStack networks, PO, subnets, tenant, ports, router, router-interface	✔	✔	✔	✔		
Create/Delete/Cleanup OpenStack Networks	✔	✔	✔			
Create/Delete OpenStack Subnets	✔	✔	✔			
Create/Delete OpenStack Ports	✔	✔	✔			
Create/Delete OpenStack Router	✔	✔	✔			
Create/Delete Router Interfaces	✔	✔	✔			
Delete OpenStack asset (DebugDeleteOSSAsset)	✔	✔	✔			✔
View vCenter Details, events, ESXI details, Physical links, Virtual Links, Disconnected links, Get server settings etc	✔	✔	✔	✔		
Register/Delete/Update vCenter	✔	✔	✔			
Set vCenter debug level	✔	✔	✔			✔

Update vCenter polling frequency, dead link clearing time	✔	✔	✔			
View SCVMM server details, Service Settings, Physical Links, Virtual Links	✔	✔	✔	✔		
Register/Delete/Update SCVMM server	✔	✔	✔			
Update SCVMM server polling frequency	✔	✔	✔			
User Management, assign roles to users, configure LDAP, view available roles in the system	✔				✔	
Notification service (Add/Delete subscribers)	✔	✔				
Execution Log View	✔	✔	✔ (Only Tenant)	✔	✔ (Only Auth and RBAC)	✔
Support Save Collection	✔	✔	✔	✔	✔	✔
Backup and Restore Operation	✔	✔				✔ (Only backup)
Install certificates	✔	✔			✔	

Assign and View EFA Roles

You can assign a role to a user and to an LDAP group.

For more information about EFA roles, see EFA RBAC Policy Enforcement on page 39.

1. To assign a role to a user, run the following command.

```
# efa auth rolemapping add --name fabricuser --role FabricAdmin --type user
Successfully added the role mapping
```

   In this example, a user named fabricuser was assigned the role of FabricAdmin.

2. To assign a role to an LDAP group, run the following command.

```
# efa auth rolemapping add --name viewer --role NetworkOperator --type group
Successfully added the role mapping.
```

   In this example, a group named viewer was assigned the role of NetworkOperator.

3. To view all role assignments, run the following command.

```
# efa auth rolemapping show
ID  Name      Role          Type
1   efauser    SystemAdmin      USER
2   fabricuser FabricAdmin      USER
3   viewer     NetworkOperator  GROUP
```

4. To delete a role assignment, run the following command

```
# efa auth rolemapping remove --id 3
Deleted role mapping successfully
```

   In this example, the role for the user with ID 3 was removed.

Configure an External LDAP Server

You configure an LDAP server for user validation and to fetch user groups.

1. To configure an external LDAP server, run the following command.

   ```
   # efa auth ldapconfig add --name ldapconfig -- host 10.x.x.x --bind-user-
   name cn=admin,dc=extrnet,dc=com --bind-user-password password --user-search-
   base ou=people,dc=extrnet,dc=com
   ```

   This example configures the bind user name and password and the DN of the node from which searches start. For a description of all supported parameters, see the **efa auth ldapconfig** command in the *Extreme Fabric Automation Command Reference, 2.2.0*.

2. To configure an LDAP server in a TPVM (for the TPVM Ubuntu OS), run the **tpvm config ldap** command from the SLX-OS command line.

   For a description of all supported parameters, see the command in the *Extreme SLX-OS Command Reference*.

EFA Certificate Management

The HTTPS server certificate from EFA is presented to a client when that client connects to its northbound interface.

Overview

The certificate is bundled with EFA and signed by the private Certificate Authority (CA) Chain. So that the certificate can be replaced with a third-party certificate acquired through trusted CAs (such as Verisign or GoDaddy), the certificate must be present in the host device that is running EFA. You can then install it with the following command:

```
$ efa certificates server --certificate <cert-filename>
--key <key-filename> [ --configfile  <config-filename ]
```

The `EFA_INSTALL_DIR` environment variable specifies where the EFA configuration file can be found. The optional configuration file can be used to specify a different file than the `efa.conf` file used by EFA for its settings.

Important
If you install your own server certificate to use with the EFA HTTPS server, remember to reinstall the certificate when you upgrade EFA.

Device configuration and certificates

During the registration of an SLX device in EFA, the following configuration changes are made on the device.

- The public certificate for verifying an EFA token is copied to the device as an OAuth2 certificate.
- EFA generates the HTTPS certificate for the SLX device. The certificate is copied to the device, HTTP mode is disabled on the device, and HTTPS is enabled on the device.
- OAuth2 is enabled as the primary mode of authentication. Fallback is set to "local login."

You can use the **efa inventory device list** command to verify the status of the certificates on the device. If the Cert/Key Saved column contains "N," then certificates are not installed.

You can use the **efa certificates device install --ips <ip-adddr> certType [http|token]** command to install the HTTPS or OAuth2 certificate on one or more devices.

Sample certificate contents

Example for a single-node deployment:

```
Subject: CN=efa.extremenetworks.com
         ……
            X509v3 Subject Alternative Name:
                DNS:efa.extremenetworks.com, IP Address:127.0.0.1,
IP Address:10.24.15.173
```

Example for a two-node deployment:

```

  Subject: CN=efa.extremenetworks.com
         ……
            X509v3 Subject Alternative Name:
                DNS:efa.extremenetworks.com, IP Address:127.0.0.1, IP Address:
10.24.15.178,
IP Address:10.24.15.174, IP Address:10.24.15.253
```

Certificate troubleshooting

Issue	Resolution
My device is registered but the certificates do not appear on the SLX device.	Try the following: • Ensure that the device is running at least SLX-OS 20.1.x. • Ensure that the time on the SLX device and the time on the EFA host device are synchronized. • Ensure that the certificates are installed. Run the **efa certificates device install** command.
How do I check the certificate provided by EFA through its ingress interface?	Run the following command. The output should indicate that `efa.extremenetworks.com` is present. `$ openssl s_client -connect <EFA_IP_ADDR>:443`

Back up and Restore the EFA Database

You can back up and restore the EFA database and services, such as goFabric-service and goAuth-service.

You can restore a backed-up database for various reasons, such as if the database becomes corrupted or you want to revert to a previous configuration. The backup process creates a backup tar file, which you specify for the restore process.

1. To back up the database, run the following command as a root user with administrative privileges.

   When prompted, provide the directory path for the backup tar file.

   ```
   # efa_backup

   Provide your backup absolute directory path: /root/backup/bk_v1
   ```

```
backing "dcapp_asset" database to /root/backup directory
backing "dcapp_fabric" database to /root/backup directory
backing "dcapp_tenant" database to /root/backup directory
backing "dcapp_vcenter" database to /root/backup directory
backing "dcapp_hyperv" database to /root/backup directory
backing "efa_openstack" database to /root/backup directory
root@administrator-00:~#
```

2. To restore the database, run the following command as a root user with administrative privileges.

   When prompted, provide the directory path for EFA and for the backup tar file.

   ```
   # efa_restore
   EFA installed on single-node
   Provide your EFA directory path: /root/efa
   Provide database backup tar file: /root/backup/bk_v1
   Provided backup directory...
   stopping service gofabric-service
   deployment.apps "gofabric-service" deleted
   stopping service goauth-service
   deployment.apps "goauth-service" deleted
   stopping service gorbac-service
   deployment.aps/filebeat-service created
   …
   ...
   ...
   configmap/kibana-config unchanged
   service/kibana-service unchanged
   deployment.apps/kibana-service created
   configmap/logstash-service unchanged
   configmap/logstash-config unchanged
   configmap/logstash-pipeline unchanged
   service/logstash-service unchanged
   deployment.apps/logstash-service created
   configmap/metricbeat-config unchanged
   deployment.apps/metricbeat-service created
   Completed deploying elk stack
   ~/backup/bk_v1
   Successfully restored application database
   ```

   After the database is successfully restored, a list of all PODs is displayed, showing their status, number of restarts, and age.

Display EFA Running Configurations

You can view the running-config of all current EFA configurations for core services.

The output is displayed in the following order: Asset, Fabric, Tenant commands. The command output contains the default values for each configuration line item.

You can use the command output for CLI playback on an empty EFA deployment, which is a useful tool for recovery.

Run the **`efa show-running-config`** command.

```
$ efa show-running-config

efa inventory device register --ip "10.24.80.191" --username admin --password password

efa inventory device setting update --ip "10.24.80.191" --maint-mode-enable-on-reboot  No
--maint-mode-enable No --health-check-enable No --health-check-interval 6m
--health-check-heartbeat-miss-threshold 2 --config-backup-periodic-enable Yes
--config-backup-interval 24h --number-of-config-backups 4
```

```
efa inventory device register --ip "10.24.80.192" --username admin --password password

efa inventory device setting update --ip "10.24.80.192" --maint-mode-enable-on-reboot  No
--maint-mode-enable No --health-check-enable No --health-check-interval 6m
--health-check-heartbeat-miss-threshold 2 --config-backup-periodic-enable Yes
--config-backup-interval 24h --number-of-config-backups 4

efa fabric create --name "default" --type clos --stage 3 --description "Default Fabric"
```

This example shows only a partial list of typical output.

Audit Trail Logging

EFA provides full audit trail logging, including the successes and failures of user actions, which creates a 1-to-1 mapping between every action coming from EFA and a corresponding audit trail event from SLX.

Any configuration action on an SLX devices results in the generation of an audit trail. The name of the user is extracted from the token that the user logged in with. The user is assigned the role of `admin` as the default role on the device.

For OpenStack, the user name has the following format: `<OpenStack tenant UUID> - <OpenStack user name> - <EFA tenant name>`.

The following is an example of the audit log message for NETCONF or SSH sessions:

```
78 AUDIT, 2020/01/26-14:04:21 (GMT), [DCM-1006], INFO, DCMCFG, <ClientUserID>/
<ClientRole>/10.6.46.51/SSH/netconf,, SLX, Event: database commit transaction, Status:
Succeeded, User command: "configure config username test1 role admin password ****".
```

The `ClientUserID` and `ClientRole` values are derived from the `User` and `AuditLogRole` variables, which originate from the values in the access token when the NETCONF or SSH session was established.

Transfer of Audit Trail Data

Audit trail data from SLX devices is transferred to EFA for delivery upstream using JSON structured data.

The data is transferred to an upstream web server at a predefined URL that is registered with EFA.

Incoming syslog messages from SLX to EFA are converted by a logging service on EFA into JSON data, as in the following example:

```
"message_id": "9999",
  "message": "Hello world",
  "source_ip": "192.168.10.1",
  "user": "admin",
  "severity": "INFO",
  "timestamp", "2020-02-11 19:23:58.383304",
  "extra_data": {}
}
```

EFA sends the messages by POST requests to an upstream web receiver.

Elasticsearch, Logstash, Kibana Integration

In the EFA ecosystem, ELK (Elasticsearch, Logstash, Kibana) is implemented in the same network as the Application stack.

URLs to access the ELK stack

Elasticsearch: `http://<host_ip>:30920`

Kibana: `http://<host_ip>:30601`

Sample log

```
@timestamp:December 13th 2018, 22:18:12.929 source
:/var/log/dcapp/fabric/fabric.log offset:513,560 message:{"level":"info","msg":
"Fabric service Health status OK ","time":"2018-12-12T18:03:04Z"} prospector.type:log
json.level:info json.msg:Fabric service Health status OK json.time:2018-12-12T18:03:04Z
beat.name:5d2a1a83ed27 beat.hostname:5d2a1a83ed27 beat.version:6.2.2 _id:
YdN4qGcBzheJSFbXB7U5 _type:doc _index:filebeat-6.2.2-2018.12.13 _score:1
```

Table 12: Log tags

Tag	Description
source	Provides the information about which service the log belongs to.
level	Provides the level of log, for example, whether a log is “Error” or “Info” or “Warning”.
_id	Each log is numbered with a unique ID.
json.msg	Contains details about the operation or error message in this field.
timestamp	Details about when the operation was performed. Gives exact time of log creation.

Infra level

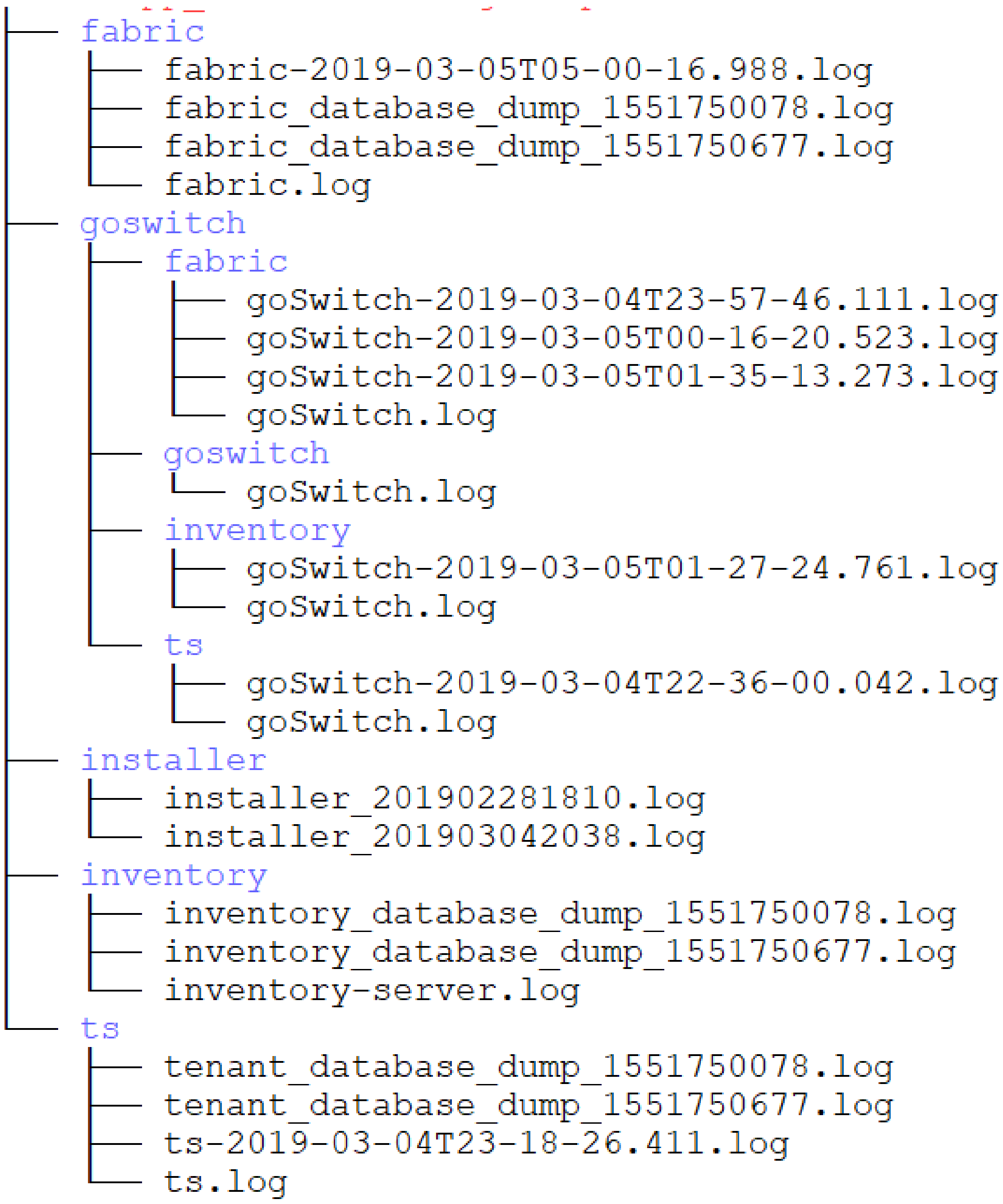
```
# docker logs k3s
```

To obtain a *<container-id>*, run **`docker ps`**.

The ELK stack is deployed as part of the deployment, which helps analyze the application-specific logs. Logs for the services are available in the host at `/var/log/efa`.

Application level

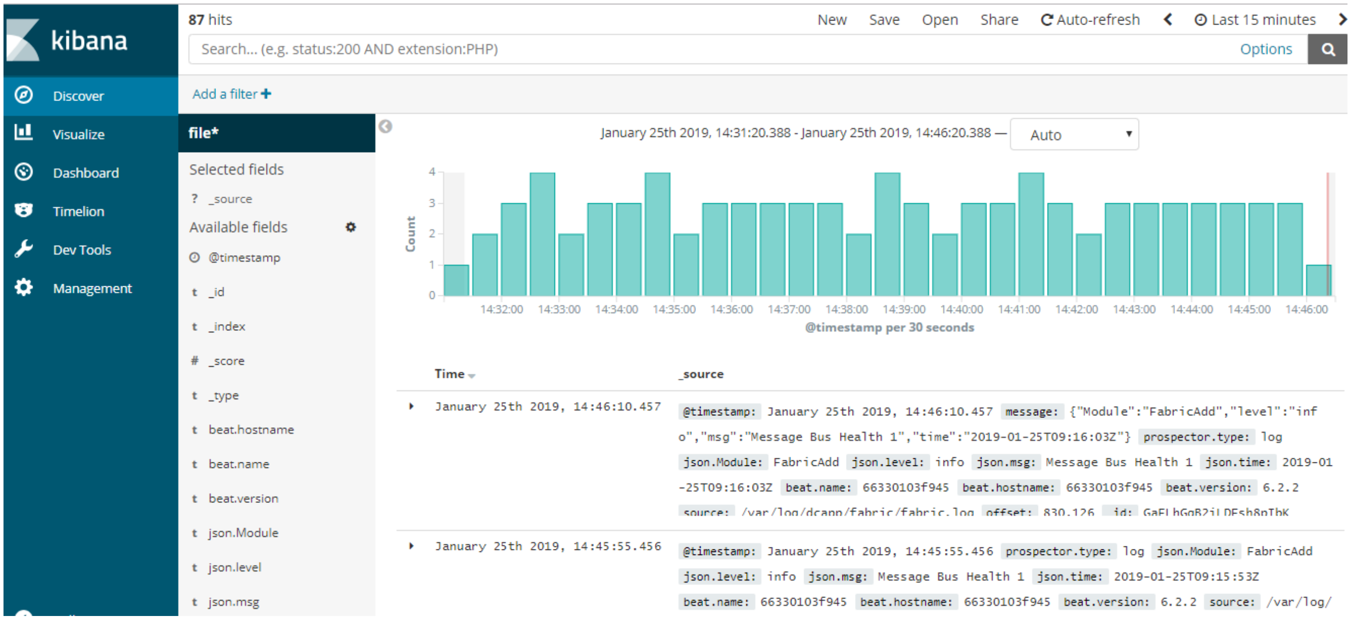
The ELK stack helps analyze the application-specific logs. Logs for the services are available in the host at `/apps/efa_logs`.



```
├── fabric
│   ├── fabric-2019-03-05T05-00-16.988.log
│   ├── fabric_database_dump_1551750078.log
│   ├── fabric_database_dump_1551750677.log
│   └── fabric.log
├── goswitch
│   ├── fabric
│   │   ├── goSwitch-2019-03-04T23-57-46.111.log
│   │   ├── goSwitch-2019-03-05T00-16-20.523.log
│   │   ├── goSwitch-2019-03-05T01-35-13.273.log
│   │   └── goSwitch.log
│   ├── goswitch
│   │   └── goSwitch.log
│   ├── inventory
│   │   ├── goSwitch-2019-03-05T01-27-24.761.log
│   │   └── goSwitch.log
│   └── ts
│       ├── goSwitch-2019-03-04T22-36-00.042.log
│       └── goSwitch.log
├── installer
│   ├── installer_201902281810.log
│   └── installer_201903042038.log
├── inventory
│   ├── inventory_database_dump_1551750078.log
│   ├── inventory_database_dump_1551750677.log
│   └── inventory-server.log
└── ts
    ├── tenant_database_dump_1551750078.log
    ├── tenant_database_dump_1551750677.log
    ├── ts-2019-03-04T23-18-26.411.log
    └── ts.log
```

Figure 5: Application level log

Logs are visualized on a Kibana dashboard. The following is an example.



```
January 25th 2019, 14:46:10.457   @timestamp: January 25th 2019, 14:46:10.457 message: {"Module":"FabricAdd","level":"inf
                                  o","msg":"Message Bus Health 1","time":"2019-01-25T09:16:03Z"} prospector.type: log
                                  json.Module: FabricAdd json.level: info json.msg: Message Bus Health 1 json.time: 2019-01
                                  -25T09:16:03Z beat.name: 66330103f945 beat.hostname: 66330103f945 beat.version: 6.2.2
                                  source: /var/log/dcapp/fabric/fabric.log offset: 830,126 _id: GaFLhGqB2iLDEsh8pTbK

January 25th 2019, 14:45:55.456   @timestamp: January 25th 2019, 14:45:55.456 prospector.type: log json.Module: FabricAdd
                                  json.level: info json.msg: Message Bus Health 1 json.time: 2019-01-25T09:15:53Z
                                  beat.name: 66330103f945 beat.hostname: 66330103f945 beat.version: 6.2.2 source: /var/log/
```

Figure 6: Kibana discover

Logging and Log Files

Log directories vary by deployment type and by application.

Logs are saved to the following locations:

- Non-TPVM EFA deployments: `/var/log/efa`. The installation logs in the `/var/log/efa/installer` directory are a good source for discovering the reason for a failure.
- TPVM EFA deployments: `/apps/efa_logs`
- Kubernetes log files: `/var/log/pods`

The **`efa supportsave`** script gathers all logs, database dumps, pod logs, and deployment details and then compresses them into a ZIP folder. You can share this ZIP folder with Extreme support personnel when troubleshooting an issue.

Data Consistency

EFA ensures that SLX devices have the correct configuration before allowing traffic.

Overview

EFA is the data owner and Single Source of Truth (SSOT) for Fabric configuration.

The following figure shows an overview of how data is rendered consistent among EFA services.

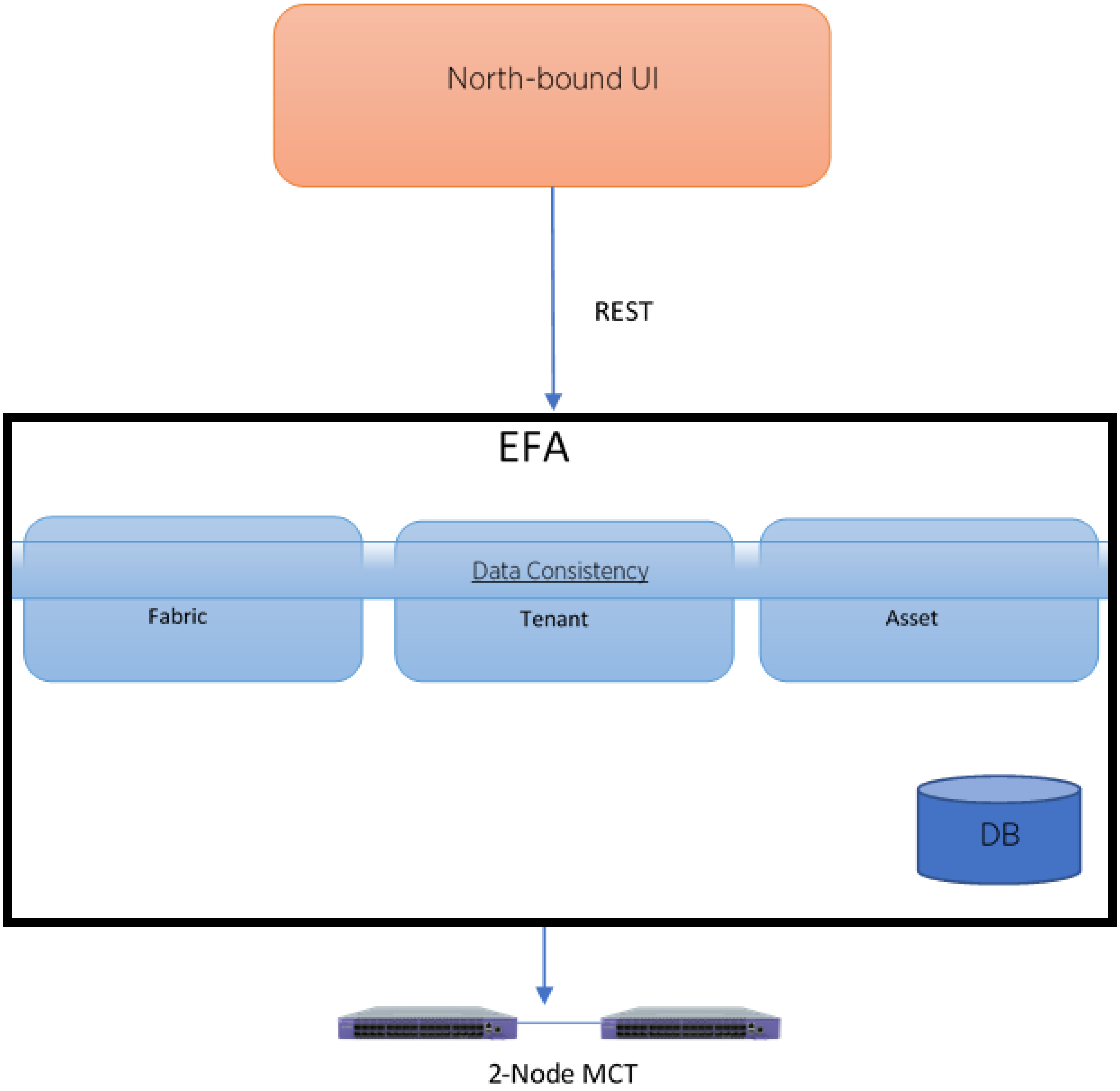
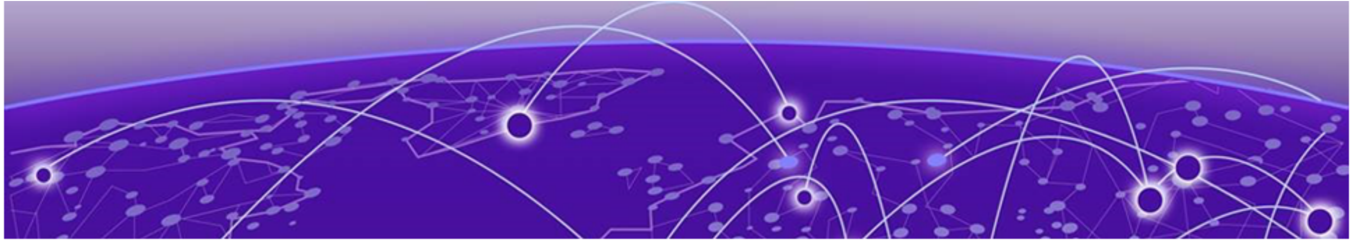


Figure 7: Data consistency overview

North-bound applications request the REST APIs to perform various operations on EFA. EFA ensures that the operations leave EFA and the Fabric in a consistent state.

Limitations

- You cannot use the SLX CLI to configure the entities that are managed by EFA.
- EFA can reconcile only those entities or configurations that it manages.
- EFA cannot modify out-of-band entities or configurations unless they conflict with the configurations that it manages.



Fabric Infrastructure Provisioning

Fabric Service Overview

Fabric Service is responsible for automating the Fabric BGP underlay and EVPN overlay. By default, the EVPN overlay is enabled but can be disabled before provisioning if desired. Fabric Service exposes the CLI and REST API to clients for automating the Fabric underlay and overlay configuration.

Fabric Service features include:

- Small Data Center Topology (non-Clos support)
- Support for 3- and 5-stage Clos Fabrics
- Support for MCT configuration
- Support for Eco-System Integration; Openstack, VMWare vCenter, Microsoft Hyper-V/SCVMM

Underlay automation includes Interface Configurations (IP Numbered), BGP Underlay for Spine and Leaf, BFD, and MCT configurations. Overlay automation includes EVPN and Overlay Gateway configuration. Fabric Service is deployed along with Inventory Service and Tenant Service.

IP Fabric and Clos Orchestration Overview

A Fabric is a logical container for holding a group of devices. Here it denotes a collection of devices that are connected in a Fabric topology and on which you can configure underlay and overlay.

Fabric service provides following features:

- 3-stage Clos automation
- 5-stage Clos automation
- Small Data Center automation
- Multi-Fabric automation
- Fabric topology view

- Fabric validation, error reporting, and recovery
- Single-homed leaf or multi-homed (MCT) leaf

Fabric CLIs and REST APIs provide the following:

- Mechanism to create a Fabric composed of multiple DC points of delivery (PoDs).
- Mechanism to configure Fabric settings. Fabric settings are collections of settings that control the various parameters of the Fabric being managed, for example, Layer 2 and Layer 3 MTU, and BGP maximum paths.
- Mechanism to fetch per-device errors occurring during Fabric configuration, for which you can take corrective or remedial actions.

Errors occurring on the device during Fabric creation are tagged against the devices and can be retrieved from the CLI and REST APIs for use in taking corrective or remedial actions.

SLX Device Prerequisites for Fabric Service

The following items are required before configuring Fabric Automation.

- Management IP addresses must be configured on all switches.
- SLX devices must have the appropriate firmware version. Refer to the Supported Platform Matrix.
- SLX 9850: Fabric links must be enabled manually , through `no shut`.
- SLX 9540: The appropriate TCAM profile must be set and the switch rebooted.

```
device# conf
Entering configuration mode terminal
device(config)# hardware
device(config-hardware)# profile tcam vxlan-ext
%Warning: To activate the new profile config, run 'copy running-config startup-config'
followed by 'reload system'.
device(config-hardware)#
```

- Any breakout ports on SLX devices must be configured manually.
- Refer to the release specific *Extreme SLX-OS Management Configuration Guide* for configuration steps for each platform.

Clos Overview

Extreme Fabric Automation (EFA) is a microservices-based application that manages the life cycle of IP Fabric Clos and Small Data Center deployments. All of the microservices support REST APIs that are detailed by OpenAPI.

EFA offers unique flexibility in supporting multiple IP Fabric topologies based on a BGP underlay with a BGP or EVPN overlay:

- Small Data Center Fabric (non-Clos topology from a single switch pair up to four switch pairs)
- 3-stage Clos (Leaf / Spine)
- 5-stage Clos (Leaf / Spine / Super Spine)

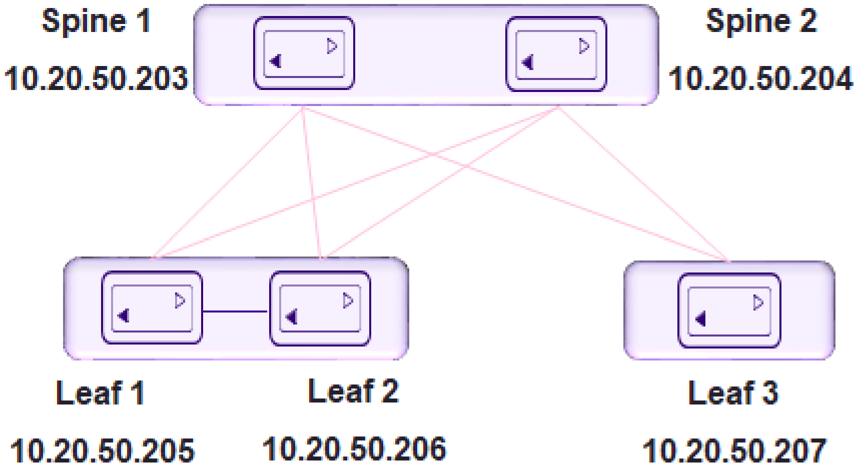
Tenant Network onboarding services are supported on all the topologies, allowing you to create connectivity for devices connected to the fabric, such as compute (servers), storage, and connectivity to external routers or gateways.

Configure 3-Stage Clos Automation

The 3-stage topology has two layers of devices: Leaf and Spine. All links between Leaf and Spine must be connected. Spine nodes should not be interconnected.

There are four steps to configuring a three-stage topology: create the fabric, register the devices, validate and add the devices, and provision the configuration on the devices.

The following is an example of a three-stage fabric.



1. Create the Fabric.
2. Add a device or devices to the Fabric.
3. Validate the Fabric Topology.

   During the addition of a device to a Fabric and during Fabric configuration, Clos topology validations are performed. If the validation contains errors, the errors are reported to the user as a response to the **`fabric device add`** or **`fabric configure`** operations. Any Fabric topology errors can be exported to a CSV or DOT file. The following topology validations are performed:

   - Leaf nodes must connect to all the Spine nodes.
   - A Spine node must connect to all the leaf nodes.
   - A Border Leaf node connects to all the Spine nodes.
   - A Spine node connects to all the Border Leaf .
   - No more than two Leaf nodes connect to each other.
   - No more than two Border Leaf nodes connect to each other.
   - Border Leaf node and L node are not connected to each other.
   - Spine nodes are not connected to each other.
   - Super-spine nodes are not connected to each other.
   - If a Leaf node is marked as "multi-homed", then the node must have an MCT neighbor.
   - If a Leaf node is marked as "single-homed", then the node is not connected to other Leaf nodes.
   - If a Border Leaf node is marked as "multi-homed", then the node must have an MCT neighbor.
   - If a Border Leaf node is marked as "single-homed", then the node is not connected to other Border Leaf nodes.
   - Device role (such as Leaf, Border-leaf, Spine, or Super-spine) is validated for a given device platform type (for example, SLX 9840 cannot be added as a leaf).

4. Configure the Fabric on the devices.

Example: Create the fabric.

```
efa fabric create --name stage3
```

Example: Add a device to the fabric.

```
efa fabric device add-bulk --leaf 10.20.50.205,10.20.50.206,10.20.50.207 --spine
10.20.50.203,10.20.50.204
--name stage3 --username admin --password password
```

Example: Configure the fabric on the device.

```
efa fabric configure --name stage3
```

Configure 5-Stage Clos Automation

The five-stage topology has three layers of devices: Leaf, Spine, and Super Spine. All links between Leaf and Spine must be connected. Spine nodes should not be interconnected. Similarly, all the links between the Spine and Super-spine must be connected. A border Leaf can be directly connected to a Super-spine, but there should not be any connection between a border Leaf and a Spine.

There are four steps to configuring a five-stage topology: create the fabric, register the devices, validate and add the devices, and provision the configuration on the devices.

The following image is an example of a five stage fabric. The following steps describe how to configure a five-stage fabric topology.

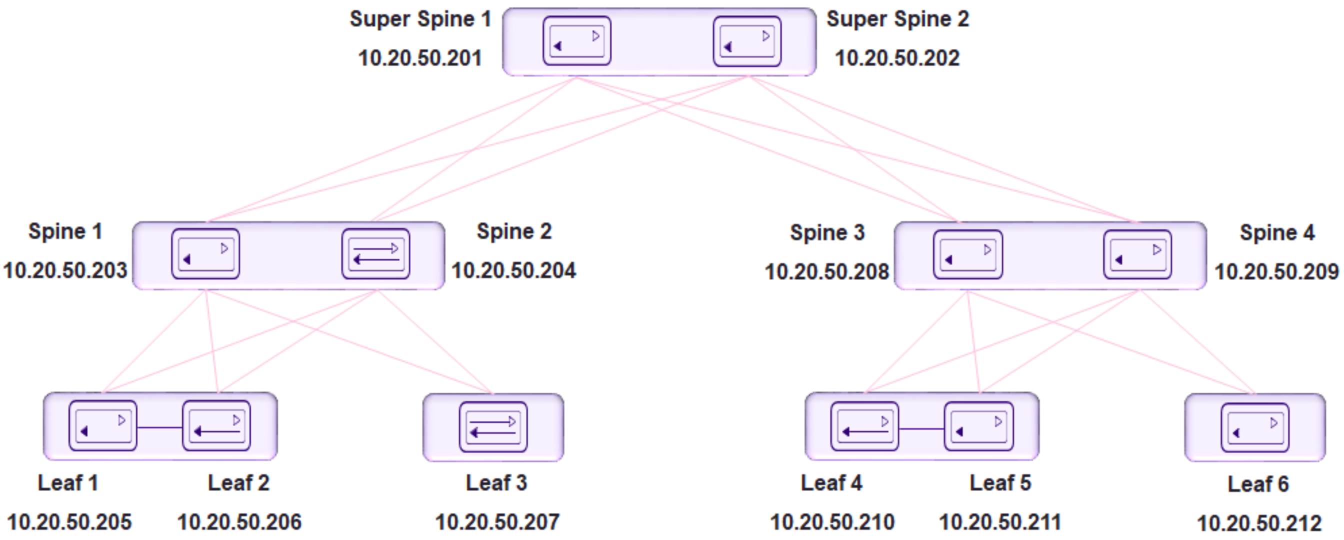


Figure 8: Five-Stage Clos Automation

Note
5-stage Clos can be built top to bottom or bottom to top. The following example builds from the top down.

1. Create the Fabric using the `efa fabric create` command.

2. Add a device or devices to the 5-stage Fabric using the `efa fabric device add`. The user must provide device credentials as part of this command if the devices are not already registered with the inventory.

   A device must be registered with Inventory Service before being added to a Fabric. Fabric Service supports IP numbered configuration. Each interface on a link between Leaf and Spine is assigned an IP address. eBGP peering uses these IP addresses.

   Multiple devices can be added to an existing fabric using the `efa fabric device add-bulk` command. If a username and password are provided, then the devices will be auto registered with the inventory service.

3. Use the `efa fabric configure` command to validate and configure the Fabric Topology.

   During the addition of a device to a Fabric and during Fabric configuration, Clos topology validations are performed. If the validation contains errors, the errors are reported to the user as a response to the **`fabric device add`** or **`fabric configure`** operations. Any Fabric topology errors can be exported to a CSV or DOT file.

   The following topology validations are performed:

   - Leaf nodes must connect to all the Spine nodes.
   - A Spine node must connect to all the Leaf nodes.
   - A Border Leaf node connects to all the Spine nodes.
   - A Spine node connects to all the Border Leaf nodes
   - No more than two Leaf nodes connect to each other.
   - No more than two Border Leaf nodes connect to each other.
   - Border Leaf node and Leaf node are not connected to each other.
   - Spine nodes are not connected to each other.
   - Super-spine nodes are not connected to each other.
   - If a Leaf node is marked as "multi-homed", then the node must have an MCT neighbor.
   - If a Leaf node is marked as "single-homed", then the node is not connected to other Leaf nodes.
   - If a Border Leaf node is marked as "multi-homed", then the node must have an MCT neighbor.
   - If a Border Leaf node is marked as "single-homed", then the node is not connected to other Border Leaf nodes.
   - Device role (such as Leaf, Border-leaf, Spine, and Super-spine) is validated for a given device platform type (for example, SLX 9840 cannot be added as a leaf).

Example: Create the fabric.

```
efa fabric create --name stage5
```

Add a device to the 5-stage fabric.

```
efa fabric device add--name stage5 --username admin --password password --leaf
10.20.50.205,10.20.50.206,10.20.50.207
--spine 10.20.50.203,10.20.50.204 --three-stage-pod podA --super-spine
```

Example: Add multiple devices to the 5-stage fabric.

```
efa fabric device add-bulk --name stage5 --username admin --password password --leaf
10.20.50.205,10.20.50.206,10.20.50.207
--spine 10.20.50.203,10.20.50.204 --three-stage-pod podA --super-spine
10.20.50.201,10.20.50.202 --five-stage-pod podC
```

Example: Validate and configure the fabric toplogy.

```
efa fabric configure --name stage5
```

Fabric Topology View

Fabric topology view displays the physical topology (LLDP neighbors), underlay topology (BGP neighbors) and the overlay topology (VxLAN tunnels) established between the nodes of the Fabric.

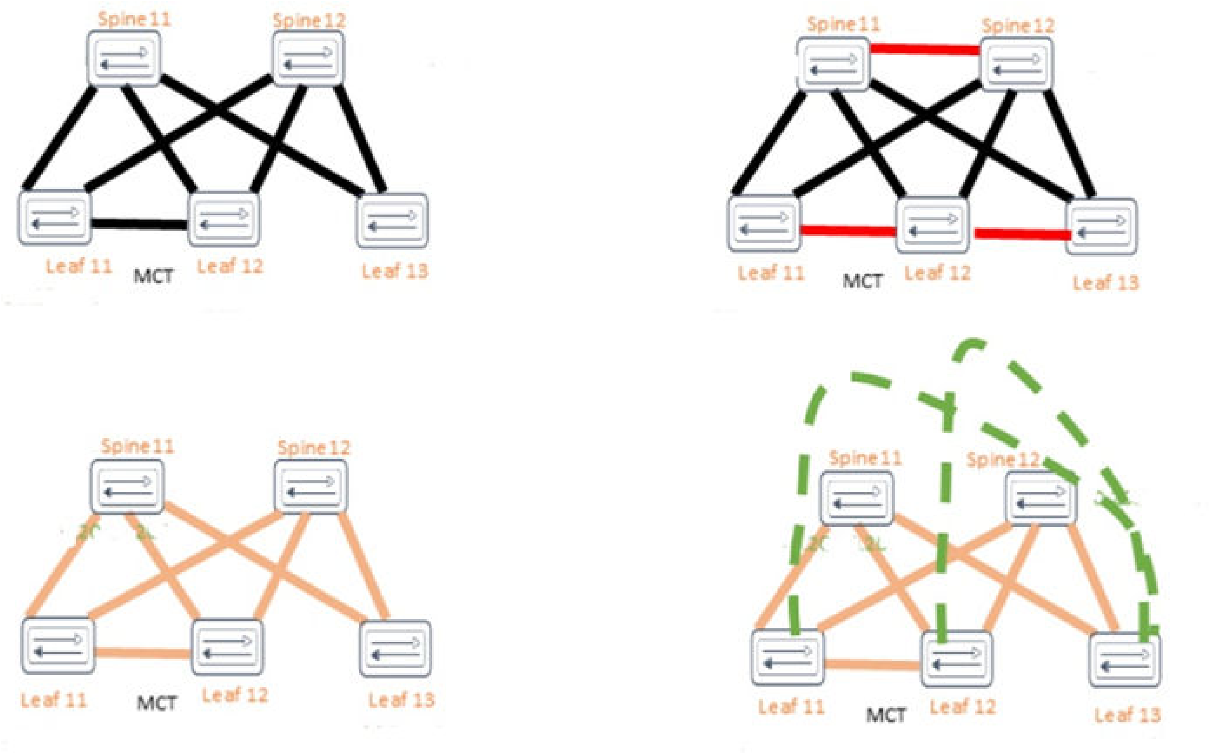


Figure 9: 3-stage Clos Topology View

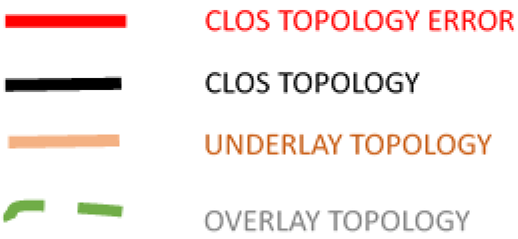


Figure 10: 3-stage Clos Topology View Key

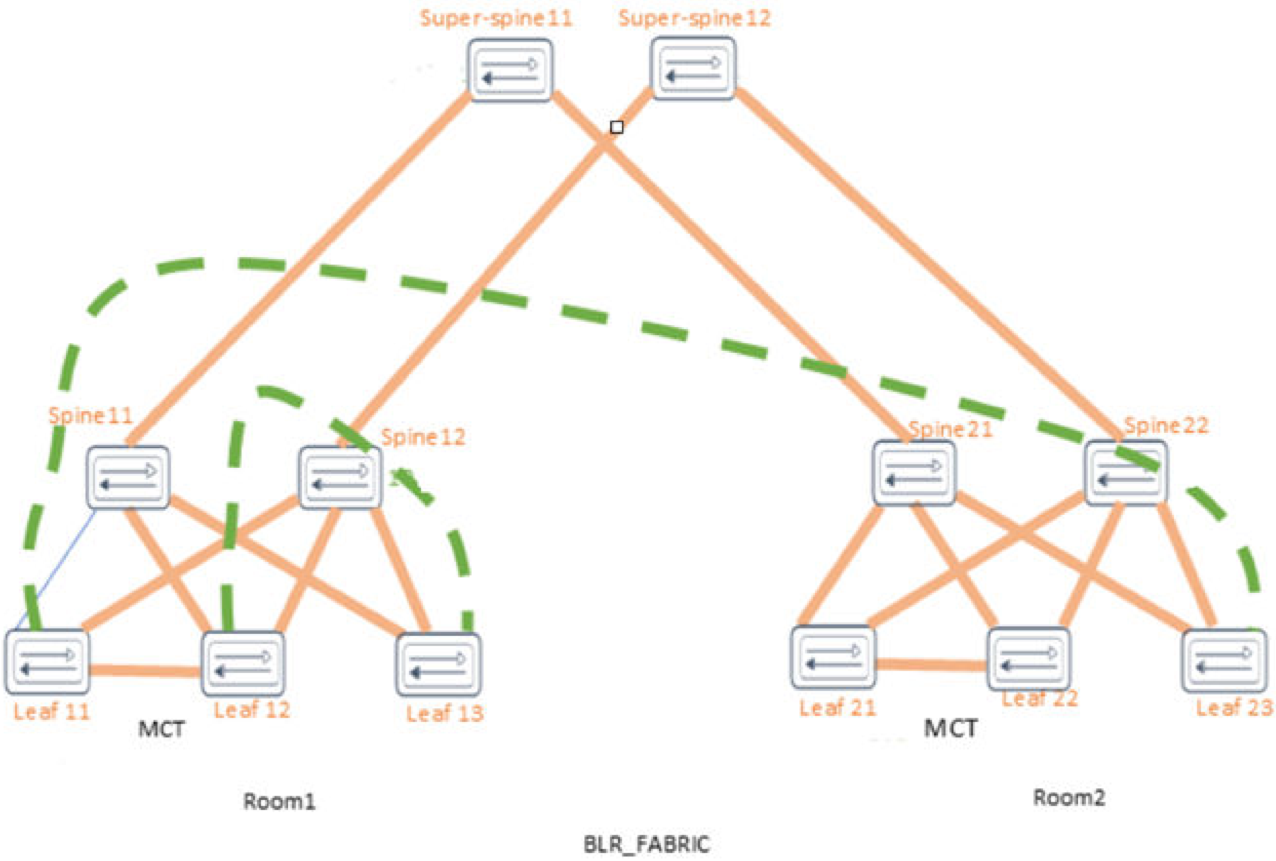


Figure 11: 5-stage Clos Topology View

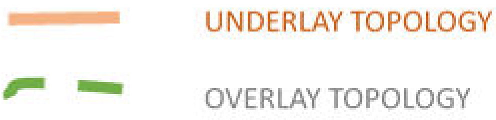


Figure 12: 5-stage Clos Topology View Key

Non-Clos Small Data Center Overview

Support for Small DC Fabric offers CLI commands along with a REST API, similar to that of Clos Fabric.

Non-Clos fabric is supported on SLX 9150, SLX 9140, and SLX 9250 devices as follows:

- Single rack automation. Each rack consists of two node MCT pair.
- Multi-rack automation
- Multi-homed leaf (MCT)
- Overlay only automation
- Fabric topology view
- Fabric validation and troubleshooting

Supported Non-Clos Topologies

Small data centers are supported on SLX 9140, SLX 9150, and SLX 9250.

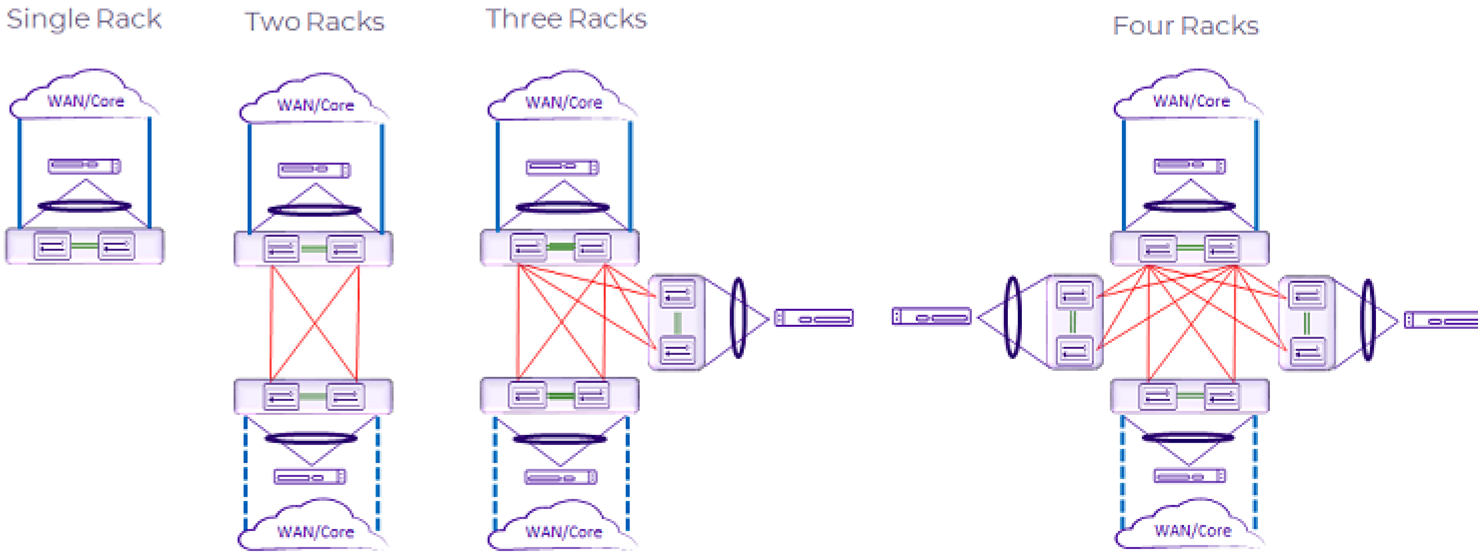


Figure 13: Supported small data center topologies

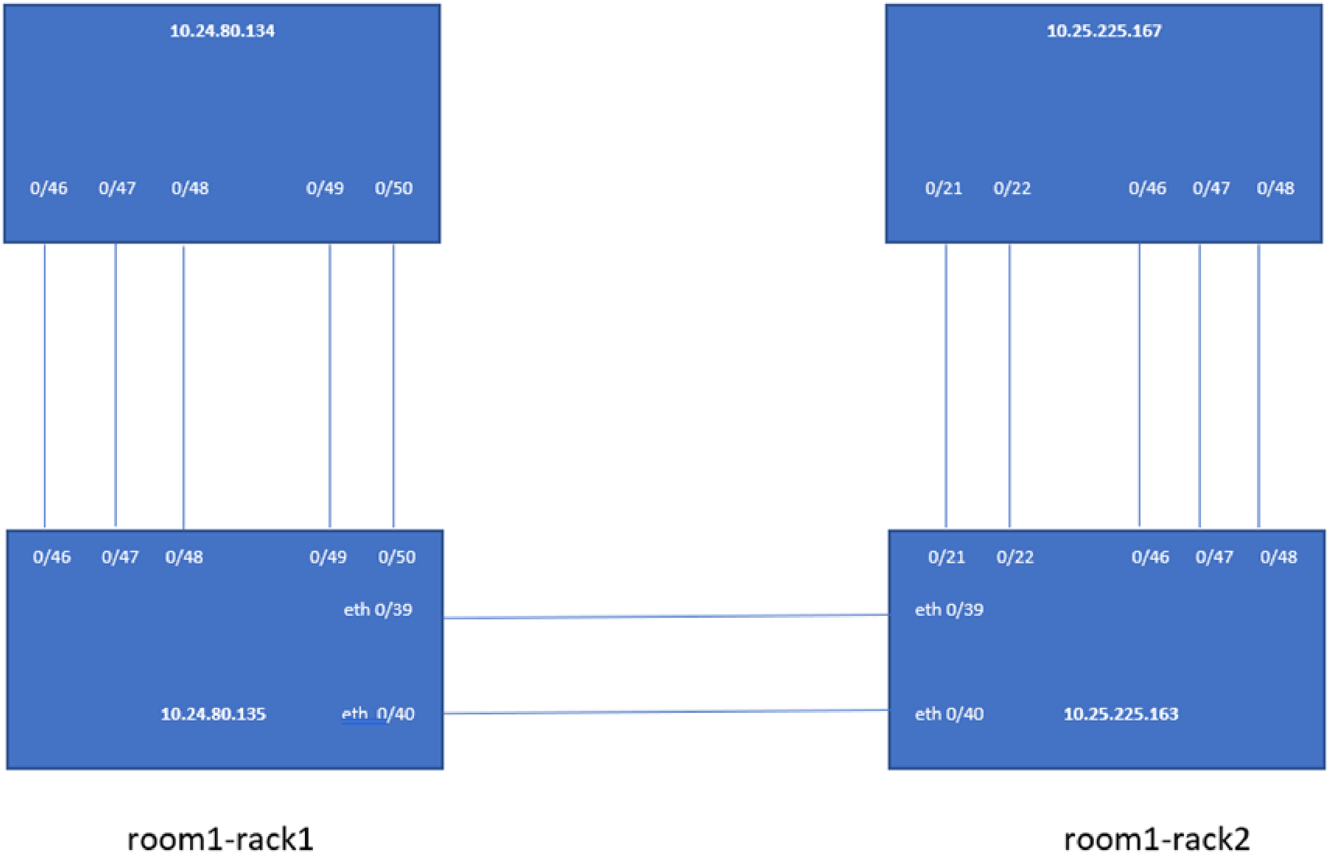


Figure 14: Non-Clos multi-rack config automation

Configuration and Topology Validations

During the addition of a device to a fabric and during fabric configuration, Non-Clos configuration and topology validations are performed. If the validation errors out, the same is reported for taking the appropriate corrective action. The encountered fabric errors can be exported to a .csv or .dot file.

Configuring Non-Clos Small Data Center Automation

There are four steps to configuring a Non-Clos Small Data Center topology; Create the fabric, Register the devices, Validate and Add the devices, and Provision the configuration on the devices. The following steps describe how to configure a small data center topology.

1. Create the Fabric.

   ```
   $ efa fabric create --name extr-fabric  --type non-clos
   ```

2. Add a device or devices to the Fabric. The user must provide device credentials as part of this command if the devices are not already registered with the inventory.

   ```
   $ efa fabric device add --name extr-fabric --ip 10.24.80.134 --rack room1-rack1 --
   username admin --password password
   $ efa fabric device add --name extr-fabric --ip 10.24.80.135 --rack room1-rack1 --
   username admin --password password

   $ efa fabric device add --name extr-fabric --ip 10.25.225.163 --rack room1-rack2 --
   username admin --password password
   $ efa fabric device add --name extr-fabric --ip 10.25.225.167 --rack room1-rack2 --
   username admin --password password
   ```

   A device must be registered with Inventory Service before being added to a Fabric. Fabric Service supports IP numbered configuration. Each interface on a link between leaf and spine is assigned an IP address. eBGP peering use these IP addresses.

   Multiple devices can be added to an existing fabric using the `efa fabric device add-bulk` command. If a username and password are provided, then the devices will be auto registered with the inventory service.

   ```
   $ efa fabric device add-bulk --name extr-fabric --rack room1-rack1 --ip
   10.24.80.134,10.24.80.135 --rack room1-rack2 --ip 10.25.225.163,10.25.225.167
   ```

3. Use the `efa fabric configure` command to validate and configure the Fabric Topology.

   ```
   $ efa fabric configure --name extr-fabric
   ```

   During each of the steps; `create, add,` and `configure`, Clos topology validations are performed. If the validation contains errors, the errors are reported to the user. The encountered Fabric topology errors can be exported to a CSV or DOT file.

   If the addition of devices to a Fabric is successful, then the configuration is pushed to all the devices of the Fabric using the `efa fabric configure` command above.

Use the following commands to display detailed information about the Small Data Center topology and configuration.

- `efa fabric export`: Export fabric details to a .CSV file.
- `efa fabric show`: Display the details of the fabric.
- `efa fabric topology show physical`: Displays physical connectivity of the devices in a fabric.

- `efa fabric topology show underlay`: Displays the underlay connectivity - the BGP neighborship and the state of BGP sessions of the devices in a fabric.
- `efa fabric topology show overlay`: Displays the overlay connectivity of the devices in a fabric.

Overview of Day-0 Operations

Day-0 operations consist of forming the Fabric.

This table provides examples of the commands that you use to create a non-Clos Fabric with two SLX devices. For more information about command parameters, see the command topic in the *Extreme Fabric Automation Command Reference*.

Table 13: Day-0 operations

Operation	Command
Create a Fabric	**`efa fabric create --name CNCF type non-clos`**
Select MTC ports	**`efa fabric setting update --rack-ld-mct-ports '0/29,0/30,0/31,0/32' --name CNCF`**
Enable backup routing	**`efa fabric setting update --backup-routing-enable Yes --name CNCF`**
Disable VLAN VNI auto-map	**`efa fabric setting update --vni-auto-map No --name CNCF`**
Add the first device	**`efa fabric device add --ip 10.24.80.158 --hostname slx-a --rack pod1 --username admin --password password --name CNCF`**
Add the second device	**`efa fabric device add --ip 10.24.80.159 --hostname slx-b --rack pod1 --username admin --password password --name CNCF`**
Configure the Fabric	**`efa fabric configure --name CNCF`**

Fabric Configuration Reconciliation

The following Fabric configuration reconciliation cases and actions are supported:

- Case:
  - MISSING
  - CONFLICT
- Action:
  - PUSH Parent
  - REMOVE Parent
  - PUSH Child
  - REMOVE Child
  - MERGE

The topics in this section show how various switch configurations are handled during reconciliation by Fabric Services.

EVPN

EFA Config	`evpn fabric-2`
	`route-target both auto ignore-as`
	`rd auto`
	`duplicate-mac-timer 5 max-count 3`
	`!`

Switch Config	Empty
Case	MISSING Parent Config
Action	PUSH Entire Config
Remarks	If the switch does not contain the `evpn config` of the name intended, push the intended `efa config`.

EFA Config	**`evpn fabric-2`**
	`route-target both auto ignore-as`
	`rd auto`
	`duplicate-mac-timer 5 max-count 3`
	`!`

Switch Config	**`evpn fabric-3`**
	`route-target both auto ignore-as`
	`rd auto`
	`duplicate-mac-timer 7 max-count 5`
	`vlan add 100`
	`!`

Case	MISSING Parent Config
Action	REMOVE Parent and PUSH Entire Config
Remarks	If the switch does not contain the intended `evpn config` and contains a different name, remove the existing `evpn config` and push the intended `evpn config`.

EFA Config	`evpn fabric-2`
	`route-target both auto ignore-as`
	`rd auto`
	`duplicate-mac-timer 5 max-count 3`
	`!`

Switch Config	`evpn fabric-2`
	`vlan 200`
	`!`

Case	MISSING Child Config

Action	MERGE Config
Remarks	If the switch contains additional configuration pushed by the tenant, but does not contain all the intended child configurations, merge the configurations.

EFA Config	`evpn fabric-2`
	`route-target both auto ignore-as`
	`rd auto`
	`duplicate-mac-timer 5 max-count 3`
	`!`

Switch Config	`evpn fabric-2`
	`route-target both auto ignore-as`
	`rd auto`
	`duplicate-mac-timer 7 max-count 5`
	`vlan add 100`
	`!`

Case	CONFLICT Child Config
Action	PUSH Child Config
Remarks	If the switch contains conflicting `evpn config`, replace the config with the intended config..

IP MTU

EFA Config	`ip mtu 9100`
Switch Config	Empty
Case	MISSING Config
Action	PUSH Config
Remarks	If the switch configuration does not contain `mtu config`, push the new configuration.

EFA Config	`ip mtu 9100`
Switch Config	`ip mtu 8999`
Case	CONFLICT Config
Action	PUSH Config
Remarks	If the switch configuration contains conflicting config, remove the existing configuration and push the new configuration.

MTU

EFA Config	`mtu 9216`
Switch Config	Empty
Case	MISSING Config

Action	PUSH Config
Remarks	If the switch configuration does not contain `mtu config`, push the new configuration.

EFA Config	`mtu 9216`
Switch Config	`mtu 9316`
Case	CONFLICT Config
Action	PUSH Config
Remarks	If the switch configuration contain conflicting configuration, remove the existing configuration and push the new configuration.

IP Anycast Gateway

EFA Config	`ip anycast-gateway-mac 0201.0101.0101`
Switch Config	Empty
Case	MISSING Config
Action	PUSH Config
Remarks	If the switch does not contain the configuration, push the new configuration.

EFA Config	`ip anycast-gateway-mac 0201.0101.0101`
Switch Config	`ip anycast-gateway-mac 0301.0101.0101`
Case	CONFLICT Config
Action	REMOVE and PUSH Config
Remarks	If the switch contains conflicting configuration, remove the existing configuration and push the new configuration.

IPV6 Anycast Gateway

EFA Config	`ipv6 anycast-gateway-mac 0201.0101.0102`
Switch Config	Empty
Case	MISSING Config

Action	PUSH Config
Remarks	If the switch does not contain the configuration, push the new configuration.

EFA Config	`ipv6 anycast-gateway-mac 0201.0101.0102`
Switch Config	`ipv6 anycast-gateway-mac 0301.0101.0103`
Case	CONFLICT Config
Action	REMOVE and PUSH Config
Remarks	If the switch contains conflicting configuration, remove the existing configuration and push the new configuration.

IP Router ID

EFA Config	`ip router-id 172.31.254.20`
Switch Config	Empty
Case	MISSING Config
Action	PUSH Config
Remarks	If the switch does not contain the configuration, push the new configuration.

EFA Config	`ip router-id 172.31.254.20`
Switch Config	`ip router-id 172.31.254.22`
Case	CONFLICT Config
Action	PUSH Config
Remarks	If the switch contains conflicting configuration, remove the existing configuration and push the new configuration.

MAC Address Table

EFA Config	`mac-address-table learning-mode conversational`
	`mac-address-table aging-time conversational 500`
	`mac-address-table aging-time 3333`
	`mac-address-table mac-move detect`
	`mac-address-table mac-move limit 10`

Switch Config	Empty
Case	MISSING Config

Action	PUSH Config
Remarks	If the switch does not contain the configuration, push the new configuration.

EFA Config	`mac-address-table learning-mode conversational`
	`mac-address-table aging-time conversational 500`
	`mac-address-table aging-time 3333`
	`mac-address-table mac-move detect`
	`mac-address-table mac-move limit 10`

Switch Config	`mac-address-table learning-mode conversational`
	`mac-address-table aging-time conversational 600`
	`mac-address-table aging-time 3333`
	`mac-address-table mac-move detect`
	`mac-address-table mac-move limit 10`

Case	CONFLICT Config
Action	PUSH Config
Remarks	If the switch contains conflicting configuration, remove the existing configuration and push the new configuration.

EFA Config	`mac-address-table learning-mode conversational`
	`mac-address-table aging-time conversational 500`
	`mac-address-table aging-time 3333`
	`mac-address-table mac-move detect`
	`mac-address-table mac-move limit 10`

Switch Config	`mac-address-table learning-mode conversational`
	`mac-address-table aging-time 3333`
	`mac-address-table mac-move detect`
	`mac-address-table mac-move limit 10`

Case	MISSING Config
Action	PUSH Config
Remarks	If the switch does not contain the intended configuration, push the new configuration.

Overlay Gateway

EFA Config	`overlay-gateway fabric-2`
	`map vni auto`
	`activate`
	`!`

Switch Config	Empty
Case	MISSING Config

Action	PUSH Config
Remarks	If the switch does not contain the intended configuration, push the new configuration.

EFA Config	overlay-gateway fabric-2 ip interface Loopback 2 map vni auto activate !
Switch Config	overlay-gateway fabric-2 ip interface Loopback 2 map vlan 2000 vni 5000 map vni auto map bridge-domain 4095 vni 8100 activate !
Case	No Change
Action	No Action
Remarks	If the switch configuration and the intended configuration are same, accept that the switch configuration has additional configuration.

EFA Config	overlay-gateway fabric-2 **ip interface Loopback 2** **map vni auto** **activate** !
Switch Config	overlay-gateway fabric-2 **ip interface Loopback 3** map vlan 2000 vni 5000 map vni auto map bridge-domain 4095 vni 8100 activate !
Case	CONFLICT Child Config
Action	PUSH Child Config
Remarks	If the switch configuration and the intended configuration conflict with the child configuration, remove the existing child configuration and push the new child configuration.

EFA Config	overlay-gateway fabric-2 ip interface Loopback 2 **map vni auto** **activate** !
Switch Config	overlay-gateway fabric-2 ip interface Loopback 2 map vlan 2000 vni 5000 map bridge-domain 4095 vni 8100

	`activate` `!`
Case	MISSING Child Config
Action	PUSH Child Config
Remarks	If the switch does not have the child configuration, push the child configuration.

Cluster Fabric

EFA Config	`cluster fabric-2-cluster-1`
	`peer 10.20.20.9`
	`peer-interface Port-channel 64`
	`peer-keepalive`
	`auto`
	`!`

Switch Config	Empty
Case	MISSING Config
Action	PUSH Config
Remarks	If the switch does not contain the configuration, push the new configuration.

EFA Config	**`cluster fabric-2-cluster-1`**
	`peer 10.20.20.9`
	`peer-interface Port-channel 64`
	`peer-keepalive`
	`auto`
	`!`

Switch Config	**`cluster fabric-3-cluster-1`**
	`peer 10.20.20.9`
	`peer-interface Port-channel 64`
	`peer-keepalive`
	`auto`
	`!`

Case	MISSING Parent Config
Action	PUSH Entire Config
Remarks	If the switch does not contain the parent configuration, push the entire configuration.

Interface Ethernet MCT Cluster

EFA Config	`interface Ethernet 0/11`
	`description clusterPeerIntfMember`
	`channel-group 64 mode active type standard`
	`lacp timeout long`
	`no shutdown`
	`!`

Switch Config	Empty

Case	MISSING Config
Action	PUSH Config
Remarks	If the switch does not contain the intended configuration, push the configuration.

Interface Ethernet Links Leaf/Spine

EFA Config	Leaf 1 Config
	`interface Ethernet 0/26`
	`description Link to 10.24.48.133 Spine`
	`ip address 10.10.10.20/31`
	`bfd interval 400 min-rx 400 multiplier 5`
	`no shutdown`
	`!`
	Leaf 2 Config
	`interface Ethernet 0/26`
	`description Link to 10.24.48.133 Spine`
	`ip address 10.10.10.21/31`
	`bfd interval 400 min-rx 400 multiplier 5`
	`no shutdown`
	`!`

Switch Config	Empty
Case	MISSING Config

Action	PUSH Config to the both the Leafs
Remarks	If the switch does not contain the intended configuration, push the configuration.

EFA Config	Leaf 1 Config interface Ethernet 0/26 description Link to 10.24.48.133 Spine ip address 10.10.10.20/31 bfd interval 400 min-rx 400 multiplier 5 no shutdown ! Leaf 2 Config interface Ethernet 0/26 description Link to 10.24.48.133 Spine ip address 10.10.10.21/31 bfd interval 400 min-rx 400 multiplier 5 no shutdown !
Switch Config	Leaf 1 Config interface Ethernet 0/26 description Link to 10.24.48.133 Spine **ip address 10.10.10.20/31** bfd interval 400 min-rx 400 multiplier 5 no shutdown ! Leaf 2 Config interface Ethernet 0/26 description Link to 10.24.48.133 Spine **ip address 10.10.10.24/31** bfd interval 400 min-rx 400 multiplier 5 no shutdown !
Case	CONFLICT Config
Action	Remove Child Config and PUSH Child Config
Remarks	If the switch Configuration on Leaf 2 does not have a valid IP Pair configuration, remove the child configuration and push the child configuration with a valid IP Pair.

EFA Config	interface Ethernet 0/26 description Link to 10.24.48.133 Spine ip address 10.10.10.20/31 **bfd interval 400 min-rx 400 multiplier 5**

	no shutdown !
Switch Config	interface Ethernet 0/26 description Link to 10.24.48.133 Spine ip address 10.10.10.20/31 no shutdown !
Case	MISSING Config
Action	PUSH Config
Remarks	If the switch does not contain the intended configuration, push the new configuration.

EFA Config	interface Ethernet 0/26 description Link to 10.24.48.133 Spine ip address 10.10.10.20/31 **bfd interval 400 min-rx 400 multiplier 5** **no shutdown** !
Switch Config	interface Ethernet 0/26 description Link to 10.24.48.133 Spine ip address 10.10.10.20/31 **bfd interval 500 min-rx 500 multiplier 6** **no shutdown** !
Case	CONFLICT Child Config
Action	PUSH Child Config
Remarks	If the switch contains conflicting child configuration, remove the existing child configuration and push the intended child configuration.

Interface Loopback

EFA Config	interface Loopback 2 ip address 172.31.254.19/32 no shutdown !
Switch Config	Empty
Case	MISSING Config
Action	PUSH Config
Remarks	If the switch does not contain the intended configuration, push the configuration.

Interface Port Channel

EFA Config	`interface Port-channel 64`
	`description MCTPeerInterface`
	`ip address 10.20.20.8/31`
	`bfd interval 400 min-rx 400`
	`multiplier 5`
	`no shutdown`
	`!`

Switch Config	Empty
Case	MISSING Config
Action	PUSH Config
Remarks	If the switch does not contain the intended configuration, push the configuration.

EFA Config	`interface Port-channel 64`
	`description MCTPeerInterface`
	`ip address 10.20.20.8/31`
	`bfd interval 400 min-rx 400`
	`multiplier 5`
	`no shutdown`
	`!`

Switch Config	`interface Port-channel 64`
	`description MCTPeerInterface`
	`ip address 10.20.20.8/31`
	`no shutdown`
	`!`

Case	MISSING Child Config
Action	PUSH Child Config
Remarks	If the switch does not contain the child configuration, push the child configuration.

EFA Config	`interface Port-channel 64`
	`description MCTPeerInterface`
	`ip address 10.20.20.8/31`
	`bfd interval 400 min-rx 400`
	`multiplier 5`
	`no shutdown`
	`!`

Switch Config	`interface Port-channel 64`
	`description MCTPeerInterface`
	`ip address 10.20.20.8/31`
	`bfd interval 500 min-rx 500`
	`multiplier 6`
	`no shutdown`
	`!`

Case	CONFLICT Child Config

Action	PUSH Child Config
Remarks	If the switch contains the conflicting child configuration, remove the conflicting child configuration and push the child configuration.

EFA Config	Leaf 1
	`interface Port-channel 64`
	`description MCTPeerInterface`
	`ip address 10.20.20.8/31`
	`bfd interval 400 min-rx 400`
	`multiplier 5`
	`no shutdown`
	`!`
	Leaf 2
	`interface Port-channel 64`
	`description MCTPeerInterface`
	`ip address 10.20.20.9/31`
	`bfd interval 400 min-rx 400`
	`multiplier 5`
	`no shutdown`
	`!`

Switch Config	Leaf 1
	`interface Port-channel 64`
	`description MCTPeerInterface`
	`ip address 10.20.20.20/31`
	`bfd interval 400 min-rx 400`
	`multiplier 5`
	`no shutdown`
	`!`
	Leaf 2
	`interface Port-channel 64`
	`description MCTPeerInterface`
	`ip address 10.20.20.9/31`
	`bfd interval 400 min-rx 400`
	`multiplier 5`
	`no shutdown`
	`!`

Case	CONFLICT Child Config
Action	REMOVE Child Config, PUSH Child Config
Remarks	If the switch configuration contains conflicting child configuration in terms of invalid Pair IP between two leafs, remove the conflicting child configuration and push the child config with a valid Pair IP.

Router BGP

EFA Config	router bgp local-as 65000 capability as4-enable fast-external-fallover bfd interval 400 min-rx 400 multiplier 5 neighbor spine-group peer-group neighbor spine-group remote-as 64512 neighbor spine-group description To Spine neighbor spine-group bfd neighbor 10.10.10.21 peer-group spine-group neighbor 10.20.20.9 remote-as 65000 neighbor 10.20.20.9 next-hop-self neighbor 10.20.20.9 bfd address-family ipv4 unicast network 172.31.254.19/32 maximum-paths 8 graceful-restart ! address-family ipv6 unicast ! address-family l2vpn evpn graceful-restart neighbor spine-group encapsulation vxlan neighbor spine-group next-hop-unchanged neighbor spine-group enable-peer-as-check neighbor spine-group activate ! !
Switch Config	Empty
Case	MISSING Config
Action	PUSH Config
Remarks	If the switch does not contain the intended configuration, push the new configuration.

EFA Config	router bgp local-as 65000 capability as4-enable fast-external-fallover **bfd interval 400 min-rx 400 multiplier 5** neighbor spine-group peer-group neighbor spine-group remote-as 64512 neighbor spine-group description To Spine **neighbor spine-group bfd** neighbor 10.10.10.21 peer-group spine-group neighbor 10.20.20.9 remote-as 65000 neighbor 10.20.20.9 next-hop-self

	neighbor 10.20.20.9 bfd address-family ipv4 unicast network 172.31.254.19/32 maximum-paths 8 graceful-restart ! address-family ipv6 unicast ! address-family l2vpn evpn graceful-restart neighbor spine-group encapsulation vxlan neighbor spine-group next-hop-unchanged neighbor spine-group enable-peer-as-check neighbor spine-group activate ! !
Switch Config	router bgp local-as 65000 capability as4-enable fast-external-fallover neighbor spine-group peer-group neighbor spine-group remote-as 64512 neighbor spine-group description To Spine neighbor 10.10.10.21 peer-group spine-group neighbor 10.20.20.9 remote-as 65000 neighbor 10.20.20.9 next-hop-self address-family ipv4 unicast network 172.31.254.19/32 maximum-paths 8 graceful-restart ! address-family ipv6 unicast ! address-family l2vpn evpn graceful-restart neighbor spine-group encapsulation vxlan neighbor spine-group next-hop-unchanged neighbor spine-group enable-peer-as-check neighbor spine-group activate ! !
Case	MISSING Child Config
Action	PUSH Child Config
Remarks	If the switch does not contain the child configuration, push the child configuration.

EFA Config	router bgp local-as 65000 capability as4-enable fast-external-fallover

	bfd interval 400 min-rx 400 multiplier 5 neighbor spine-group peer-group neighbor spine-group remote-as 64512 neighbor spine-group description To Spine neighbor spine-group bfd neighbor 10.10.10.21 peer-group spine-group neighbor 10.20.20.9 remote-as 65000 **neighbor 10.20.20.9 next-hop-self** neighbor 10.20.20.9 bfd address-family ipv4 unicast network 172.31.254.19/32 maximum-paths 8 graceful-restart ! address-family ipv6 unicast ! address-family l2vpn evpn graceful-restart neighbor spine-group encapsulation vxlan neighbor spine-group next-hop-unchanged neighbor spine-group enable-peer-as-check neighbor spine-group activate ! !
Switch Config	router bgp local-as 65000 capability as4-enable fast-external-fallover bfd interval 400 min-rx 400 multiplier 5 neighbor spine-group peer-group neighbor spine-group remote-as 64512 neighbor spine-group description To Spine neighbor spine-group bfd neighbor 10.10.10.21 peer-group spine-group neighbor 10.20.20.9 remote-as 65000 neighbor 10.20.20.9 bfd address-family ipv4 unicast network 172.31.254.19/32 maximum-paths 8 graceful-restart ! address-family ipv6 unicast ! address-family l2vpn evpn graceful-restart neighbor spine-group encapsulation vxlan neighbor spine-group next-hop-unchanged neighbor spine-group enable-peer-as-check neighbor spine-group activate

	! !
Case	MISSING Child Config
Action	PUSH Child Config
Remarks	If the switch does not contain the child configuration, push the child configuration.

EFA Config	router bgp local-as 65000 capability as4-enable fast-external-fallover bfd interval 400 min-rx 400 multiplier 5 neighbor spine-group peer-group neighbor spine-group remote-as 64512 neighbor spine-group description To Spine neighbor spine-group bfd neighbor 10.10.10.21 peer-group spine-group neighbor 10.20.20.9 remote-as 65000 neighbor 10.20.20.9 next-hop-self neighbor 10.20.20.9 bfd address-family ipv4 unicast network 172.31.254.19/32 maximum-paths 8 graceful-restart ! address-family ipv6 unicast ! address-family l2vpn evpn graceful-restart neighbor spine-group encapsulation vxlan neighbor spine-group next-hop-unchanged neighbor spine-group enable-peer-as-check neighbor spine-group activate ! !
Switch Config	router bgp local-as 65000 capability as4-enable fast-external-fallover bfd interval 400 min-rx 400 multiplier 5 neighbor spine-group peer-group neighbor spine-group remote-as 64512 neighbor spine-group description To Spine neighbor spine-group bfd neighbor 10.10.10.21 peer-group spine-group neighbor 10.20.20.9 remote-as 65000 neighbor 10.20.20.9 next-hop-self neighbor 10.20.20.9 bfd

	`address-family ipv4 unicast`
	`!`
	`address-family ipv6 unicast`
	`!`
	`address-family l2vpn evpn`
	`graceful-restart`
	`neighbor spine-group encapsulation`
	`vxlan`
	`neighbor spine-group next-hop-`
	`unchanged`
	`neighbor spine-group enable-peer-`
	`as-check`
	`neighbor spine-group activate`
	`!`
	`!`

Case	MISSING Child Config
Action	PUSH Child Config
Remarks	If the switch does not contain the child configuration, push the child configuration.

EFA Config	`router bgp`
	`local-as 65000`
	`capability as4-enable`
	`fast-external-fallover`
	`bfd interval 400 min-rx 400`
	`multiplier 5`
	`neighbor spine-group peer-group`
	`neighbor spine-group remote-as`
	`64512`
	`neighbor spine-group description`
	`To Spine`
	`neighbor spine-group bfd`
	`neighbor 10.10.10.21 peer-group`
	`spine-group`
	`neighbor 10.20.20.9 remote-as 65000`
	`neighbor 10.20.20.9 next-hop-self`
	`neighbor 10.20.20.9 bfd`
	`address-family ipv4 unicast`
	`network 172.31.254.19/32`
	`maximum-paths 8`
	`graceful-restart`
	`!`
	`address-family ipv6 unicast`
	`!`
	`address-family l2vpn evpn`
	`graceful-restart`
	`neighbor spine-group encapsulation`
	`vxlan`
	`neighbor spine-group next-`
	`hopunchanged`
	`neighbor spine-group enable-peer-`
	`ascheck`
	`neighbor spine-group activate`
	`!`
	`!`

Switch Config	`router bgp`
	`local-as 65000`
	`capability as4-enable`
	`fast-external-fallover`

	`bfd interval 400 min-rx 400 multiplier 5`
	`neighbor spine-group peer-group`
	`neighbor spine-group remote-as 64512`
	`neighbor spine-group description To Spine`
	`neighbor spine-group bfd`
	`neighbor 10.10.10.21 peer-group spine-group`
	`neighbor 10.20.20.9 remote-as 65000`
	`neighbor 10.20.20.9 next-hop-self`
	`neighbor 10.20.20.9 bfd`
	`address-family ipv4 unicast`
	`network 172.31.254.19/32`
	`maximum-paths 8`
	`graceful-restart`
	`!`
	`address-family ipv6 unicast`
	`!`
	`address-family l2vpn evpn`
	`!`
	`!`

Case	MISSING Child Config
Action	PUSH Child Config
Remarks	If the switch does not contain the child configuration, push the child configuration.

EFA Config	`router bgp`
	`local-as 65000`
	`capability as4-enable`
	`fast-external-fallover`
	`bfd interval 400 min-rx 400 multiplier 5`
	`neighbor spine-group peer-group`
	`neighbor spine-group remote-as 64512`
	`neighbor spine-group description To Spine`
	`neighbor spine-group bfd`
	`neighbor 10.10.10.21 peer-group spine-group`
	`neighbor 10.20.20.9 remote-as 65000`
	`neighbor 10.20.20.9 next-hop-self`
	`neighbor 10.20.20.9 bfd`
	`address-family ipv4 unicast`
	`network 172.31.254.19/32`
	`maximum-paths 8`
	`graceful-restart`
	`!`
	`address-family ipv6 unicast`
	`!`
	`address-family l2vpn evpn`
	`graceful-restart`
	`neighbor spine-group encapsulation vxlan`
	`neighbor spine-group next-hop-unchanged`
	`neighbor spine-group enable-peer-as-check`

	`neighbor spine-group activate`
	`!`
	`!`

Switch Config	`router bgp`
	`local-as 65000`
	`capability as4-enable`
	`fast-external-fallover`
	`bfd interval 500 min-rx 500 multiplier 7`
	`neighbor spine-group peer-group`
	`neighbor spine-group remote-as 64512`
	`neighbor spine-group description To Spine`
	`neighbor spine-group bfd`
	`neighbor 10.10.10.21 peer-group spine-group`
	`neighbor 10.20.20.9 remote-as 65000`
	`neighbor 10.20.20.9 next-hop-self`
	`neighbor 10.20.20.9 bfd`
	`address-family ipv4 unicast`
	`network 172.31.254.19/32`
	`maximum-paths 8`
	`graceful-restart`
	`!`
	`address-family ipv6 unicast`
	`!`
	`address-family l2vpn evpn`
	`graceful-restart`
	`neighbor spine-group encapsulation vxlan`
	`neighbor spine-group next-hop-unchanged`
	`neighbor spine-group enable-peer-as-check`
	`neighbor spine-group activate`
	`!`
	`!`

Case	CONFLICT Child Config
Action	PUSH Child Config
Remarks	If the switch contains conflicting child configuration, push the new child configuration.

Brownfield Fabric Service Overview

In a Brownfield deployment, the legacy software coexists with the new software.

The Brownfield Fabric Service provides the following functionality:

- Ability to migrate the Fabric deployed through an older version of EFA to a newer version of EFA. For instance, if an older EFA server is dismantled, and a newer EFA version or the same EFA version is installed on a different server.
- Ability to migrate from EFA on the TPVM to EFA on an external server.

With this feature, you can migrate the Fabric being configured fully or partially through the use of the SLX CLI or out-of-band means, provided there are no conflicts with EFA Fabric settings. Brownfield deployments are not supported for Tenant Services. The Fabric Service learns and fetches the configuration on the devices through the Inventory Service, performs validation checks, and generates appropriate errors for deviations in the configuration.

Pre-validation of Configuration

This section covers the use case where the devices have a preexisting configuration which may have been configured either through Embedded Fabric/CLI/EFA or some out-of-band means.

When devices with preexisting configuration are added to EFA, fabric service performs certain validations before adding devices to the EFA fabric. If any of the configuration that is retrieved from the devices does not fall under the fabric settings range, an error is displayed. You can perform corrective actions to add such devices to the EFA fabric.

Global Device Configuration

Use Case	Valid	Invalid
L2 MTU	If the value fetched from the device is same as that configured in fabric settings.	If the value does not match, an error is displayed.
IP MTU	If the value fetched from the device is same as that configured in fabric settings.	If the value does not match, an error is displayed.
ASN	If the value fetched from the device is within the ASN range configured in fabric settings.	If the value is out of range of what is configured in fabric settings. If the value is conflicting with existing device which is already added to Fabric.

Interface Configuration

Use Case	Valid	Invalid
Interface IP Address	Received an IP address within “Link IP range” of fabric settings.	• If IP address received is out of range, a validation error is displayed. • If IP address received is within the range but is already in use/reserved by fabric, a validation error is displayed.
Loopback Interface ID	Loopback Interface ID is reconciled	
VTEP Loopback Interface ID	VTEP Loopback interface ID is reconciled	
Loopback Interface IP Address	Loopback IP address is within “Loopback IP range” of fabric settings.	If the loopback IP address is out of range of “Loopback IP range” of fabric settings, an error is displayed.

Use Case	Valid	Invalid
VE IP address	VE IP address is within "MCT Link IP Range" of fabric settings.	• If IP address received is out of range, a validation error is displayed. • If IP address received is within the range but is already in use/reserved by fabric, a validation error is displayed.
Static IP Route (Applicable for SLX 9540 and SLX 9640)	If the IP route is same as fabric intended configuration.	If the nexthop does not point to VE IP.

MCT Configuration

MCT peer and VE validations are platform specific.

Use Case	Valid	Invalid
MCT Cluster Name	MCT Cluster Name is different from fabric name in fabric properties. MCT Cluster Name learned from device is used while configuring the device, so that the cluster name is reconciled.	
Cluster Control VLAN	Cluster Control VLAN matches with "Control VLAN" of fabric settings.	If the VLAN does not match, an error is displayed.
Cluster Control VE	Cluster Control VE matches with "Control VE" of fabric settings	If the VE does not match, an error is displayed.
MCT Peer IP	Peer IP address is within IP range of "MCT Link IP range" of fabric settings.	If Peer IP address is out of IP range, an error is displayed.
MCT Peer Interface	Peer Interface type matches with fabric settings and ID is reconciled.	

Overlay Gateway Configuration

Use Case	Valid	Invalid
Overlay Gateway Name	Gateway Name is reconciled.	
VNI Auto Map	VNI auto mapping setting configured on the device matches with fabric settings.	If gateway is in activated state and VNI Auto Map setting is different from the fabric settings, an error is displayed.
Overlay Gateway Interface	Gateway Interface, for example loopback 2 is reconciled.	

EVPN Configuration

Use Case	Valid	Invalid
EVPN Name	EVPN Name is reconciled	
MAC Aging Timeout (check based on device capability, applies to SLX 9140 and SLX 9240)	Field value is overwritten by the fabric settings value.	
MAC Aging Conversation Timeout (check based on device capability, applies to SLX 9140 and SLX 9240)		
MAC Move Limit (check based on device capability)		
ArpAgingTimeout (check based on device capability)		
Duplicate Mac Timer		
Duplicate MAC Timer MAX Count		

BGP Configuration

To pre-validate the BGP configuration, the BGP configuration must be prepared similar to the add device phase. Once the BGP configuration is computed, the configuration retrieved from the device is compared against it.

Use Case	Valid	Invalid
Router ID	If the generated router id matches the one received from the device.	If the router id does not match, an error is displayed.
BFD Enable/Disable	If the BFD value from device matches the one that is computed from fabric settings. While configuring the fabric, the values computed by fabric service override the ones on the device.	NA
BFD Tx/Rx Timer Values	If the values from device match with the ones that are generated. While configuring the fabric, the values computed by fabric service override the ones on the device.	NA
Network Address	If the value is within “Loopback IP Range” of fabric settings or matches the computed value.	If the value is out of range or clashes with another IP neighbor already stored in fabric DB, an error is displayed.

Use Case	Valid	Invalid
EVPN Neighbor IP Address	If the neighbor IP address falls in range of “Link IP range” of fabric settings. There may be a case where the neighbor IP address is valid but neighbor is not part of fabric. You can ignore such validation as that configuration is a no-op for fabric.	If the value is out of range or clashes with another IP neighbor already stored in fabric DB, an error is displayed.
Remote ASN	If the ASN is within the range of fabric settings and not already in use by another neighbor.	If the ASN is out of range or already reserved.
Peer group name	If the peer group name matches the peer group that is computed.	If the value does not match, an error is displayed.

BGP Tables

The following BGP tables help in computing the diffs for the events from the inventory service.

- Router BGP table
- BGP peer group table
- BGP IP address family table
- BGP IP neighbor address table
- BGP EVPN address family table
- BGP EVPN neighbor address table

All BGP tables handle the DB migration so that upgrade from older EFA to newer EFA works.

For more information on the attributes of each table, refer to Database schema section or fabric_schema.sql file.

BGP Events

The following BGP events from inventory service are handled as part of event handling.

BGP Router Delete

When router BGP delete message is received, fabric passes through all the IP and EVPN neighbors, peer group tables and the entries corresponding to the device for which router BGP delete message is received and mark the entries as ‘create’ to configure the router BGP and its related neighbors on the device.

BGP IP Neighbor Delete

When BGP IP neighbor delete message is received, fabric passes through all the IP neighbors deleted and which exists in fabric database for a given neighbor IP or remote ASN and mark the entries as ‘create’ to configure the deleted IP neighbors on the device.

BGP IP Neighbor Update

When BGP IP neighbor update message is received, fabric passes through all the IP neighbors matching the neighbor IP for the device in the database. If any of the fabric managed attribute in the IP neighbor table is changed, fabric marks the entries as 'update' and pushes the configuration back to the device.

BGP EVPN Neighbor Delete

When BGP EVPN neighbor delete message is received, fabric passes through all the EVPN neighbors deleted and which exists in fabric database for a given neighbor IP or remote ASN and mark the entries as 'create' to configure back the deleted EVPN neighbors on the device.

BGP EVPN Neighbor Update

When BGP EVPN neighbor update message is received, fabric passes through all the EVPN neighbors matching the neighbor IP for the device in database. If any of the fabric managed attribute in the EVPN neighbor table is changed, fabric marks the entries as 'update' and pushes the configuration back to the device.

Peer Group Delete

Peer Group Delete message is received only when there are no IP/EVPN neighbors associated with it. If there are no IP/EVPN neighbors associated with it, fabric marks the Peer Group as 'delete'.

Peer Group Update

When Peer group attributes such as BFD and remote ASN change, inventory sends a peer group update message. The fabric processes this message and checks if the peer group exists in the database. If the peer group exists and there are changes to the attributes, the fabric pushes the peer group configuration with fabric intended configuration back to the device.

BGP IP Address Family Delete

BGP IP Address Family Delete message is received when the IP address-family for a device is deleted through CLI or out-of-band means. When fabric receives this message, it passes through all the IP neighbors associated with that address-family and marks the entries as 'create config' to restore all the deleted IP neighbors on the device.

BGP EVPN Address Family Delete

BGP EVPN Address Family Delete message is received when the EVPN address-family for a device is deleted through CLI or out-of-band means. When fabric receives the message, it passes through all the EVPN neighbors associated with that address-family and marks the entries as 'create config' to restore all the deleted EVPN neighbors on the device.

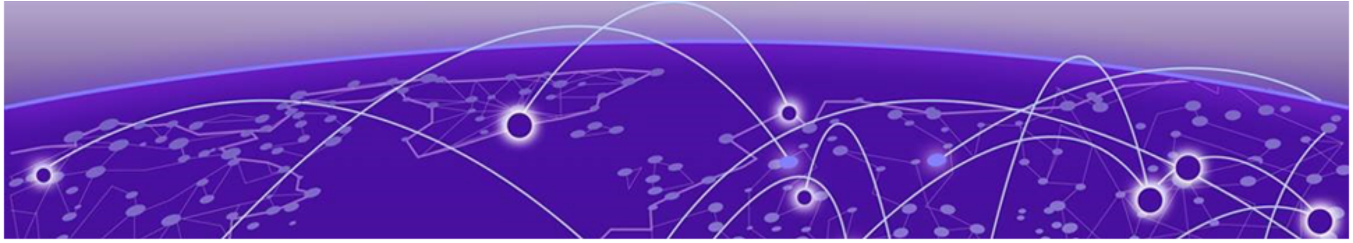
Limitations

- Migration of older version of EFA to newer version of EFA is similar to adding devices to newer fabric and configuring fabric. If there are no changes in fabric configuration, devices migrate to the newer version of EFA successfully. Limitation on pre-validations and post-validations apply. Similar limitations apply to fabric configured through CLI or out-of-band means.
- As part of pre-validations, ASN, IP address of interface, loopback interface IP, and MCT VE IP validations are done. Other pre-validations mentioned in Pre-validation of Configuration are NOT handled.

- Post-validation is NOT handled.
- If you enter a fabric name that already exists with added devices, import fails.
- If the existing fabric type mismatches with the fabric to be imported, import fails.
- If the fabric name entered is “default” and import fails, the fabric properties rollback to the original state.
- Non-Clos Embedded fabric import is not supported.
- If the user explicitly changes the device credentials which are not in-sync with Embedded fabric database, the device registration fails.

The following are not supported:

- Devices with preexisting configuration and configured through out-of-band means such as CLI and not through Embedded fabric.
- Devices configured through Embedded fabric and firmware upgraded to SLX 20.1.1 and later releases containing simplified MCT feature. Import fails if you perform firmware upgrade before importing. Import the fabric first and then perform firmware upgrade.



Tenant Services Provisioning

Tenant Services Provisioning Overview

Tenant Services exposes the CLI and REST API for automating the Tenant network configuration on the Clos and Non-Clos overlay fabric.

Tenant network configuration includes VLAN, BD, VE, EVPN, VTEP, VRF, and Router BGP configuration on the necessary fabric devices to provide L2-Extension, L3-Extension across the fabric, L2-Handoff, and L3-Handoff at the edge of the fabric.

Tenant Services provisioning automates the Tenant configuration, which can be a subset of the combinations provided by the switching hardware.

Tenant Services supports multiple fabrics.

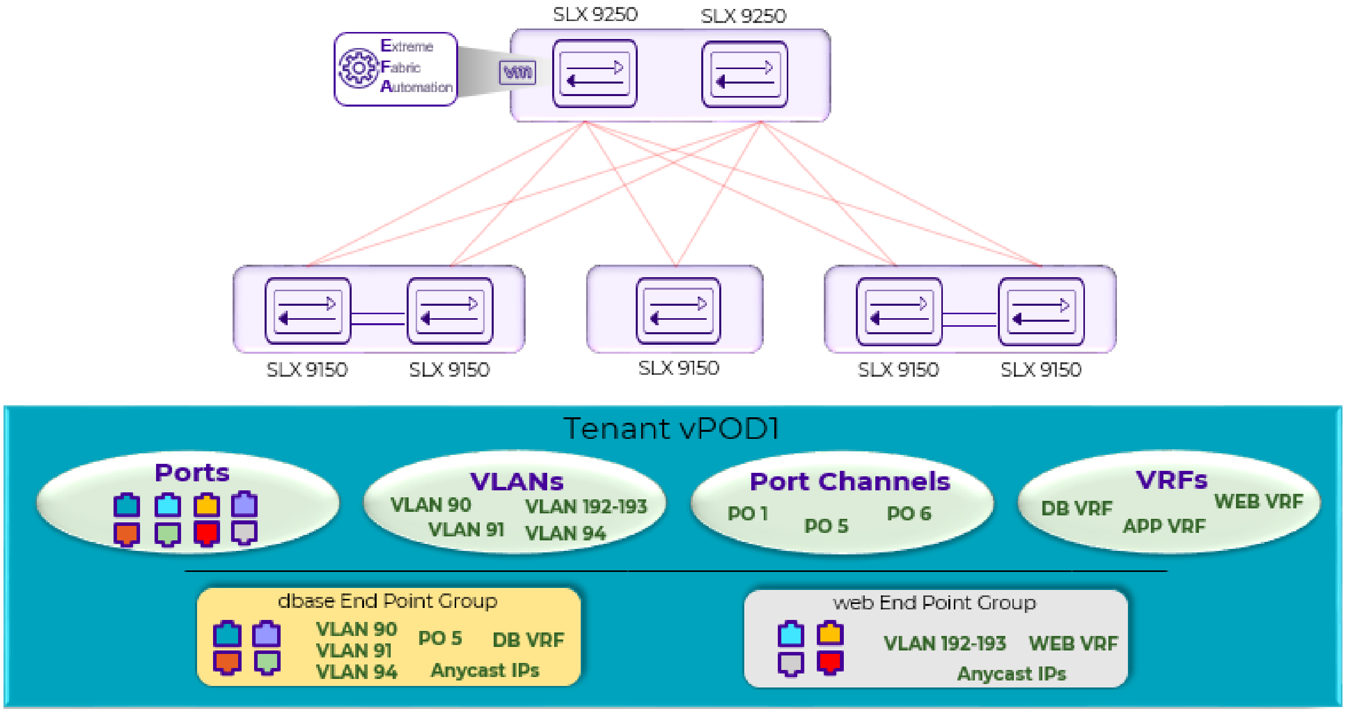


Figure 15: Tenant Services Overview

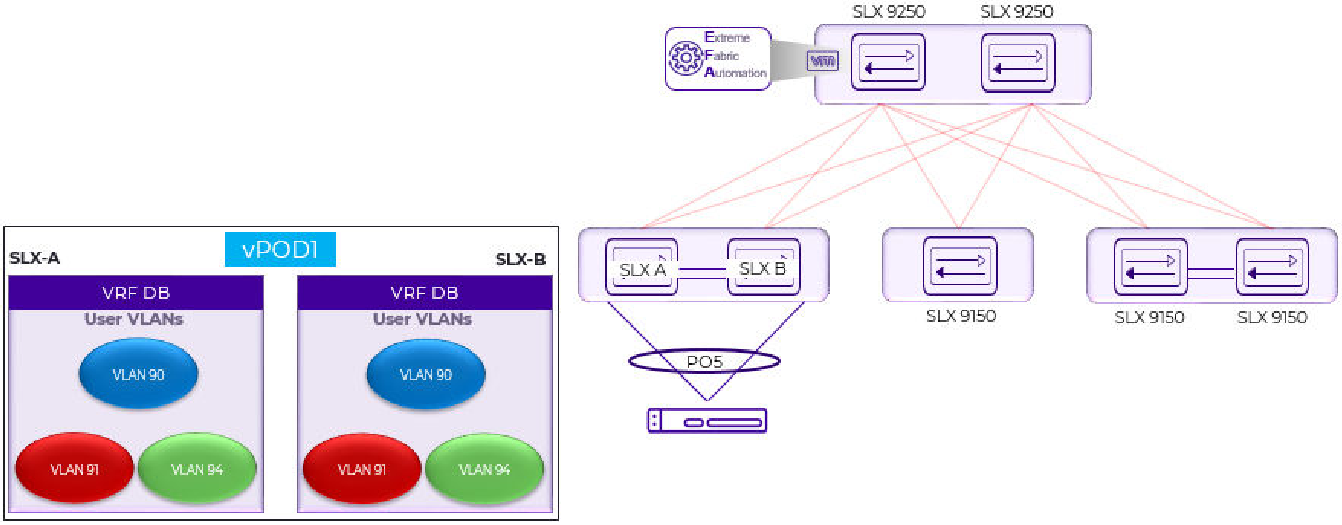


Figure 16: Tenant Name vPOD1, VRF Name DB

Tenant

A Tenant is a logical construct that owns resources as follows:

- VLAN range: Ctags pertaining to which the traffic is expected to ingress and egress.

  Note
  Ctag (Customer VLAN tag) is used to identify the customer broadcast domain. In the IP Fabric network, it represents the customer and is mapped into a VXLAN tunnel thru a VNI (virtual network identifier). The VNI is the ID used to identify the VXLAN tunnel. With auto VNI mapping the Ctag ID equals the VNI. Users can also manually map Ctags to user-defined VNIs. These VNIs can be VLAN IDs (up to 4k) or to BD's (bridge domain) IDs.

- Device ports: Ports on which the traffic is expected to ingress and egress.

VLAN-based Tenant

For a VLAN based tenant, realization of network on the device is done using VLAN and switchport VLANs. Bridge domains are used for EVPN IRB.

Bridge domain-based Tenant

For a BD based tenant, realization of network on the device is done using BD and BD-LIF. BD is used for EVPN IRB.

Scalability

Table 14: VNI scalability

VNI type	Scale
Non-auto VNI mapping	• The number of VNI (networks) supported per device = 8K [4K VLAN + 4K BD] • The maximum number of VNI (networks) supported in the fabric = [8K * number of devices in the fabric].
Auto VNI mapping	• The number of VNI (networks) supported per device = 8K [4K VLAN + 4K BD] • The number of VNI (networks) supported per fabric = 8K

Event handling

Event handling specifies the scope of the tenant configuration on the devices.

Devices are added to the Tenant service only when the Fabric is provisioned on the devices.

An event is an occurrence of a device being removed from the Fabric or from the Inventory.

- When a device is removed from the Fabric or Inventory, the device is cleaned up from Tenant Service and the Tenant configuration is removed from the device.

- User-created entities, such as Tenant, VRF, and EPG, are not deleted whereas references for ports/ port-channels of deleted devices are removed.

Clos Fabric with Non-auto VNI Maps

Auto VNI simplifies the mapping IDs by using the VLAN ID as the VNI ID, for example VLAN 100 = VNI 100.

This method of mapping works well in environments where overlapping VLANs are not being used. However, if two different tenants are using VLAN 100, VNI 100 cannot be used by both. At this point, manual mapping of VLAN to VNI is required. Extreme Fabric Automation simplifies this process by allowing VNI ranges for tenants to automate “manual” mapping to work for overlapping VLANs.

The following figure shows a 3-stage Clos topology.

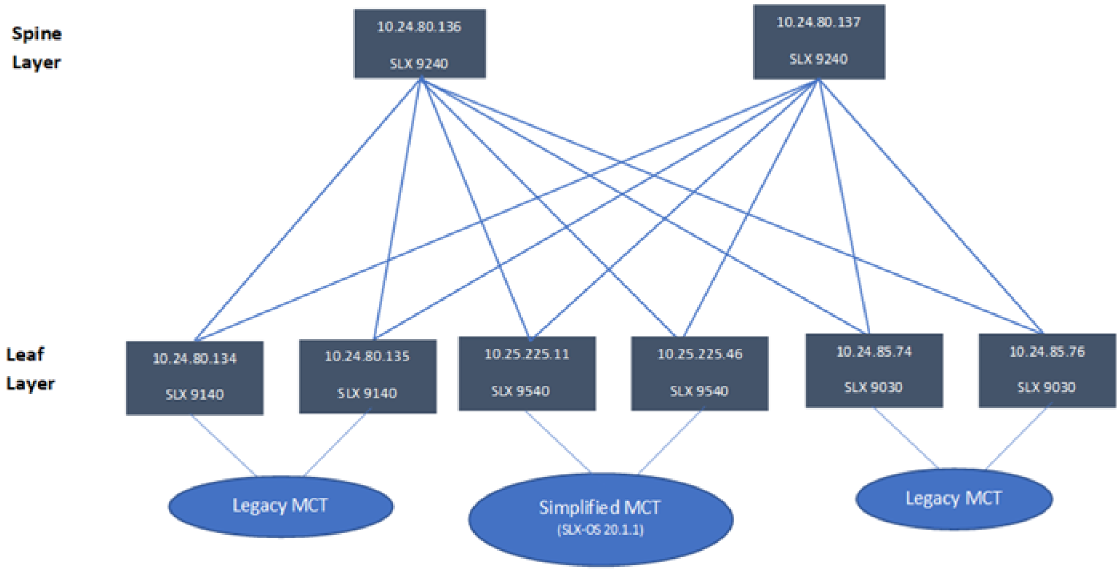


Figure 17: 3-stage Clos topology

The following commands configured the 3-stage Clos topology:

```
efa fabric create --name fabric1

efa fabric setting update --name fabric1 --vni-auto-map No

efa fabric device add-bulk --spine 10.24.80.136 --border-leaf 10.25.225.11,10.25.225.46
--leaf 10.24.80.134-135,10.24.85.74,10.24.85.76 --username admin --password password
--name fabric1

efa fabric configure --name fabric1
```

The following figure shows tenant constructs in the Clos Fabric.

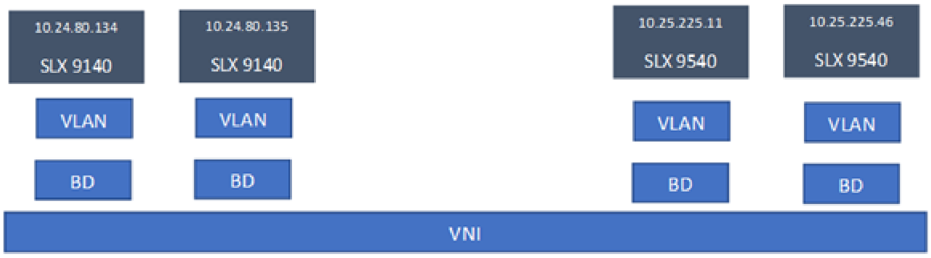


Figure 18: Scope of tenant constructs

Clos Fabric with Auto VNI Map

- In Clos fabric with auto VNI map, the VNI is statically derived using the VLAN ID or BD ID.
    - For the VLAN case, VNI = VLAN ID
    - For the BD case, VNI = 4096 + BD ID
    - User will not be able to reserve l2-vni-range or l3-vni-range for a given tenant.
    - User will not be able to provide a specific l2-vni/l3-vni in an EPG.
- VLAN Based Tenants:

  Multiple VLAN Based tenants cannot share the same VLAN, considering the multiple tenants cannot share the same VNI.
- BD Based Tenants:

  Multiple BD Based tenants can share the same VLAN, as the VLANs from each tenant will be mapped to a unique BD and further a unique VNI.

Multi Tenancy

EFA supports multi tenancy by allowing multiple tenants to have overlapping ctags and non-overlapping L2VNI. A tenant ctag will get a unique L2VNI and a unique network allocated in the fabric

The following example shows a multi tenancy configuration.

```
efa tenant create --name tenant11 --vrf-count 10 --vlan-range 2-4090 --port
10.24.80.134[0/15-17],10.24.80.135[0/15-17],10.25.225.11[0/15-17],10.25.225.46[0/15-17],
10.24.85.74[0/15-17],10.24.85.76[0/15-17] --description Subscriber1

efa tenant show
+----------+-------------+-------------+------------+-----------+-----------
+-----------------------+
|   Name   | L2VNI-Range | L3VNI-Range | VLAN-Range | VRF-Count | Enable-BD |
Ports          |
+----------+-------------+-------------+------------+-----------+-----------
+-----------------------+
| tenant11 |             |             | 2-4090     | 10        | False     |
10.24.85.74[0/15-17]   |
|          |             |             |            |           |           |
10.24.80.135[0/15-17] |
|          |             |             |            |           |           |
10.25.225.11[0/15-17] |
|          |             |             |            |           |           |
10.25.225.46[0/15-17] |
|          |             |             |            |           |           |
10.24.80.134[0/15-17] |
|          |             |             |            |           |           |
10.24.85.76[0/15-17]  |
+----------+-------------+-------------+------------+-----------+-----------
+-----------------------+

efa tenant create --name tenant12 --vrf-count 10  --vlan-range 2-4090 --port
10.24.80.134[0/18-20],10.24.80.135[0/18-20],10.25.225.11[0/18-20],10.25.225.46[0/18-20],
10.24.85.74[0/18-20],10.24.85.76[0/18-20]
Tenant Creation Failed:
         Vlan (2) overlaps with Tenant (tenant11)

efa tenant create --name tenant21 --vrf-count 10 --enable-bd --port 10.24.80.134[0/21-25],
10.24.80.135[0/21-25],10.24.85.74[0/21-25],10.24.85.76[0/21-25],10.25.225.11[0/21-25],
10.25.225.46[0/21-25]

efa tenant create --name tenant22 --vrf-count 10 --enable-bd --port 10.24.80.134[0/26-30],
10.24.80.135[0/26-30],10.24.85.74[0/26-30],10.24.85.76[0/26-30],10.25.225.11[0/26-30],
10.25.225.46[0/26-30]

efa tenant show
+----------+-------------+-------------+------------+-----------+-----------
+-----------------------+
|   Name   | L2VNI-Range | L3VNI-Range | VLAN-Range | VRF-Count | Enable-BD |
Ports         |
+----------+-------------+-------------+------------+-----------+-----------
+-----------------------+
| tenant11 |             |             | 2-4090     | 10        | False     |
10.25.225.46[0/15-17] |
|          |             |             |            |           |           |
10.25.225.11[0/15-17] |
|          |             |             |            |           |           |
10.24.80.135[0/15-17] |
|          |             |             |            |           |           |
10.24.85.74[0/15-17]  |
|          |             |             |            |           |           |
10.24.85.76[0/15-17]  |
|          |             |             |            |           |           |
10.24.80.134[0/15-17] |
+----------+-------------+-------------+------------+-----------+-----------
+-----------------------+
| tenant21 |             |             | 2-4090     | 10        | True      |
10.24.85.74[0/21-25]  |
|          |             |             |            |           |           |
10.25.225.11[0/21-25] |
|          |             |             |            |           |           |
10.25.225.46[0/21-25] |
|          |             |             |            |           |           |
10.24.80.134[0/21-25] |
|          |             |             |            |           |           |
10.24.85.76[0/21-25]  |
|          |             |             |            |           |           |
10.24.80.135[0/21-25] |
+----------+-------------+-------------+------------+-----------+-----------
+-----------------------+
| tenant22 |             |             | 2-4090     | 10        | True      |
10.24.85.76[0/26-30]  |
|          |             |             |            |           |           |
10.25.225.11[0/26-30] |
|          |             |             |            |           |           |
10.24.80.135[0/26-30] |
|          |             |             |            |           |           |
10.25.225.46[0/26-30] |
|          |             |             |            |           |           |
10.24.85.74[0/26-30]  |
|          |             |             |            |           |           |
10.24.80.134[0/26-30] |
+----------+-------------+-------------+------------+-----------+-----------
+-----------------------+

efa tenant epg create --name epg11 --tenant tenant11 --po po1115,po1215,po1315 --
switchport-mode trunk --switchport-native-vlan 11 --ctag-range 11-12

efa tenant epg create --name epg21 --tenant tenant21 --po po2121,po2221,po2321 --
switchport-mode trunk  --ctag-range 11-12

efa tenant epg create --name epg22 --tenant tenant21 --po po2122,po2222,po2322  --
switchport-mode trunk  --ctag-range 11-12

efa tenant epg show

=======================================================================
Name          : epg11
Tenant        : tenant11
Description   :
Ports         :
POs           : po1315, po1215, po1115
Port Property : switchport mode     : trunk
              : native-vlan-tagging : false
NW Policy     : ctag-range          : 11-12

Network Property [Flags : * - Native Vlan]
+------+--------+------------+---------+-------------+-------------+
| Ctag | L2-Vni | Anycast-ip | BD-name |  Dev-state  |  App-state  |
+------+--------+------------+---------+-------------+-------------+
| 11*  | 11     |            |         | provisioned | cfg-in-sync |
+------+--------+------------+---------+-------------+-------------+
| 12   | 12     |            |         | provisioned | cfg-in-sync |
+------+--------+------------+---------+-------------+-------------+

=======================================================================
=======================================================================
Name          : epg21
Tenant        : tenant21
Description   :
Ports         :
POs           : po2121, po2221, po2321
Port Property : switchport mode     : trunk
              : native-vlan-tagging : false
NW Policy     : ctag-range          : 11-12

Network Property [Flags : * - Native Vlan]
+------+--------+------------+--------------+-------------+-------------+
| Ctag | L2-Vni | Anycast-ip |   BD-name    |  Dev-state  |   App-state |
+------+--------+------------+--------------+-------------+-------------+
| 11   | 4099   |            | Auto-BD-4099 | provisioned | cfg-in-sync |
+------+--------+------------+--------------+-------------+-------------+
| 12   | 4100   |            | Auto-BD-4100 | provisioned | cfg-in-sync |
+------+--------+------------+--------------+-------------+-------------+

=======================================================================
=======================================================================
Name          : epg22
Tenant        : tenant21
Description   :
Ports         :
POs           : po2122, po2222, po2322
Port Property : switchport mode     : trunk
              : native-vlan-tagging : false
```

```
NW Policy     : ctag-range          : 11-12

Network Property [Flags : * - Native Vlan]
+------+--------+------------+--------------+-------------+-------------+
| Ctag | L2-Vni | Anycast-ip |   BD-name    |  Dev-state  |  App-state  |
+------+--------+------------+--------------+-------------+-------------+
| 12   | 4102   |            | Auto-BD-4102 | provisioned | cfg-in-sync |
+------+--------+------------+--------------+-------------+-------------+
| 11   | 4101   |            | Auto-BD-4101 | provisioned | cfg-in-sync |
+------+--------+------------+--------------+-------------+-------------+

efa tenant epg create --name epg23 --tenant tenant21 --po po2122,po2322  --switchport-
mode trunk  --ctag-range 21-22 --bridge-domain 21:Auto-BD-4101 --bridge-domain 22:Auto-
BD-4102

efa tenant epg show

======================================================================
Name          : epg11
Tenant        : tenant11
Description   :
Ports         :
POs           : po1315, po1215, po1115
Port Property : switchport mode     : trunk
              : native-vlan-tagging : false
NW Policy     : ctag-range          : 11-12

Network Property [Flags : * - Native Vlan]
+------+--------+------------+---------+-------------+-------------+
| Ctag | L2-Vni | Anycast-ip | BD-name |  Dev-state  |  App-state  |
+------+--------+------------+---------+-------------+-------------+
| 11*  | 11     |            |         | provisioned | cfg-in-sync |
+------+--------+------------+---------+-------------+-------------+
| 12   | 12     |            |         | provisioned | cfg-in-sync |
+------+--------+------------+---------+-------------+-------------+

======================================================================
======================================================================
Name          : epg21
Tenant        : tenant21
Description   :
Ports         :
POs           : po2121, po2221, po2321
Port Property : switchport mode     : trunk
              : native-vlan-tagging : false
NW Policy     : ctag-range          : 11-12

Network Property [Flags : * - Native Vlan]
+------+--------+------------+--------------+-------------+-------------+
| Ctag | L2-Vni | Anycast-ip |   BD-name    |  Dev-state  |   App-state |
+------+--------+------------+--------------+-------------+-------------+
| 11   | 4099   |            | Auto-BD-4099 | provisioned | cfg-in-sync |
+------+--------+------------+--------------+-------------+-------------+
| 12   | 4100   |            | Auto-BD-4100 | provisioned | cfg-in-sync |
+------+--------+------------+--------------+-------------+-------------+

======================================================================
======================================================================
Name          : epg22
Tenant        : tenant21
Description   :
Ports         :
POs           : po2122, po2222, po2322
Port Property : switchport mode     : trunk
```

```
              : native-vlan-tagging : false
NW Policy     : ctag-range          : 11-12

Network Property [Flags : * - Native Vlan]
+------+--------+------------+--------------+-------------+-------------+
| Ctag | L2-Vni | Anycast-ip |   BD-name    |  Dev-state  |  App-state  |
+------+--------+------------+--------------+-------------+-------------+
| 12   | 4102   |            | Auto-BD-4102 | provisioned | cfg-in-sync |
+------+--------+------------+--------------+-------------+-------------+
| 11   | 4101   |            | Auto-BD-4101 | provisioned | cfg-in-sync |
+------+--------+------------+--------------+-------------+-------------+

======================================================================
======================================================================
Name          : epg23
Tenant        : tenant21
Description   :
Ports         :
POs           :
Port Property : switchport mode     : trunk
              : native-vlan-tagging : false
NW Policy     : ctag-range          : 21-22

Network Property [Flags : * - Native Vlan]
+------+--------+------------+--------------+-------------+-------------+
| Ctag | L2-Vni | Anycast-ip |   BD-name    |  Dev-state  |  App-state  |
+------+--------+------------+--------------+-------------+-------------+
| 21   | 4101   |            | Auto-BD-4101 | provisioned | cfg-in-sync |
+------+--------+------------+--------------+-------------+-------------+
| 22   | 4102   |            | Auto-BD-4102 | provisioned | cfg-in-sync |
+------+--------+------------+--------------+-------------+-------------+

======================================================================
```

Layer 3 Network Services

The topics in this section describe Tenant provisioning in the Layer 3 network.

IPv6 Support

EFA supports provisioning of IPv6 anycast-gateway and auto provision the dependent configuration.

EFA Provisioning

```
efa tenant vrf create --name vrf1 --tenant tenant1 --rt-type both --rt 1:1

efa tenant epg create --name ten1epg1 --tenant tenant1 --port 10.24.80.134[0/15] --
switchport-mode trunk
 --ctag-range 1001 --anycast-ip 1001:10.0.1.1/24 --anycast-ipv6 1001:2001:10:0:1::1/64 --
vrf vrf1
```

Switch Config

Table 15: VNI scalability

vlan 1001 router-interface Ve 1001 suppress-nd suppress-arp ! leaf-9250-173# sh run in ve 1001 interface Ve 1001 vrf forwarding vrf1 ip anycast-address 10.0.1.1/24 ipv6 anycast-address 2001:10:0:1::1/64 no shutdown !	leaf-9250-173# sh run vrf vrf1 vrf vrf1 rd 172.31.254.13:1 evpn irb ve 8192 address-family ipv4 unicast route-target export 1:1 evpn route-target import 1:1 evpn ! address-family ipv6 unicast route-target export 1:1 evpn route-target import 1:1 evpn ! !
leaf-9250-173# show running-config router bgp address-family ipv4 unicast vrf vrf1 router bgp address-family ipv4 unicast vrf vrf1 local-as 4210000001 maximum-paths 2 !	leaf-9250-173# show running-config router bgp address-family ipv6 unicast vrf vrf1 router bgp address-family ipv6 unicast vrf vrf1 redistribute connected maximum-paths 2 !

VRF Backup Routing

Backup routing needs to be enabled when all links from a leaf device to the spine layer are down and tenant traffic is to be routed via the MCT neighbor.

EFA Provisioning

IPv4 and IPv6 range will be input at the fabric level. An IPv4 and an IPv6 address pair will be allocated to every MCT pair across all the tenant VRFs and the BGP session will be established between the same IP Pair.

Allocations happen per device:

1. Allocate a Bridge domain per VRF for backup routing.
2. Allocate a corresponding router-interface VE per BD/VRF.
3. Assign the IPv4 and IPv6 address allocated to each device on each of the VE interface.

> Note
> Same IPv4 and IPv6 address will be allocated on each of the VE interface belonging to different VRF.

4. Establish IBGP IPV4 neighborship with the MCT peer on a set of IP address per VRF.

5. Establish IBGP IPv6 neighborship with the MCT peer on a set of IPv6 address per VRF.
6. Configure "next-hop-self" on both the IPv4 and IPv6 neighbor.
7. Configure "active" on the IPv6 neighbor.

Example:

```
efa fabric setting update --name fabric1
--backup-routing-ipv4-range 21.1.1.0/24 --backup-routing-ipv6-range 2001:21:1:1::0/120
```

Example where backup routing is enabled:

```
efa fabric setting update --name nc --backup-routing-enable yes
```

Device Config

Table 16: Tenant1 VRF "vrf1"

leaf-9250-173# show running-config bridge-domain 3001 bridge-domain 3001 p2mp pw-profile default router-interface Ve 7001 bpdu-drop-enable ! leaf-9250-173# sh run in ve 7001 interface Ve 7001 vrf forwarding vrf1 ip address 21.1.1.10/31 ipv6 address 2001:21:1:1::10/127 no shutdown !	leaf-9250-173# show running-config router bgp address-family ipv4 unicast vrf vrf1 router bgp address-family ipv4 unicast vrf vrf1 local-as 4210000001 redistribute connected neighbor 21.1.1.11 remote-as 4210000001 neighbor 21.1.1.11 next-hop-self maximum-paths 2 ! ! leaf-9250-173# show running-config router bgp address-family ipv6 unicast vrf vrf1 router bgp address-family ipv6 unicast vrf vrf1 redistribute connected neighbor 2001:21:1:1::11 next-hop-self neighbor 2001:21:1:1::11 remote-as 4210000001 neighbor 2001:21:1:1::11 activate maximum-paths 2 ! !

Table 17: Tenant2 VRF "vrf2"

leaf-9250-173# show running-config bridge-domain 3002 bridge-domain 3002 p2mp pw-profile default router-interface Ve 7002 bpdu-drop-enable ! leaf-9250-173# sh run in ve 7002 interface Ve 7002 vrf forwarding vrf2 ip address 21.1.1.10/31 ipv6 address 2001:21:1:1::10/127 no shutdown !	leaf-9250-173# show running-config router bgp address-family ipv4 unicast vrf vrf2 router bgp address-family ipv4 unicast vrf vrf2 local-as 4210000001 redistribute connected neighbor 21.1.1.11 remote-as 4210000001 neighbor 21.1.1.11 next-hop-self maximum-paths 2 ! ! leaf-9250-173# show running-config router bgp address-family ipv6 unicast vrf vrf2 router bgp address-family ipv6 unicast vrf vrf2

Table 17: Tenant2 VRF “vrf2” (continued)

	redistribute connected neighbor 2001:21:1:1::11 next-hop-self neighbor 2001:21:1:1::11 remote-as 4210000001 neighbor 2001:21:1:1::11 activate maximum-paths 2 ! !

VRF Static Route Configuration

The Static Route Configuration in an option at the tenant VRF level that allows a user to provide static routes per tenant VRF.

VRF Create

```
efa tenant vrf create --name vrfv1 --tenant t3 --local-asn 65234 --ipv4-static-route-next-
hop 0.0.0.0/24,16.0.0.2
--ipv6-static-route-bfd 200::2,200::3,123,455,5 --ipv4-static-route-bfd
16.0.0.2,16.0.0.3,123,456,3
```

VRF Update

```
efa tenant vrf update --name <vrf-name> --tenant <tenant-name> --operation <static-route-
add|static-route-delete> --ipv6static-route-next-hop <destination, next-hop>
--ipv4static-route-next-hop <destination, next-hop>
```

Examples: (tenant vrf create and tenant vrf show)

```
efa tenant vrf create --name red --tenant tenant11 --ipv6static-route-next-hop
2000::/64,1001::2 --ipv6static-route-next-hop 2000::/64,1002::2 --ipv6static-route-next-
hop 2000::/64,1003::2 --ipv6static-route-next-hop 2000::/64,1004::2
--ipv6static-route-next-hop 2000::/64,1005::2 --ipv4static-route-next-hop
22.0.0.0/24,13.0.0.1 --ipv4static-route-next-hop 22.0.0.0/24,13.0.0.2
--ipv4static-route-next-hop 22.0.0.0/24,13.0.0.3 --ipv4static-route-next-hop
22.0.0.0/24,13.0.0.4 --ipv4static-route-next-hop 22.0.0.0/24,13.0.0.5

efa tenant vrf show --name red --tenant tenant11
```

```
===============================================================================
===============
Name : red
Tenant Name : tenant11
L3 VNI :
IRB BD :
IRB VE :
Route Target :
Static Route : Network->Nh1,Nh2,..
: 22.0.0.0/24->13.0.0.1,13.0.0.2,13.0.0.3,13.0.0.4,13.0.0.5
2000::/64->1001::2,1002::2,1003::2,1004::2,1005::2
Local Asn :
```

```
===================================================================================
===============
```

Switch Config

```
vrf vrf1
address-family ipv4 unicast
route-target export 1:1 evpn
route-target import 1:1 evpn
ip route 22.0.0.0/24 13.0.0.1
ip route 22.0.0.0/24 13.0.0.2
ip route 22.0.0.0/24 13.0.0.3
ip route 22.0.0.0/24 13.0.0.4
ip route 22.0.0.0/24 13.0.0.5
!
address-family ipv6 unicast
route-target export 1:1 evpn
route-target import 1:1 evpn
ipv6 route 2000::/64 1001::2
ipv6 route 2000::/64 1002::2
ipv6 route 2000::/64 1003::2
ipv6 route 2000::/64 1004::2
ipv6 route 2000::/64 1005::2
!
```

VRF BFD on Static Route

Provides an option at the tenant VRF level so static route BFD timers can be used.

VRF Create

```
efa tenant vrf create --name <vrf-name> --tenant <tenant-name>
--ipv6static-route-next-hop <destination, next-hop>
--ipv4static-route-next-hop <destination, next-hop>
--ipv6static-route-bfd <destination-ip, source-ip, min-tx, min-rx, multiplier>
--ipv4static-route-bfd < destination-ip, source-ip, min-tx, min-rx, multiplier>
```

Example:

```
efa tenant vrf create --name red --tenant tenant11
--ipv6static-route-bfd 1001::2,1001::1,100,200,5
--ipv6static-route-bfd 1002::2, 1002::1
--ipv4static-route-bfd 13.0.0.1, 13.0.0.9,200,300,6
--ipv4static-route-bfd 13.0.0.2, 13.0.0.10
```

Switch Config

```
vrf vrf1
rd 172.31.254.13:1
evpn irb ve 8192
address-family ipv4 unicast
route-target export 1:1 evpn
route-target import 1:1 evpn
ip route 22.0.0.0/24 13.0.0.1
ip route 22.0.0.0/24 13.0.0.2
ip route 22.0.0.0/24 13.0.0.3
ip route 22.0.0.0/24 13.0.0.4
ip route 22.0.0.0/24 13.0.0.5
ip route static bfd 13.0.0.1 13.0.0.9 interval 200 min-rx 300 multiplier 6
ip route static bfd 13.0.0.2 13.0.0.10
!
address-family ipv6 unicast
route-target export 1:1 evpn
route-target import 1:1 evpn
ipv6 route 2000::/64 1001::2
ipv6 route 2000::/64 1002::2
ipv6 route 2000::/64 1003::2
ipv6 route 2000::/64 1004::2
ipv6 route 2000::/64 1005::2
ipv6 route static bfd 1001::2 1001::1 interval 100 min-rx 200 multiplier 5
ipv6 route static bfd 1002::2 1002::1
!
!
```

VRF Local-ASN

At the tenant VRF level, this option allows for providing local-asn per tenant VRF.

EFA Provisioning

```
efa tenant vrf create --name <vrf-name> --tenant <tenant-name> --local-asn <local-as-for-vrf>

efa tenant vrf update --name <vrf-name> --tenant <tenant-name> --operation <local-asn-add|local-asn-delete> --local-asn <value>
```

Note
The local-asn support on IPv6 AF must be checked.

Switch Config

leaf-9250-173# sh run router bgp router bgp local-as 4200000000 capability as4-enable fast-external-fallover bfd interval 100 min-rx 100 multiplier 3 neighbor 2001:11:1::2 remote-as 4200000001 neighbor 2001:11:1::2 bfd neighbor 2001:12:1::2 remote-as 4200000001 neighbor 2001:12:1::2 bfd neighbor 2001:91:1::1 remote-as 4230000000 neighbor 10.20.20.10 remote-as 4200000000 neighbor 10.20.20.10 next-hop-self neighbor 10.20.20.10 bfd neighbor 11.1.0.2 remote-as 4200000001 neighbor 11.1.0.2 bfd neighbor 12.1.0.2 remote-as 4200000001 neighbor 12.1.0.2 bfd neighbor 91.1.0.1 remote-as 4230000000 !	address-family ipv4 unicast vrf vrf1 local-as 4210000001 redistribute connected neighbor 11.1.1.2 remote-as 4220000001 neighbor 11.1.1.2 bfd neighbor 12.1.1.2 remote-as 4220000001 neighbor 12.1.1.2 bfd neighbor 21.1.1.1 remote-as 4210000001 neighbor 21.1.1.1 next-hop-self neighbor 91.1.1.1 remote-as 4230000001 maximum-paths 2 ! address-family ipv4 unicast vrf vrf10 local-as 4210000010 redistribute connected neighbor 11.1.10.2 remote-as 4220000010 neighbor 11.1.10.2 bfd neighbor 12.1.10.2 remote-as 4220000010 neighbor 12.1.10.2 bfd neighbor 21.1.10.1 remote-as 4210000010 neighbor 21.1.10.1 next-hop-self neighbor 91.1.10.1 remote-as 4230000010 maximum-paths 2 !
leaf-9250-173# show running-config router bgp address-family ipv4 unicast vrf vrf1 router bgp address-family ipv4 unicast vrf vrf1 local-as 4210000001 maximum-paths 2 !	leaf-9250-173# show running-config router bgp address-family ipv6 unicast vrf vrf1 router bgp address-family ipv6 unicast vrf vrf1 redistribute connected maximum-paths 2 !

VRF Graceful Restart

Tenant service configures the graceful restart under each tenant vrf ipv4/v6 unicast address-family.

Note
There is no user input needed for this configuration.

Switch Config

router bgp local-as 4200000000 capability as4-enable fast-external-fallover bfd interval 100 min-rx 100 multiplier 3 address-family ipv4 unicast maximum-paths 8 **graceful-restart** ! address-family ipv4 unicast vrf vrf1 local-as 4210000001 **graceful-restart** redistribute connected maximum-paths 2 ! address-family ipv4 unicast vrf vrf2 local-as 4210000002 **graceful-restart** redistribute connected maximum-paths 2 !	address-family ipv6 unicast redistribute connected maximum-paths 8 **graceful-restart** ! address-family ipv6 unicast vrf vrf1 **graceful-restart** redistribute connected maximum-paths 2 ! address-family ipv6 unicast vrf vrf2 **graceful-restart** redistribute connected m

CEP Reload Delay

Reload delay configuration is pushed onto the necessary ports during EPG create and update operations.

Note
There is no user input needed for this configuration.

Switch Config

```
interface Ethernet 0/15
reload-delay enable 90
switchport
switchport mode trunk
switchport trunk allowed vlan add 1001
switchport trunk tag native-vlan
no shutdown
!
```

Enable Cluster Tracking on CEP Interfaces

By default, EFA enables reload-delay configuration on all CEP interfaces. Reload-delay and cluster-tracking configurations are mutually exclusive.

When cluster tracking is enabled, an interface can track the state of an MCT (multi-chassis tunnel) cluster and divert traffic to alternative paths when a cluster is down for reasons such as maintenance mode.

To enable cluster tracking on CEP (cluster edge port) interfaces, you must remove the reload-delay configuration. For more information about reload-delay, see CEP Reload Delay on page 102.

1. Remove the reload-delay configuration on an interface.

   ```
   efa inventory device execute-cli --ip 10.18.120.187
   --command "Interface ethernet 0/1, no reload-delay enable" --config
   ```

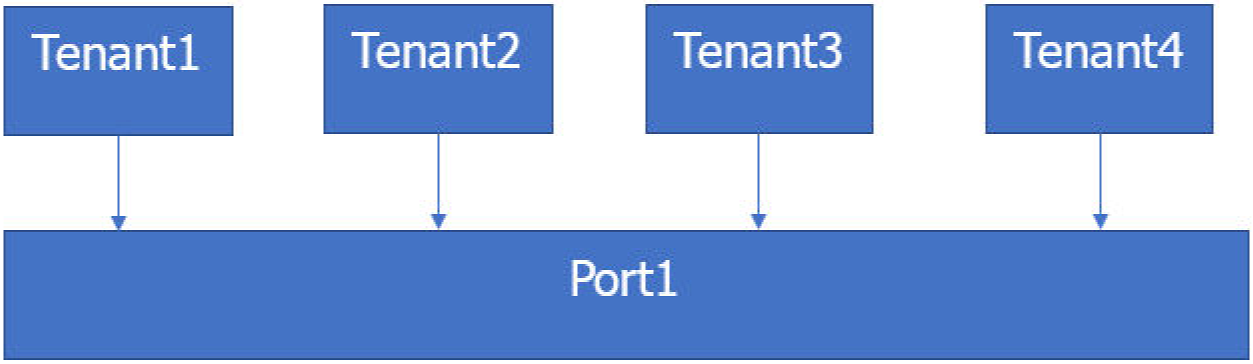
2. Configure cluster tracking.

   ```
   efa inventory device execute-cli --ip 10.18.120.187
   --command "Interface ethernet 0/1, cluster-track" --config
   ```

Sharing Resources Across Tenants Using "Shared Tenant"

1. Resource: Phy port, l2-vni-range, l3-vni-range, vlan-range, num-vrf.
2. Entity : PO, VRF, EPG
3. Ownership of resources will continue to stay per tenant.
4. Tenant construct will have a new attribute called “role = shared” and the tenant holds the resources/entities that CAN be shared across ALL the tenants.
5. Resources/entities owned/created by the shared tenant will be available for use by ALL the other tenants and not with specific set of tenants.
6. Tenant service can have one shared tenant to service all the shared resources.
7. Shared tenant can own the resources: Ports, L3VNI
8. Shared tenant can create the entities: Pos, VRFs
9. Shared tenant will not be able to create EPG.
10. Non-shared tenant cannot use the ports owned by shared tenant if the ports are already part of PO.
11. Non-shared tenant cannot create the PO using the ports owned by the shared tenant.

Shared Port Usecase (L2 Handover)



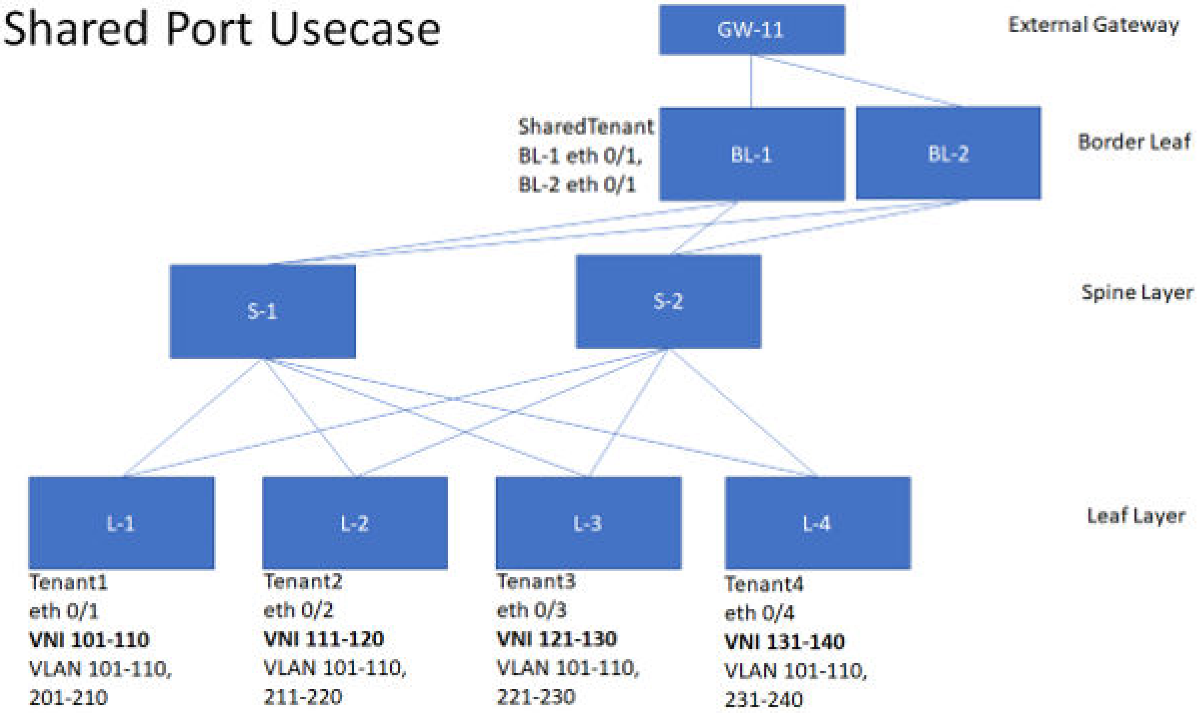


Figure 19: Shared Port Usecase

EFA Provisioning

```
efa tenant create --name tenant1 --l2-vni-range 101-110 --vlan-range 101-110,201-210 --
port L-1[0/1]

efa tenant create --name tenant2 --l2-vni-range 111-120 --vlan-range 101-110,211-220 --
port L-2[0/2]

efa tenant create --name tenant3 --l2-vni-range 121-130 --vlan-range 101-110,221-230 --
port L-3[0/3]

efa tenant create --name tenant4 --l2-vni-range 131-140 --vlan-range 101-110,231-240 --
port L-4[0/4]
```

```
efa tenant create --name SharedTenant --port BL-1[0/1],BL-2[0/1] --role shared
```

```
efa tenant epg create --name ten1epg1 --tenant tenant1 --port L-1[0/1] --switchport-mode
trunk --ctag-range 101-110 --l2-vni 101:101 --l2-vni 102:102 ….. --l2-vni 110:110

efa tenant epg create --name ten2epg1 --tenant tenant2 --port L-2[0/2] --switchport-mode
trunk --ctag-range 101-110 --l2-vni 101:111 --l2-vni 102:112 ….. --l2-vni 110:120

efa tenant epg create --name ten3epg1 --tenant tenant3 --port L-3[0/3] --switchport-mode
trunk --ctag-range 101-110 --l2-vni 101:121 --l2-vni 102:122 ….. --l2-vni 110:130

efa tenant epg create --name ten4epg1 --tenant tenant4 --port L-4[0/4] --switchport-mode
trunk --ctag-range 101-110 --l2-vni 101:131 --l2-vni 102:132 ….. --l2-vni 110:140
```

```
efa tenant epg create --name ten1epg2 --tenant tenant1 --port BL-1[0/1],BL-2[0/1] --
switchport-mode trunk --ctag-range 201-210 --l2-vni 201:101 --l2-vni 202:102 ….. --l2-vni
210:110

efa tenant epg create --name ten2epg2 --tenant tenant2 --port BL-1[0/1],BL-2[0/1] --
switchport-mode trunk --ctag-range 211-220 --l2-vni 211:111 --l2-vni 212:112 ….. --l2-vni
220:120

efa tenant epg create --name ten3epg2 --tenant tenant3 --port BL-1[0/1],BL-2[0/1] --
switchport-mode trunk --ctag-range 221-230 --l2-vni 221:121 --l2-vni 212:122 ….. --l2-vni
230:130

efa tenant epg create --name ten4epg2 --tenant tenant4 --port BL-1[0/1],BL-2[0/1] --
switchport-mode trunk --ctag-range 231-240 --l2-vni 231:131 --l2-vni 212:132 ….. --l2-vni
240:140
```

Shared PO Usecase (L2 Handover)

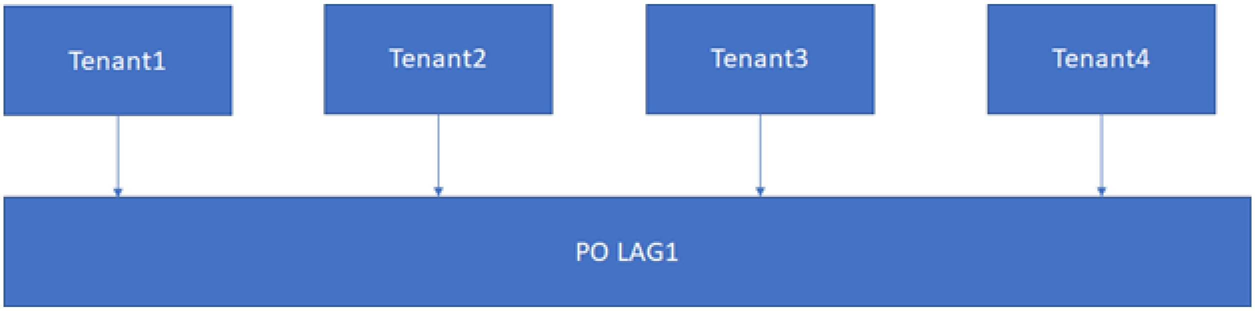


Figure 20: Shared PO Usecase (L2 Handover)

EFA Provisioning

```
efa tenant create --name SharedTenant --port BL-1[0/1],BL-2[0/1] --l3-vni-range 1001-1010
--vrf-count 10 --role shared
efa tenant vrf create --name red --tenant SharedTenant

efa tenant epg create --name ten1epg1 --tenant tenant1 --port L-1[0/1] --switchport-mode
trunk --ctag-range 101-102 --l2-vni 101:101 --l2-vni 102:102 --anycast-ip
101:10.10.10.1/24 --vrf red --l3-vni 1001

efa tenant epg create --name ten2epg1 --tenant tenant2 --port L-2[0/2] --switchport-mode
trunk --ctag-range 101-102 --l2-vni 101:111 --l2-vni 102:112 --anycast-ip
101:10.10.11.1/24 --vrf red --l3-vni 1001

efa tenant epg create --name ten3epg1 --tenant tenant3 --port L-3[0/3] --switchport-mode
trunk --ctag-range 101-102 --l2-vni 101:121 --l2-vni 102:122 --anycast-ip
101:10.10.12.1/24 --vrf red --l3-vni 1001

efa tenant epg create --name ten1epg1 --tenant tenant4 --port L-4[0/4] --switchport-mode
trunk --ctag-range 101-102 --l2-vni 101:131 --l2-vni 102:132 --anycast-ip
101:10.10.13.1/24 --vrf red --l3-vni 1001
```

Shared PO Usecase (L3 Handoff)

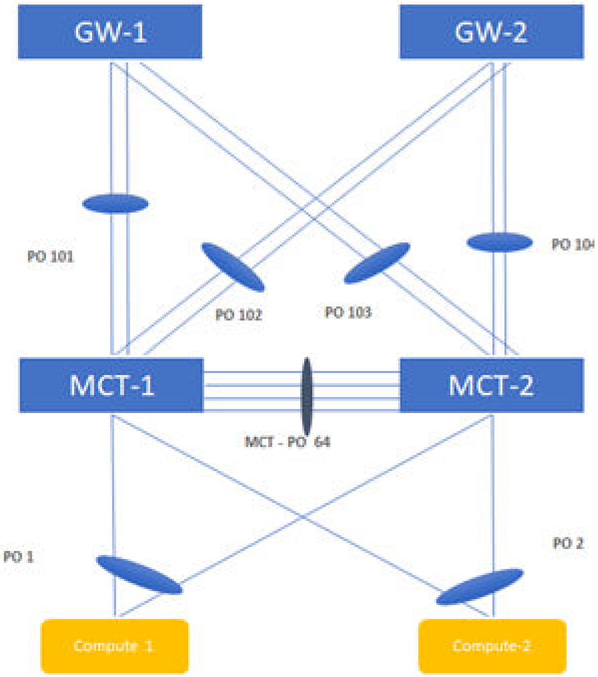
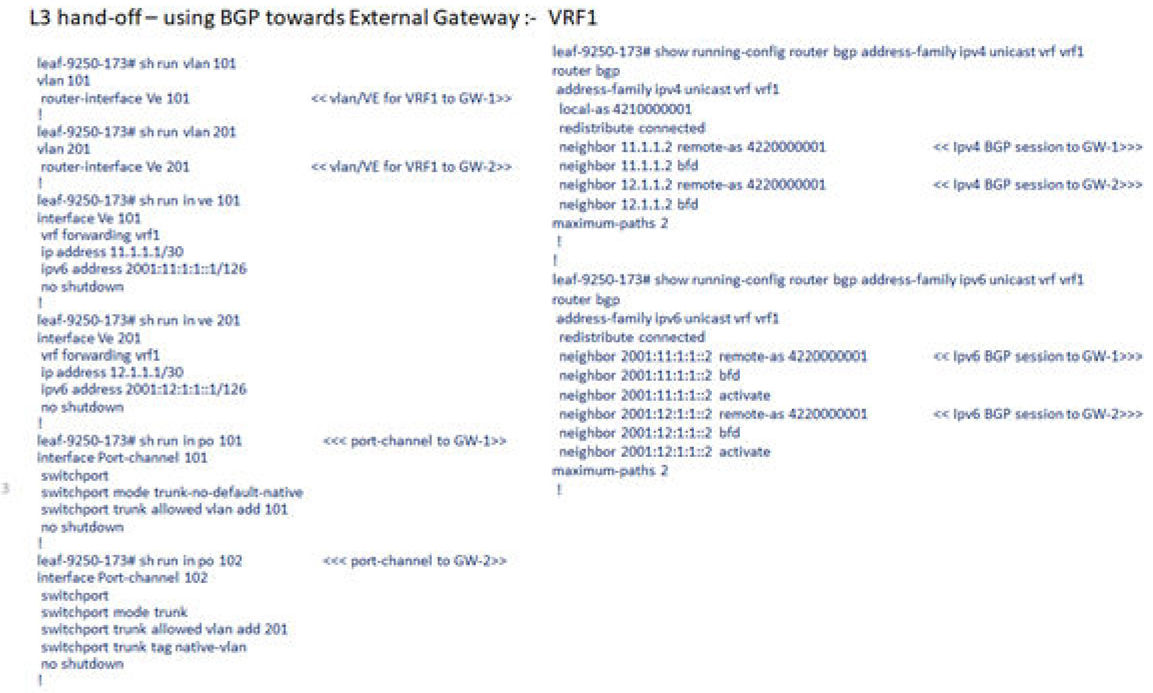


Figure 21: Topology Diagram

Switch Config

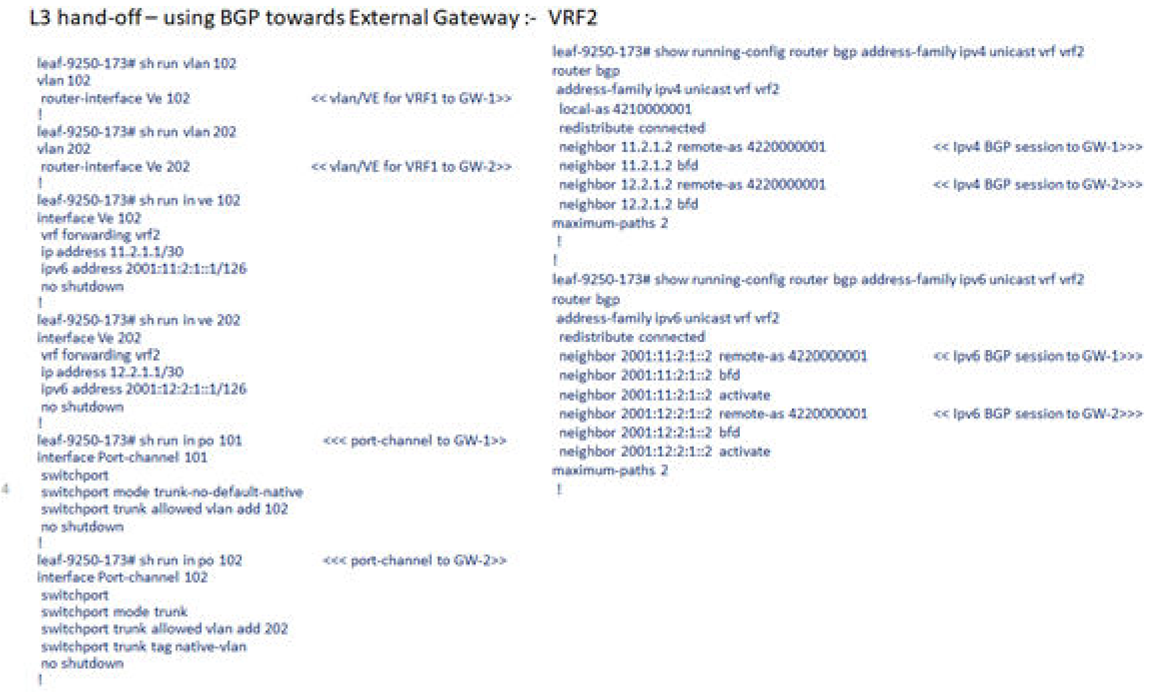


L3 hand-off – using BGP towards External Gateway :- VRF1

```
leaf-9250-173# sh run vlan 101
vlan 101
 router-interface Ve 101                          << vlan/VE for VRF1 to GW-1>>
!
leaf-9250-173# sh run vlan 201
vlan 201
 router-interface Ve 201                          << vlan/VE for VRF1 to GW-2>>
!
leaf-9250-173# sh run in ve 101
interface Ve 101
 vrf forwarding vrf1
 ip address 11.1.1.1/30
 ipv6 address 2001:11:1:1::1/126
 no shutdown
!
leaf-9250-173# sh run in ve 201
interface Ve 201
 vrf forwarding vrf1
 ip address 12.1.1.1/30
 ipv6 address 2001:12:1:1::1/126
 no shutdown
!
leaf-9250-173# sh run in po 101                    <<< port-channel to GW-1>>
interface Port-channel 101
 switchport
 switchport mode trunk-no-default-native
 switchport trunk allowed vlan add 101
 no shutdown
!
leaf-9250-173# sh run in po 102                    <<< port-channel to GW-2>>
interface Port-channel 102
 switchport
 switchport mode trunk
 switchport trunk allowed vlan add 201
 switchport trunk tag native-vlan
 no shutdown
!
```

```
leaf-9250-173# show running-config router bgp address-family ipv4 unicast vrf vrf1
router bgp
 address-family ipv4 unicast vrf vrf1
  local-as 4210000001
  redistribute connected
  neighbor 11.1.1.2 remote-as 4220000001              << Ipv4 BGP session to GW-1>>>
  neighbor 11.1.1.2 bfd
  neighbor 12.1.1.2 remote-as 4220000001              << Ipv4 BGP session to GW-2>>>
  neighbor 12.1.1.2 bfd
maximum-paths 2
 !
!
leaf-9250-173# show running-config router bgp address-family ipv6 unicast vrf vrf1
router bgp
 address-family ipv6 unicast vrf vrf1
  redistribute connected
  neighbor 2001:11:1:1::2 remote-as 4220000001        << Ipv6 BGP session to GW-1>>>
  neighbor 2001:11:1:1::2 bfd
  neighbor 2001:11:1:1::2 activate
  neighbor 2001:12:1:1::2 remote-as 4220000001        << Ipv6 BGP session to GW-2>>>
  neighbor 2001:12:1:1::2 bfd
  neighbor 2001:12:1:1::2 activate
maximum-paths 2
 !
```

Figure 22: L3 hand-off using BGP towards External Gateway: VRF1



L3 hand-off – using BGP towards External Gateway :- VRF2

```
leaf-9250-173# sh run vlan 102
vlan 102
 router-interface Ve 102                          << vlan/VE for VRF1 to GW-1>>
!
leaf-9250-173# sh run vlan 202
vlan 202
 router-interface Ve 202                          << vlan/VE for VRF1 to GW-2>>
!
leaf-9250-173# sh run in ve 102
interface Ve 102
 vrf forwarding vrf2
 ip address 11.2.1.1/30
 ipv6 address 2001:11:2:1::1/126
 no shutdown
!
leaf-9250-173# sh run in ve 202
interface Ve 202
 vrf forwarding vrf2
 ip address 12.2.1.1/30
 ipv6 address 2001:12:2:1::1/126
 no shutdown
!
leaf-9250-173# sh run in po 101                    <<< port-channel to GW-1>>
interface Port-channel 101
 switchport
 switchport mode trunk-no-default-native
 switchport trunk allowed vlan add 102
 no shutdown
!
leaf-9250-173# sh run in po 102                    <<< port-channel to GW-2>>
interface Port-channel 102
 switchport
 switchport mode trunk
 switchport trunk allowed vlan add 202
 switchport trunk tag native-vlan
 no shutdown
!
```

```
leaf-9250-173# show running-config router bgp address-family ipv4 unicast vrf vrf2
router bgp
 address-family ipv4 unicast vrf vrf2
  local-as 4210000001
  redistribute connected
  neighbor 11.2.1.2 remote-as 4220000001              << Ipv4 BGP session to GW-1>>>
  neighbor 11.2.1.2 bfd
  neighbor 12.2.1.2 remote-as 4220000001              << Ipv4 BGP session to GW-2>>>
  neighbor 12.2.1.2 bfd
maximum-paths 2
 !
!
leaf-9250-173# show running-config router bgp address-family ipv6 unicast vrf vrf2
router bgp
 address-family ipv6 unicast vrf vrf2
  redistribute connected
  neighbor 2001:11:2:1::2 remote-as 4220000001        << Ipv6 BGP session to GW-1>>>
  neighbor 2001:11:2:1::2 bfd
  neighbor 2001:11:2:1::2 activate
  neighbor 2001:12:2:1::2 remote-as 4220000001        << Ipv6 BGP session to GW-2>>>
  neighbor 2001:12:2:1::2 bfd
  neighbor 2001:12:2:1::2 activate
maximum-paths 2
 !
```

Figure 23: L3 hand-off using BGP towards External Gateway: VRF2

EFA Provisioning

```
efa tenant create --name tenant1 --l2-vni-range 1001-1010 --vlan-range 1001-1010 --port
BL-1[0/11],BL-2[0/11] --l3-vni-range 10001-10010 --vrf-count 10

efa tenant create --name tenant2 --l2-vni-range 1101-1110 --vlan-range 1101-1110 --port
BL-1[0/21],BL-2[0/21] --l3-vni-range 20001-20010 --vrf-count 10
```

```
efa tenant vrf create --name vrf1 --tenant Tenant1

efa tenant vrf create --name vrf2 --tenant Tenant2
```

```
efa tenant epg create --name ten1epg1 --tenant tenant1 --port BL-1[0/11] --switchport-
mode trunk --ctag-range 1001 --l2-vni 1001:1001 --anycast-ip 1001:10.10.10.1/24 --vrf
vrf1 --l3-vni 1001

efa tenant epg create --name ten2epg1 --tenant tenant2 --port BL-1[0/21] --switchport-
mode trunk --ctag-range 1101 --l2-vni 1101:1101 --anycast-ip 1101:10.10.11.1/24 --vrf
vrf2 --l3-vni 1002
```

```
efa tenant create --name SharedTenant --port BL-1[0/1-8],BL-2[0/1-8] --role shared
```

```
efa tenant po create --name po101 --tenant SharedTenant --speed 10Gbps --negotiation
active --port BL-1[0/1],BL-1[0/2]

efa tenant po create --name po102 --tenant SharedTenant --speed 10Gbps --negotiation
active --port BL-1[0/3],BL-1[0/4]
```

VRF1

```
efa tenant epg create --name ten1epg2 --tenant tenant1 --type l3-handover --po po101 --
switchport-mode trunk --ctag-range 101 --vrf vrf1  --local-ipv4-address 11.1.1.1/30 --
local-ipv6-address 2001:11:1:1::1/126 --remote-ipv4-address 11.1.1.2 --remote-ipv6-
address 2001:11:1:1::2 --remote-as 4220000001 --bfd --bfd-interval 100 --bfd-min-rx 200 --
bfd-multiplier 10
```

```
efa tenant epg create --name ten1epg3 --tenant tenant1 --type l3-handover --po po102 --
switchport-mode trunk --ctag-range 201 --vrf vrf1 --local-ipv4-address 12.1.1.1/30 --
local-ipv6-address 2001:12:1:1::1/126 --remote-ipv4-address 12.1.1.2 --remote-ipv6-
address 2001:12:1:1::2 --remote-as 4220000001 --bfd --bfd-interval 100 --bfd-min-rx 200 --
bfd-multiplier 10
```

VRF2

```
efa tenant epg create --name ten2epg2 --tenant tenant2 --type l3-handover --po po101 --
switchport-mode trunk --ctag-range 102 --vrf vrf2  --local-ipv4-address 11.2.1.1/30 --
local-ipv6-address 2001:11:2:1::1/126 --remote-ipv4-address 11.2.1.2 --remote-ipv6-
address 2001:11:1:1::2 --remote-as 4220000001 --bfd --bfd-interval 100 --bfd-min-rx 200 --
bfd-multiplier 10
```

```
efa tenant epg create --name ten2epg3 --tenant tenant2 --type l3-handover --po po102 --
switchport-mode trunk --ctag-range 202 --vrf vrf2 --local-ipv4-address 12.2.1.1/30 --
local-ipv6-address 2001:12:2:1::1/126 --remote-ipv4-address 12.2.1.2 --remote-ipv6-
address 2001:12:2:1::2 --remote-as 4220000001 --bfd  --bfd-interval 100 --bfd-min-rx 200
--bfd-multiplier 10
```

BGP as a Service

BGP as a service allows creation and deletion of BGP neighbors on a given fabric device. The VRF will need to created on the fabric device via EPG create/update prior to the BGP neighbours.

BGP Service Create

```
efa tenant service bgp create --name <service-name> --tenant <tenant-name>

--ipv4-unicast-neighbor <device-ip,vrf-name:ipv4-neighbor,remote-as,bfd-enable(t/f),bfd-
interval,bfd-rx,bfd-mult>

--ipv6-unicast-neighbor <device-ip,vrf-name:ipv4-neighbor,remote-as,bfd-enable(t/f),bfd-
interval,bfd-rx,bfd-mult>
```

Example:

```
efa tenant service bgp create --name bgpservice1 --tenant tenant1 --ipv4-unicast-neighbor
10.24.80.150,red:10.20.30.40,5000
```

BGP Service Show

Example:

```
efa tenant service bgp show
```

Name: bgpservice1
Tenant: tenant1
State: bs-state-created

Device IP	VRF	AFI	SAFI	REMOTE IP	REMOTE ASN	BFD Enabled	BFD Interval	BFD Rx	BFD Multiplier	Dev-state	App-state
10.24.80.150	red	ipv4	unicast	10.20.30.40	5000	false	0	0	0	provisioned	cfg-in-sync

BGP Service Update: peer-add

```
efa tenant service bgp update --name <service-name> --tenant <tenant-name>
--operation peer-add --ipv4-unicast-neighbor <nhbr-info> --ipv6-unicast-neighbor <nhbr-
info>
```

Example:

```
efa tenant service bgp update --name bgpservice1 --tenant tenant1 --operation peer-add --
ipv6-unicast-neighbor 10.24.80.150,red:10::40,5000
```

Name: bgpservice1
Tenant: tenant1
State: bs-state-created

Device IP	VRF	AFI	SAFI	REMOTE IP	REMOTE ASN	BFD Enabled	BFD Interval	BFD Rx	BFD Multiplier	Dev-state	App-state
10.24.80.150	red	ipv6	unicast	10::40	5000	false	0	0	0	provisioned	cfg-in-sync
10.24.80.150	red	ipv4	unicast	10.20.30.40	5000	false	0	0	0	provisioned	cfg-in-sync

BGP Service Update : peer-delete

```
efa tenant service bgp update --name <service-name> --tenant <tenant-name>
--operation peer-delete --ipv4-unicast-neighbor <nhbr-info> --ipv6-unicast-neighbor <nhbr-
info>
```

Example:

```
efa tenant service bgp update --name bgpservice1 --tenant tenant1 --operation peer-delete
--ipv4-unicast-neighbor 10.24.80.150,red:10.20.30.40
```

BGP Service Delete

```
efa tenant service bgp delete --name <service-name> --tenant <tenant-name>
```

Example:

```
efa tenant service bgp delete --name bgpservice1 --tenant tenant1
```

Local IP Configuration

You can add and delete local IP address configurations.

You can add and delete local IP address configurations during the following operations:

- Creating EPGs
- Adding or deleting CTAG ranges
- Adding or deleting VRFs

The Local IP address is configured on the VE interface assigned to a particular tenant network. You can select different local IP addresses for each device in a tenant network.

EPG create with local IP configuration

```
efa tenant epg create --name ten1epg1 --tenant tenant1 --vrf red
--switchport-mode trunk --ctag-range 11 --anycast-ip 11:10.10.11.1/24
--port 10.24.80.150[0/1],10.24.80.151[0/1]
--local-ip 11,10.24.80.150:11.22.33.41/24 --local-ip 11,10.24.80.151:11.22.34.41/24

efa tenant epg show
Name: ten1epg1
Tenant: tenant1
Description:
Type: extension
Ports : 10.24.80.151[0/1]
      : 10.24.80.150[0/1]
Port Property : switchport mode     : trunk
              : native-vlan-tagging : false
NW Policy: ctag-range       :11
              : vrf                 : red
              : vrf-State           : vrf-device-created
              : vrf-Device-State    : provisioned
              : vrf-App-State       : cfg-refreshed
              : l3-vni              : 8190
Network Property [Flags : * - Native Vlan]

| Ctag | L2-Vni | Anycast-IPv4 | Anycast-IPv6 | BD-name |                Local IP (Device-
IP->Local-IP)                |  Dev-state  |    App-state    |
+------+--------+-------------+-------------+---------
+--------------------------------------------------------------+-------------
+-----------------+
| 11   | 11     |10.10.11.1/24 |              |         |    10.24.80.151-
>11.22.34.41/24                              | provisioned | cfg-refreshed   |
|      |        |              |              |
|                                                              |
|                 |
|      |        |              |              |         |    10.24.80.150-
>11.22.33.41/24                              |             |                 |
```

EPG update local-ip-delete operation

```
efa tenant epg update --name epgv20 --tenant tenant1 --operation
local-ip-delete --local-ip 11,10.24.80.150:11.22.33.41/24

efa tenant epg show
Name          : epgv20
Tenant        : t3
Description   :
Type          : l3-hand-off
Ports         : 10.20.50.209[0/27]
POs           : posv9
Port Property : switchport mode     : trunk
              : native-vlan-tagging : false
NW Policy     : ctag-range          : 201-202
              : vrf                 : vrfv20
              : l3-vni              : 5110

Network Property [Flags : * - Native Vlan]
+------+--------+--------------+--------------+---------
+--------------------------------------------------------------+-------------
+-----------------+
| Ctag | L2-Vni | Anycast-IPv4 | Anycast-IPv6 | BD-name |                Local IP (Device-
IP->Local-IP)                |  Dev-state  |    App-state    |
+------+--------+--------------+--------------+---------
+--------------------------------------------------------------+-------------
+-----------------+
| 201  | 201    |              |              |         |    10.20.50.209-
>44.4.4.5/24                                 | provisioned | cfg-in-sync     |
|      |        |              |              |         |
4444:44::5/120                              |             |                 |
|      |        |              |              |         |    10.20.50.208-
>44.4.4.4/24                                 |             |                 |
|      |        |              |              |         |
4444:44::4/120                              |             |                 |
+------+--------+--------------+--------------+---------
+--------------------------------------------------------------+-------------
+-----------------+
| 202  | 202    |              |              |         |    10.20.50.209-
>44.4.5.5/24                                 | provisioned | cg-in-sync      |
|      |        |              |              |         |
4444:45::5/120                              |             |                 |
+------+--------+--------------+--------------+---------
+--------------------------------------------------------------+-------------
+-----------------+

For 'unstable' entities, run 'efa tenant po/vrf show' for details
==========================================================================================
=====================================================================
```

EPG update local-ip-add operation

```
efa tenant epg update --name ten1epg1 --tenant tenant1
--operation local-ip-add --local-ip 11,10.24.80.150:11.22.33.41/24

efa tenant epg show
```

```
Name: ten1epg1
Tenant: tenant1
Description:
Type: extension
Ports : 10.24.80.151[0/1]
      : 10.24.80.150[0/1]
Port Property : switchport mode     : trunk
              : native-vlan-tagging : false
NW Policy: ctag-range       :11
              : vrf                 : red
              : vrf-State           : vrf-device-created
              : vrf-Device-State    : provisioned
              : vrf-App-State       : cfg-refreshed
              : l3-vni              : 8190

Network Property [Flags : * - Native Vlan]
+------+--------+--------------+--------------+---------
+---------------------------------------------------------------+-------------
+------------------+
| Ctag | L2-Vni | Anycast-IPv4 | Anycast-IPv6 | BD-name |                Local IP (Device-
IP->Local-IP)                |  Dev-state  |    App-state    |
+------+--------+--------------+--------------+---------
+---------------------------------------------------------------+-------------
+------------------+
| 11   | 11     |              |              |         |  10.24.80.151-
>11.22.34.41/24                                | provisioned | cfg-in-sync     |
|      |        |              |              |
|                                                               |
|                 |
|      |        |              |              |         |    10.24.80.150-
>11.22.33.41/24                              |             |
```

Tenant Configuration Reconciliation

The guidelines for reconciliation are as follows:

- EFA is the Single Source of Truth
- EFA modifies switch configurations only when there are conflicts with EFA intended configurations
- EFA always attempts to merge its intended configurations with Switch Configurations.

The following table lists the supported reconciliation cases and actions:

Reconciliation Case	Action
Missing config on the switch	Push the EFA intended config to the switch
Partial config on the switch	Merge the EFA intended config to the switch
Conflicting config on the switch	Replace with EFA intended config

The following sections show how various switch configurations are handled during reconciliation by Tenant Services.

VLAN Missing

Configuration	Details
EFA Intended Config	vlan 900 router-interface Ve 900 suppress-arp !
Switch Current Config	Empty
Action	Missing Config, Push Config to Switch

VLAN Merge

Configuration	Details
EFA Intended Config	vlan 900 **router-interface Ve 900** **suppress-arp** !
Switch Current Config	vlan 900 !
Action	Partial Config, Merge Config to Switch

VLAN Conflict

Configuration	Details
EFA Intended Config	vlan 900 **router-interface Ve 900** suppress-arp !
Switch Current Config	vlan 900 **router-interface Ve 800** suppress-arp !
Action	Conflict Config, Replace Config (Specific Config)

Interface Missing

Configuration	Details
EFA Intended Config	interface Ethernet 0/1 switchport switchport mode trunk switchport trunk allowed vlan add 900 no switchport trunk tag native-vlan no shutdown !
Switch Current Config	Empty
Action	Missing Config, Push Config to Switch

Interface Merge

Configuration	Details

EFA Intended Config	`interface Ethernet 0/1`
	`switchport`
	`switchport mode trunk`
	`switchport trunk allowed vlan add 900`
	`no switchport trunk tag native-vlan`
	`no shutdown`
	`!`

Switch Current Config	`interface Ethernet 0/1`
	`switchport`
	`switchport mode trunk`
	`switchport trunk allowed vlan add 900`
	`!`

Action	Partial Config, Merge Config to Switch

Configuration	Details

EFA Intended Config	`interface Ethernet 0/1`
	`switchport`
	`switchport mode trunk`
	`switchport trunk allowed vlan add 900-920`
	`no switchport trunk tag native-vlan`
	`no shutdown`
	`!`

Switch Current Config	`interface Ethernet 0/1`
	`switchport`
	`switchport mode trunk`
	`switchport trunk allowed vlan add 900-910`
	`no switchport trunk tag native-vlan`
	`no shutdown`
	`!`

Action	Partial Config, Merge Config to Switch

Configuration	Details

EFA Intended Config	`interface Ethernet 0/1`
	`switchport`
	`switchport mode trunk`
	`switchport trunk allowed vlan add 900`
	`no switchport trunk tag native-vlan`
	`no shutdown`
	`!`

Switch Current Config	`interface Ethernet 0/1`
	`switchport`
	`switchport mode access`

Configuration	Details

	`switchport access vlan 900`
	`!`

Action	Conflict Config, Merge the Config to Switch

Configuration	Details

EFA Intended Config	`interface Ethernet 0/1`
	`switchport`
	`switchport mode trunk`
	`switchport trunk allowed vlan add 900-901`
	`switchport trunk native-vlan 901`
	`no switchport trunk tag native-vlan`
	`no shutdown`
	`!`

Switch Current Config	`interface Ethernet 0/1`
	`switchport`
	`switchport mode trunk`
	`switchport trunk allowed vlan add 900-901`
	`switchport trunk native-vlan 900`
	`no switchport trunk tag native-vlan`
	`no shutdown`
	`!`

Action	Conflict Config, Merge the Config (specific Config)

Port Channel Missing

Configuration	Details

EFA Intended Config	`interface Port-channel 1`
	`description Port-channel po1`
	`switchport`
	`switchport mode trunk`
	`switchport trunk allowed vlan add 2000-2001`
	`no switchport trunk tag native-vlan`
	`no shutdown`
	`!`

Switch Current Config	Empty
Action	Missing Config, Push Config to Switch

-
-

Port Channel Merge

Configuration	Details
EFA Intended Config	interface Port-channel 1 description Port-channel po1 switchport switchport mode trunk switchport trunk allowed vlan add 2000-2001 no switchport trunk tag native-vlan no shutdown !
Switch Current Config	interface Port-channel 1 description Port-channel po1 !
Action	Partial Config, Merge Config to Switch

Configuration	Details
EFA Intended Config	**interface Port-channel 1** description Port-channel po1 switchport switchport mode trunk switchport trunk allowed vlan add 2000-2001 no switchport trunk tag native-vlan no shutdown !
Switch Current Config	interface Port-channel 2 description Port-channel po1 switchport switchport mode trunk switchport trunk allowed vlan add 2000-2001 no switchport trunk tag native-vlan no shutdown !
Action	Conflict Config, Merge Config (Specific Config)

VRF Missing

Configuration	Details
EFA Intended Config	vrf v1 rd 172.31.254.10:102 evpn irb ve 8191 address-family ipv4 unicast route-target export 102:102 evpn route-target import 102:102 evpn ! !
Switch Current Config	Empty
Action	Missing Config, Push Config to Switch

VRF Merge

Configuration	Details

EFA Intended Config	`vrf v1`
	`rd 172.31.254.10:102`
	`evpn irb ve 8191`
	`address-family ipv4 unicast`
	`route-target export 102:102 evpn`
	`route-target import 102:102 evpn`
	`!`
	`!`

Switch Current Config	`vrf v1`
	`rd 172.31.254.10:102`
	`!`

Action	Partial Config, Merge Config to Switch

Configuration	Details

EFA Intended Config	`vrf v1`
	`rd 172.31.254.10:102`
	`evpn irb ve 8191`
	`address-family ipv4 unicast`
	`route-target export 102:102 evpn`
	`route-target import 102:102 evpn`
	`!`
	`!`

Switch Current Config	`vrf v1 rd 172.31.254.10:102 evpn irb ve 8191 address-family ipv4 unicast route-target export 102:102 evpn ! !`
Action	Partial Config, Merge Config to Switch

VRF Conflict

Configuration	Details

EFA Intended Config	`vrf v1`
	`rd 172.31.254.10:102`
	`evpn irb ve 23`
	`address-family ipv4 unicast`
	`route-target export 102:102 evpn`
	`route-target import 102:102 evpn`
	`!`
	`!`

Switch Current Config	`vrf v1`
	`rd 172.31.254.10:103`
	`evpn irb ve 23`
	`address-family ipv4 unicast`
	`route-target export 102:102 evpn`
	`route-target import 102:102 evpn`
	`!`
	`!`

Action	Conflict Config, Replace Config (Specific Config)

Configuration	Details

EFA Intended Config	`vrf v1`
	`rd 172.31.254.10:102`
	`evpn irb ve 8191`
	`address-family ipv4 unicast`
	`route-target export 102:102 evpn`
	`route-target import 102:102 evpn`
	`!`
	`!`

Switch Current Config	`vrf v1`
	`rd 172.31.254.10:102`
	`evpn irb ve 8192`
	`address-family ipv4 unicast`
	`route-target export 103:103 evpn`
	`route-target import 103:103 evpn`
	`!`
	`!`

Action	Conflict Config, Replace Config (Specific Config)

InterfaceVe Missing

Configuration	Details

EFA Intended Config	`interface Ve 2000`
	`vrf forwarding v1`
	`ip anycast-address 10.10.10.10/24`
	`no shutdown`
	`!`

Switch Current Config	Empty
Action	Missing Config, Push Config to Switch

InterfaceVe Merge

Configuration	Details
EFA Intended Config	interface Ve 2000 vrf forwarding v1 **ip anycast-address 10.10.10.10/24** **no shutdown** !
Switch Current Config	interface Ve 2000 vrf forwarding v1 !
Action	Partial Config, Merge Config to Switch

InterfaceVe Conflict

Configuration	Details
EFA Intended Config	interface Ve 2000 vrf forwarding v1 ip anycast-address 10.10.10.10/24 no shutdown !
Switch Current Config	interface Ve 2000 vrf forwarding v2 ip anycast-address 10.10.10.10/24 no shutdown !
Action	Conflict Config, Replace Config (Specific Config)

Configuration	Details
EFA Intended Config	interface Ve 2000 vrf forwarding v1 **ip anycast-address 10.10.10.10/24** no shutdown !
Switch Current Config	interface Ve 2000 vrf forwarding v1 ip anycast-address 11.11.11.11/24 no shutdown !
Action	Conflict Config, Replace full / specific Config

EVPN Missing

Configuration	Details

EFA Intended Config	`evpn default`
	`route-target both auto ignore-as`
	`rd auto`
	`duplicate-mac-timer 5 max-count 3`
	`bridge-domain add 4095`
	`vlan add 2000-2001`
	`!`

Switch Current Config	Empty
Action	Missing Config Push Config to Switch

EVPN Merge

Table 18:

Configuration	Details

EFA Intended Config	`evpn default`
	`route-target both auto ignore-as`
	`rd auto`
	`duplicate-mac-timer 5 max-count 3`
	`bridge-domain add 4095`
	`vlan add 2000-2001`
	`!`

Switch Current Config	`evpn default`
	`route-target both auto ignore-as`
	`rd auto`
	`duplicate-mac-timer 5 max-count 3`
	`bridge-domain add 4095`
	`vlan add 2000`
	`!`

Action	Partial Config, Merge specific Config to Switch

EVPN Conflict

Configuration	Details

EFA Intended Config	`evpn default`
	`route-target both auto ignore-as`
	`rd auto`
	`duplicate-mac-timer 5 max-count 3`
	`bridge-domain add 4095`
	`vlan add 2000-2001`
	`!`

Switch Current Config	`evpn fabric1`
	`!`

Action	Missing Config Push Config to Switch

Configuration	Details

EFA Intended Config	`evpn default`
	`route-target both auto ignore-as`
	`rd auto`
	`duplicate-mac-timer 5 max-count 3`
	`bridge-domain add 4095`
	`vlan add 2000-2001`
	`!`

Switch Current Config	`evpn default`
	`route-target both auto ignore-as`
	`rd auto`
	`duplicate-mac-timer 5 max-count 3`
	`bridge-domain add 4095`
	`vlan add 3500`
	`!`

Action	Conflict Config, Merge the Config (Specific Config)

Overlay Gateway Missing

Configuration	Details

EFA Intended Config	`overlay-gateway default`
	`ip interface Loopback 2`
	`map vni auto`
	`Activate`
	`!`

Switch Current Config	Empty
Action	Missing Config Push Config to Switch

Overlay Gateway Merge

Configuration	Details

EFA Intended Config	`overlay-gateway default`
	`ip interface Loopback 2`
	`map vlan 2000 vni 5000`
	`map vlan 2001 vni 5001`
	`map bridge-domain 4095 vni 8100`
	`activate`
	`!`

Switch Current Config	`overlay-gateway default`
	`ip interface Loopback 2`
	`map vlan 2000 vni 5000`
	`map bridge-domain 4095 vni 8100`
	`activate`
	`!`

Action	Partial Config, Merge specific Config to Switch

Overlay Gateway Conflict

Configuration	Details

EFA Intended Config	`overlay-gateway default`
	`ip interface Loopback 2`
	`map vlan 2000 vni 5000`
	`map bridge-domain 4095 vni 8100`
	`activate`
	`!`

Switch Current Config	`overlay-gateway fabric1`
	`!`

Action	Conflict Config, Replace Config (Specific Config)

Configuration	Details

EFA Intended Config	`overlay-gateway default`
	`ip interface Loopback 2`
	`map vlan 2000 vni 5000`
	`map bridge-domain 4095 vni 8100`
	`activate`
	`!`

Switch Current Config	`overlay-gateway default`
	`ip interface Loopback 2`
	`map vni auto`
	`activate`
	`!`

Action	Conflict Config, Replace Config (Specific Config)

Configuration	Details

EFA Intended Config	`overlay-gateway default`
	`ip interface Loopback 2`
	`map vlan 2000 vni 5000`
	`map bridge-domain 4095 vni 8100`
	`activate`
	`!`

Switch Current Config	`overlay-gateway default`
	`ip interface Loopback 2`
	`map vlan 2000 vni 5000`
	`map bridge-domain 4095 vni 8101`
	`activate`
	`!`

Action	Conflict Config, Replace Config (Specific Config)

Cluster Client Missing

Cluster Client reconciliation is supported only in SLX-OS releases earlier than SLX 20.1.1.

Configuration	Details

EFA Intended Config	`cluster nonclos-cluster-1`
	`peer 10.20.20.7`
	`peer-interface Port-channel 64`
	`peer-keepalive`
	`  auto`
	`!`
	`member vlan all`
	`member bridge-domain all`

Switch Current Config	Empty
Action	Missing Config, Push Config to Switch

Cluster Client Merge

Cluster Client reconciliation is supported only in SLX-OS releases earlier than SLX 20.1.1.

Configuration	Details

EFA Intended Config	`cluster nonclos-cluster-1`
	`peer 10.20.20.7`
	`peer-interface Port-channel 64`
	`peer-keepalive`
	`  auto`
	`!`
	`member vlan all`
	`member bridge-domain all`

Switch Current Config	`cluster default-cluster-1 1`
	`peer-interface Port-channel 64`
	`peer 10.20.20.4`
	`df-load-balance`
	`deploy`
	`client default-cluster-1 1`
	`!`

Action	Partial Config, Merge Specific Config to Switch

Cluster Client Conflict

Cluster Client reconciliation is supported only in SLX-OS releases earlier than SLX 20.1.1.

Configuration	Details

EFA Intended Config	`cluster nonclos-cluster-1`
	`peer 10.20.20.7`
	`peer-interface Port-channel 64`
	`peer-keepalive`
	`  auto`
	`!`
	`member vlan all`
	`member bridge-domain all`

Switch Current Config	`Cluster missing`
	`Or`
	`cluster default-cluster-2 2`

Action	Display validation error This is an unexpected behavior as Fabric Service must have already reconciled this entity.

Configuration	Details

EFA Intended Config	`cluster nonclos-cluster-1`
	`peer 10.20.20.7`
	`peer-interface Port-channel 64`
	`peer-keepalive`
	`  auto`
	`!`
	`member vlan all`
	`member bridge-domain all`

Switch Current Config	Empty
Action	Conflict Config, Replace Client Config (Verify)

Configuration	Details

EFA Intended Config	`cluster nonclos-cluster-1`
	`peer 10.20.20.7`
	`peer-interface Port-channel 64`
	`peer-keepalive`
	`  auto`
	`!`
	`member vlan all`
	`member bridge-domain all`

Switch Current Config	`cluster default-cluster-1 1`
	`peer-interface Port-channel 64`
	`peer 10.20.20.4`
	`df-load-balance`
	`deploy`
	`client default-cluster-1 1`
	`client-interface Port-channel 2`
	`Esi auto lacp`
	`deploy`

Configuration	Details
	`!` `!`
Action	Conflict Config, Replace Client Config (Verify)

Bridge Domain Missing

Configuration	Details

EFA Intended Config	`bridge-domain 1 p2mp`
	`router-interface Ve 4097`
	`logical-interface ethernet 0/11.100`
	`suppress-arp`
	`!`

Switch Current Config	Empty
Action	Missing Config, Push Config to Switch

Bridge Domain Merge

Configuration	Details

EFA Intended Config	`bridge-domain 1 p2mp`
	`router-interface Ve 4097`
	`logical-interface ethernet 0/11.100`
	`suppress-arp`
	`!`

Switch Current Config	`bridge-domain 1 p2mp`
	`router-interface Ve 4097`
	`!`

Action	Partial Config, Merge Config to Switch

Bridge Domain Conflict

Configuration	Details

EFA Intended Config	`bridge-domain 1 p2mp`
	`router-interface Ve 4097`
	`logical-interface ethernet 0/11.100`
	`suppress-arp`
	`!`

Switch Current Config	`bridge-domain 1 p2mp`
	`router-interface Ve 4098`
	`logical-interface ethernet 0/11.100`
	`suppress-arp`
	`!`

Action	Conflict Config, Replace Config (Specific Config)

LIF Missing

Configuration	Details

EFA Intended Config	`interface Ethernet 0/11`
	`switchport`
	`switchport mode trunk`
	`no switchport trunk tag native-vlan`
	`no shutdown`
	`logical-interface ethernet 0/11.100`
	`name 0/11.100`
	`vlan 100`
	`!`
	`!`

Switch Current Config	Empty
Action	Missing Config, Push Config to Switch

LIF Merge

Configuration	Details

EFA Intended Config	`interface Ethernet 0/11`
	`switchport`
	`switchport mode trunk`
	`no switchport trunk tag native-vlan`
	`no shutdown`
	`logical-interface ethernet 0/11.100`
	`name 0/11.100`
	`vlan 100`
	`!`
	`!`

Switch Current Config	`interface Ethernet 0/11`
	`switchport`
	`switchport mode trunk`
	`no switchport trunk tag native-vlan`
	`no shutdown`
	`logical-interface ethernet 0/11.100`
	`name 0/11.100`
	`!`
	`!`

Action	Partial Config, Merge Config to Switch

LIF Conflict

Configuration	Details

EFA Intended Config	`interface Ethernet 0/11`
	`switchport`
	`switchport mode trunk`
	`no switchport trunk tag native-vlan`
	`no shutdown`
	`logical-interface ethernet 0/11.100`
	`name 0/11.100`
	`vlan 100`
	`!`
	`!`

Switch Current Config	`interface Ethernet 0/11`
	`switchport`
	`switchport mode trunk`
	`no switchport trunk tag native-vlan`
	`no shutdown`
	`logical-interface ethernet 0/11.100`
	`name 0/11.100`
	`vlan 101`
	`!`
	`!`

Action	Conflict Config, Replace Config (verify once)

Port-Channel LACP Timeout

An option at the tenant port-channel level where an `lacp timeout short` configuration can be set on all the port-channel member interfaces and ICL port-channel members.

EFA Provisioning

```
efa tenant po create --name <po-name> --tenant <tenant-name> --speed <value> --
negotiation <value> --port <value> --number <value> --lacp-timeout <short|long>

efa tenant po update --name <po-name> --tenant <tenant-name> --operation <lacp-timeout-
update> --lacp-timeout <short|long>
```

Example:

```
efa tenant po create --name po1315 --speed 10Gbps --port 10.25.225.11[0/15],
10.25.225.46[0/15] --negotiation active --tenant tenant11 --lacp-timeout short
```

Switch Config

Avalanche-11#show running-config interface ethernet 0/15
interface Ethernet 0/15
channel-group 121 mode active type standard
lacp timeout short
no shutdown
!
Avalanche-46#show running-config interface ethernet 0/15
interface Ethernet 0/15
channel-group 121 mode active type standard
lacp timeout short

```
no shutdown
!
```

EFA Device Management

Periodic Device Discovery

Tenant and Fabric Services use periodic discovery to detect out-of-sync configurations on the devices. Fabric and Tenant Services act on the published events and update the database to reflect the status of the devices as in-sync and out-of-sync.

You can perform on-demand device discovery using the command line.

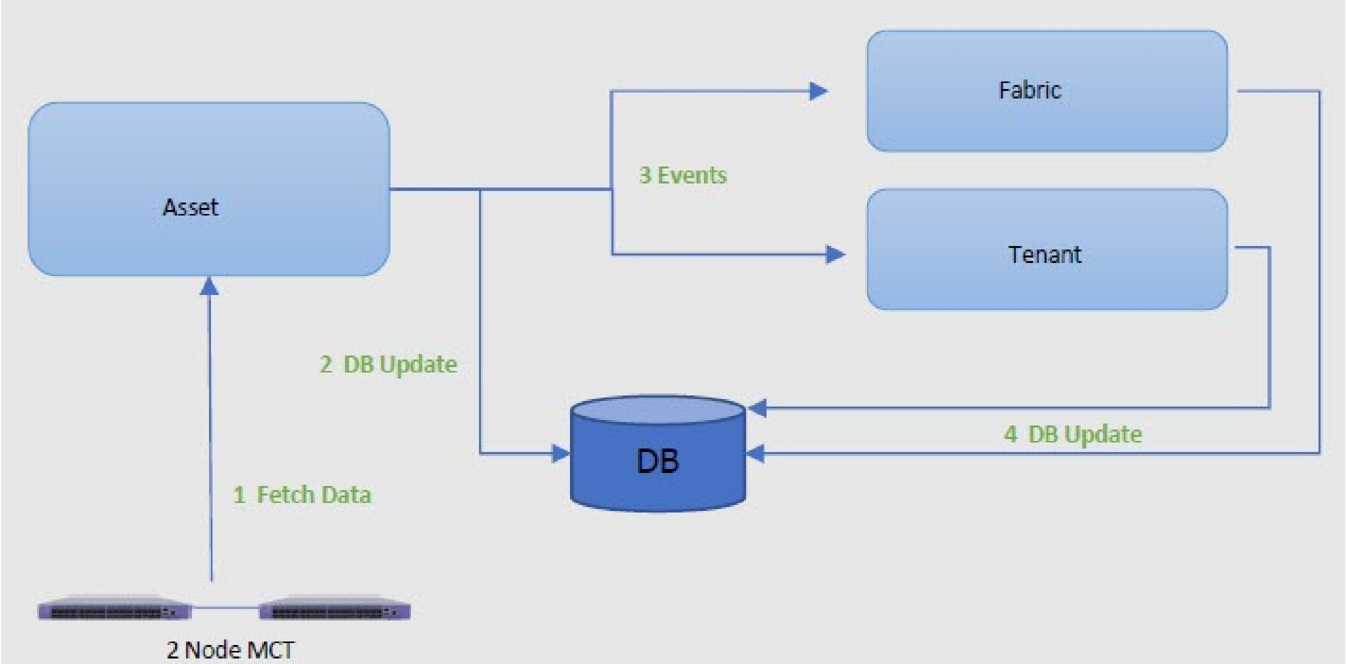


Figure 24: Device discovery and database updates

The Asset Service periodically discovers devices in the Fabric at an interval that you can configure. The default is every hour, with valid values ranging from 15 minutes to 24 hours. You can use the `efa`

inventory device discovery-time list command to view the current discovery interval for a device or for the Fabric. You can use the **efa inventory device discovery-time update** command to configure the discovery interval.

Device Image Management

Using maintenance mode, you can download firmware on one or more devices in the IP Fabric with the minimal disruption to data path traffic. Both Clos and Non-Clos Fabrics are supported. The maintenance mode feature is supported only on SLX devices running SLXOS 20.1.1 and later.

EFA supports the following firmware download features:

- Firmware download with maintenance mode supporting the following:
  - Asynchronously launched operations
  - Sanity and pre-install script check
  - Set convergence timeout, enable, and disable
  - Persisting the running configuration so that running configuration and maintenance mode configuration are preserved after reboot
  - Firmware download with no commit option to enable restoration of firmware to a previous version
- Firmware host registration, with support for register, update, delete, and list operations
- Firmware download preparation, with support for add, remove, and list operations
- Firmware download execute to start the firmware download with maintenance mode asynchronous operation
- Firmware download show, to display a table of devices in the Fabric and their corresponding status

Note
When a device is in maintenance mode, no configuration changes are allowed.

Limitations

- The device firmware must be SLX-OS 20.1.1 or later to support firmware download with maintenance mode for a hitless firmware upgrade.
- This feature assumes an existing host that contains SLX-OS firmware images ready to be downloaded.
- This feature should not be launched on a device where EFA TPVM is deployed.
- If you downgrade software from version 20.1.2a to 20.1.1, you must manually remove certificates.

Supported Devices

The SLX-OS firmware download with maintenance mode is supported on the following SLX devices running SLX-OS 20.1.1 and later.

- SLX 9540
- SLX 9640
- SLX 9150-48Y

- SLX 9150-48XT
- SLX 9250

Hitless Firmware Upgrade

A hitless firmware upgrade utilizes the maintenance mode switch feature to gracefully divert traffic away from the switch to alternate paths. The switch can be put into maintenance mode and a firmware upgrade can be performed. The switch can safely be rebooted and the new firmware activated without traffic loss. The switch can be taken out of maintenance mode and allow traffic on the newly upgraded switch.

Upgrade Super-Spine Firmware in Clos

1. Enabling maintenance mode on a super-spine involves BGP. The graceful_shutdown parameter will be sent to all the super-spine's underlay neighbors (all connected spines). Each neighbor will process the graceful_shutdown and refresh their routes to use the alternate path. Maintenance mode is enabled on the first super-spine and traffic is diverted over to the second super-spine.
2. The running-configuration is saved on the first super-spine to preserve all current configurations including the maintenance mode enable configuration.
3. The firmware on the first super-spine can now be upgraded and rebooted for firmware activation without traffic loss.
4. Once the new firmware is activated, maintenance mode can be disabled. The graceful_shutdown parameter is removed from all the underlay neighbors and traffic to the first super-spine is allowed again.
5. The running-config is persisted again to ensure the maintenance mode disabled state is retained.

   The same process can be carried out on the second super-spine to upgrade the firmware without traffic loss.

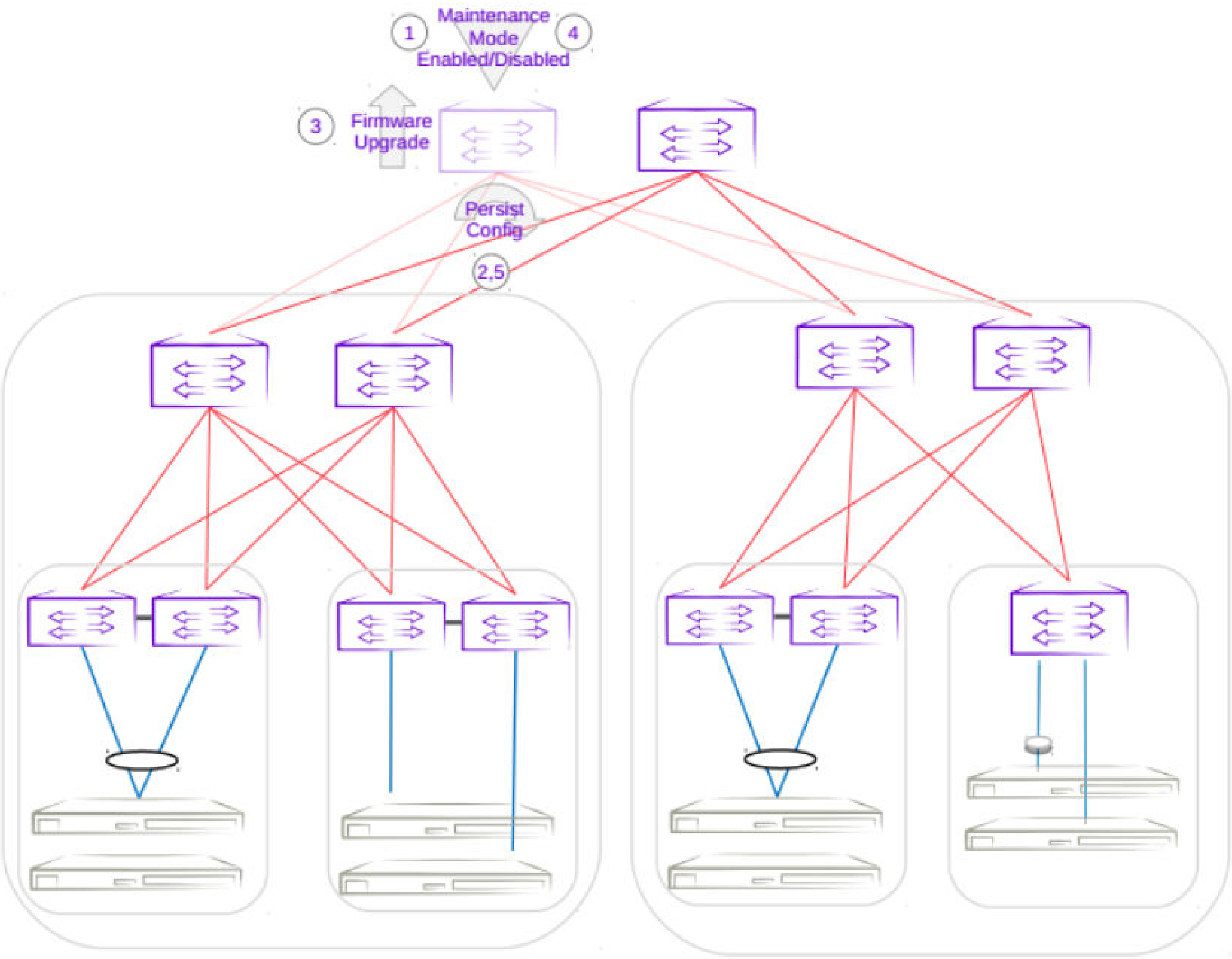


Figure 25: First super-spine firmware upgrade with maintenance mode

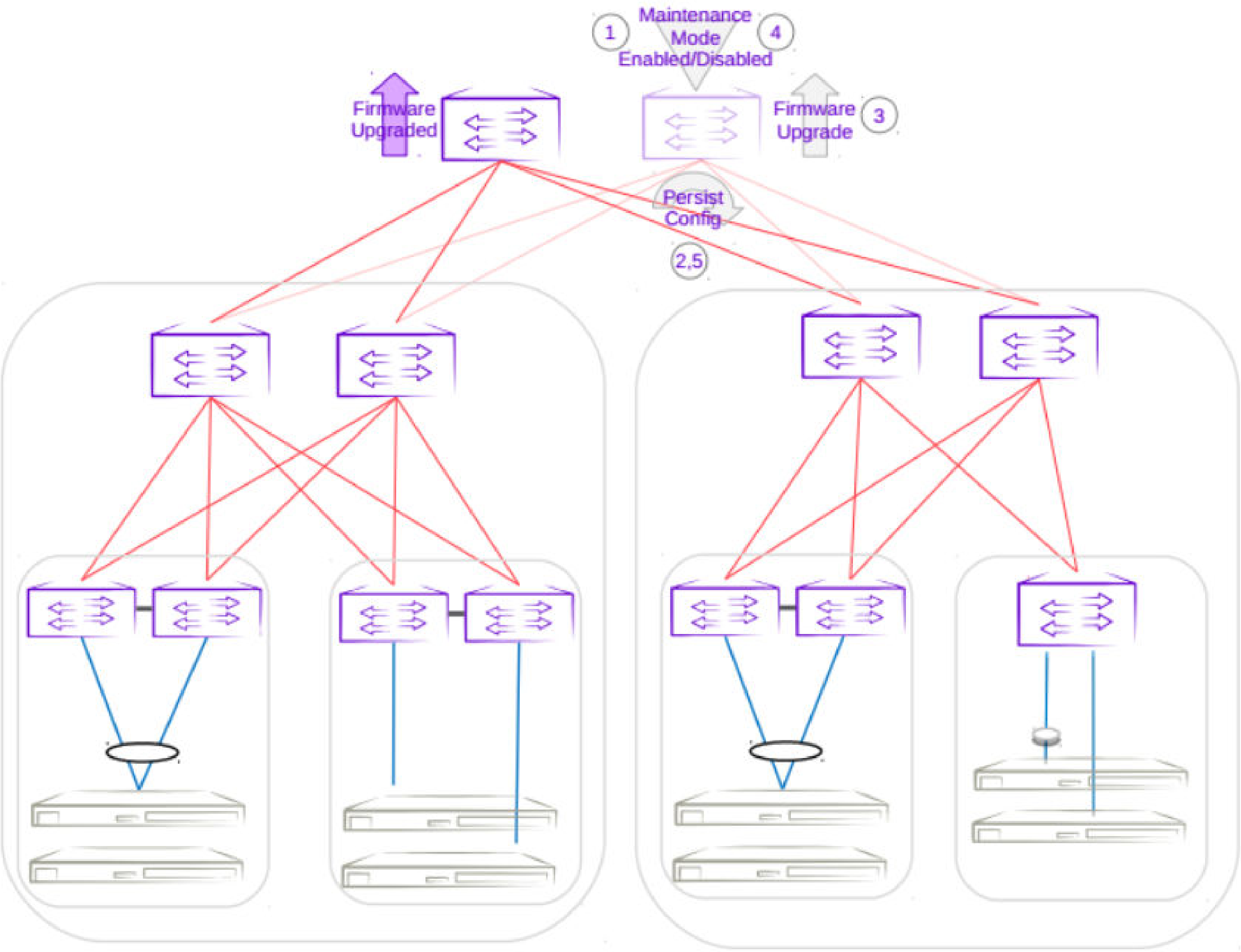


Figure 26: Second super-spine firmware upgrade with maintenance mode

Upgrade Spine in Clos

1. Enabling maintenance mode on a spine also involves BGP. The graceful_shutdown parameter will be sent out to all the spine's underlay neighbors (all leafs in the pod and super-spines). The neighbors will no longer send traffic to the first spine going into maintenance mode and redirect traffic to an alternate path.
2. The running-configuration is saved on the first spine to preserve all current configurations including the maintenance mode enable configuration.
3. The firmware on the first spine can now be upgraded and rebooted for firmware activation without traffic loss.
4. After the firmware is upgraded, the maintenance mode is disabled to allow traffic again through the upgraded spine.
5. The running-config is saved again to ensure the maintenance mode config remains disabled.

   The same process can be carried out on the second spine to upgrade the firmware without traffic loss.

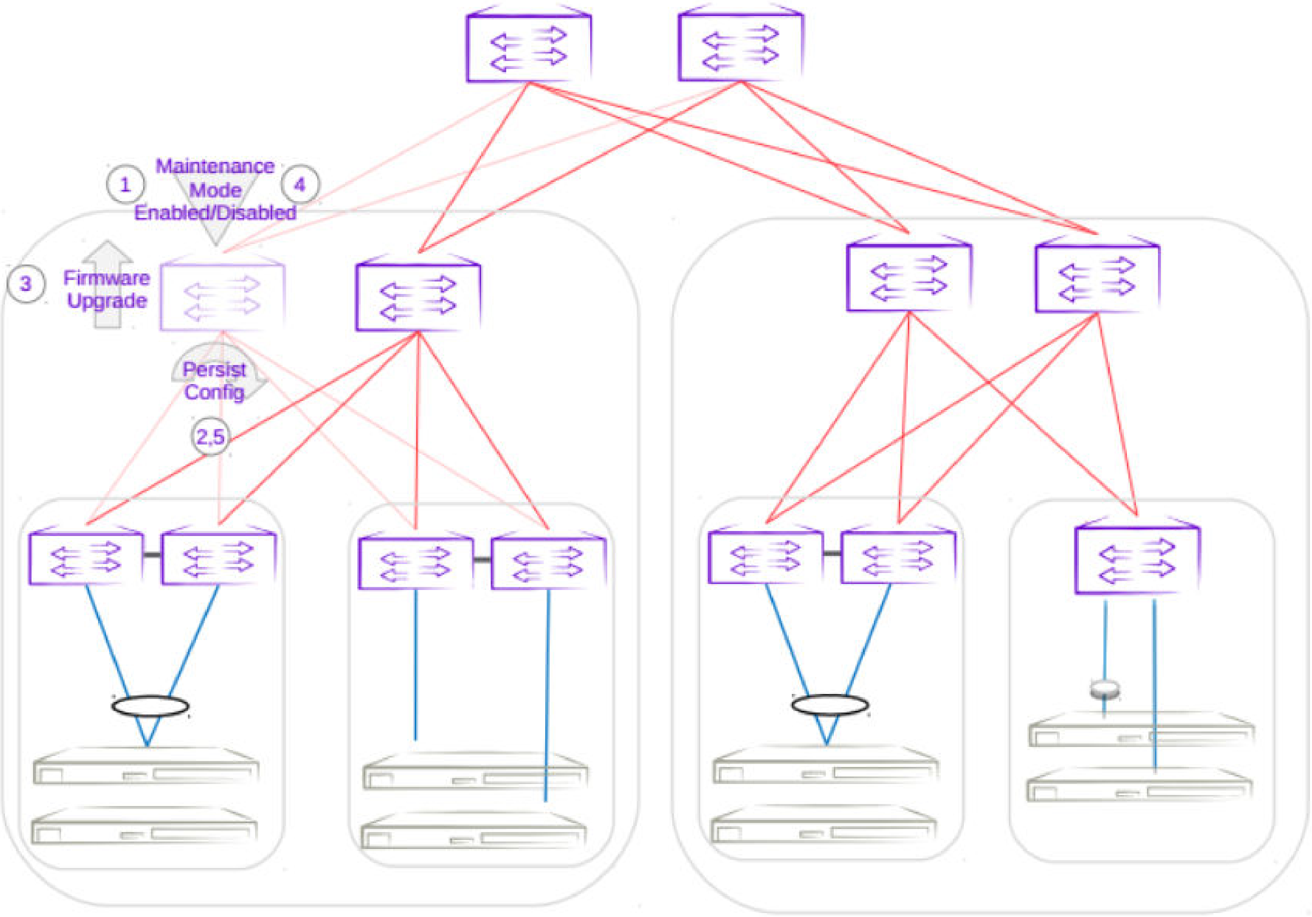


Figure 27: First spine firmware upgrade with maintenance mode

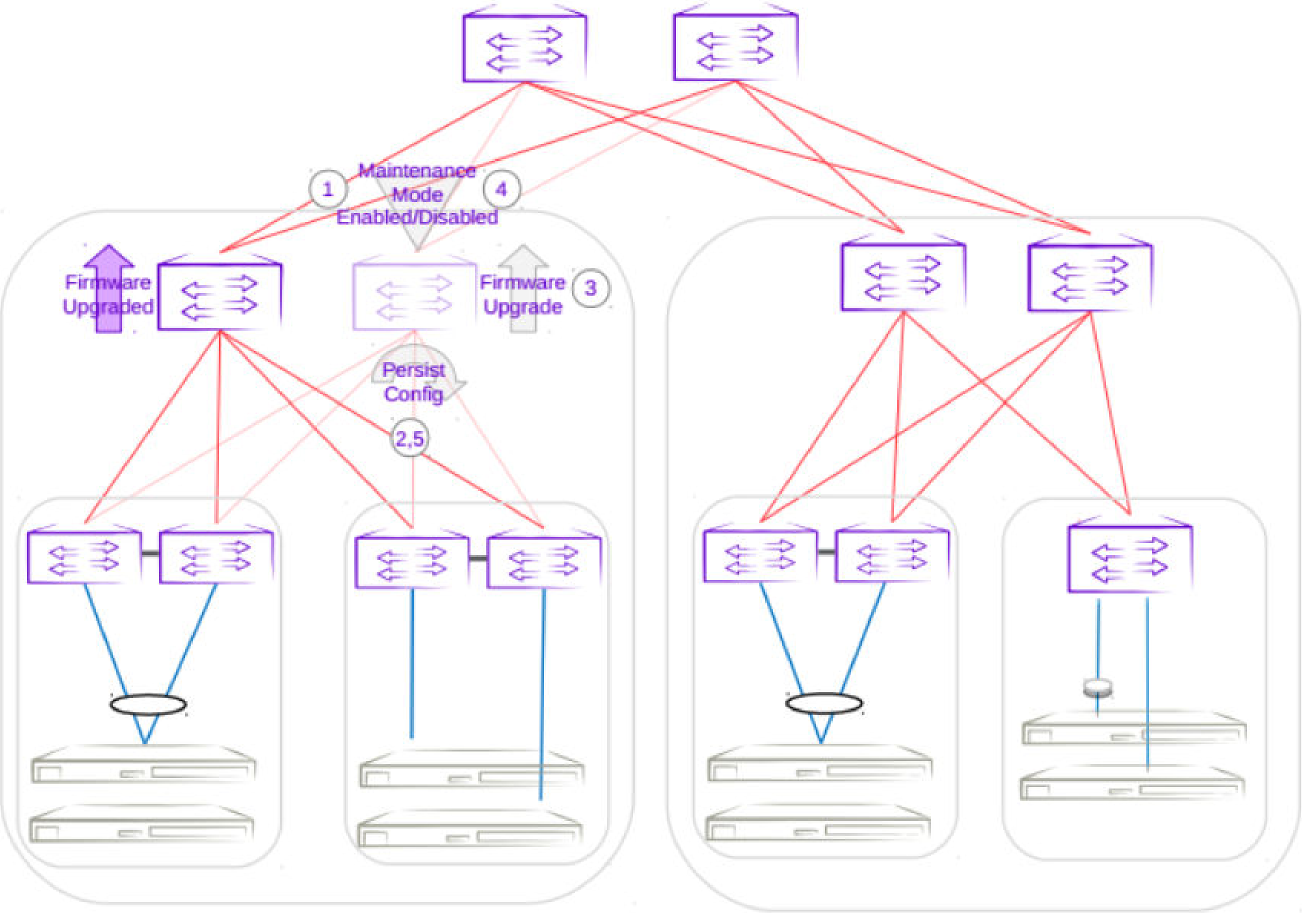


Figure 28: Second spine firmware upgrade with maintenance mode

Upgrade MCT Leaf Pair with Dual Homed Servers in Clos

1. Enabling maintenance mode on an MCT leaf involves BGP and MCT/NSM. The graceful_shutdown parameter will be sent out to all the leaf's underlay neighbors (all spines in the pod). The neighbors will no longer send traffic to the MCT leaf going into maintenance mode and redirect traffic from spines to the peer MCT leaf. MCT will instruct the peer leaf to become the designated forwarder, ICL will be shut down, and CCE ports for clients will also be shut down. Traffic from dual-homed servers will be redirected to the peer leaf. With maintenance mode enabled, traffic is completely redirected to the peer leaf.
2. The running-configuration is saved on the first MCT leaf to preserve all current configurations including the maintenance mode enable configuration.
3. The firmware on the MCT leaf can now be upgraded and rebooted for firmware activation without traffic loss.
4. After the firmware is upgraded, the maintenance mode is disabled to allow traffic again through the upgraded MCT leaf.
5. The running-config is saved again to ensure the maintenance mode config remains disabled.

   The same process can be carried out on the second MCT leaf to upgrade the firmware without traffic loss.

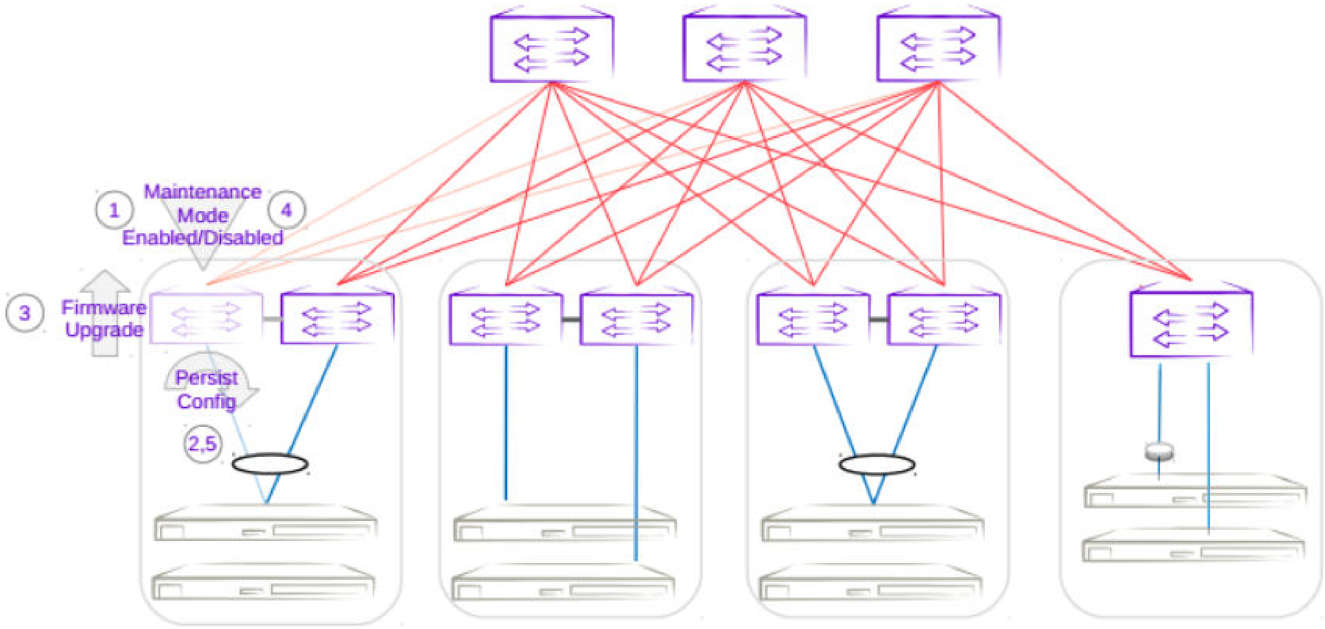


Figure 29: First MCT leaf firmware upgrade with maintenance mode

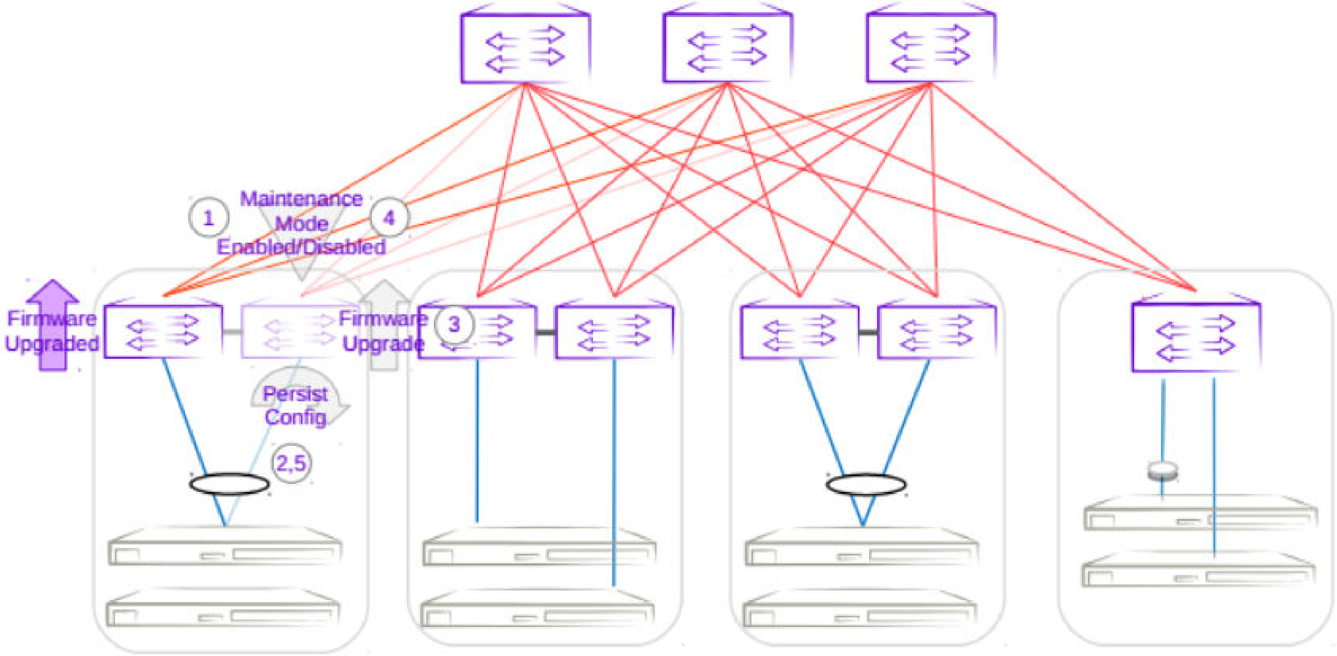


Figure 30: Second MCT leaf firmware upgrade with maintenance mode

Upgrade Three Rack Centralized in Non-Clos

1. Enabling maintenance mode on one of the leafs in the centralized MCT leaf pair follows the same behavior as the MCT leaf pair in Clos topology. The only difference is the iBGP L3 backup link between MCT leaf pairs. Maintenace mode will result in the traffic being redirected to the peer leaf in the centralized MCT leaf pairs.

2. The running-configuration is saved on the first MCT leaf to preserve all current configurations including the maintenance mode enable configuration.
3. The firmware on the MCT leaf can now be upgraded and rebooted for firmware activation without traffic loss.
4. After the firmware is upgraded, the maintenance mode is disabled to allow traffic again through the upgraded MCT leaf.
5. The running-config is saved again to ensure the maintenance mode config remains disabled.

   The same process can be carried out on the second MCT leaf to upgrade the firmware without traffic loss.

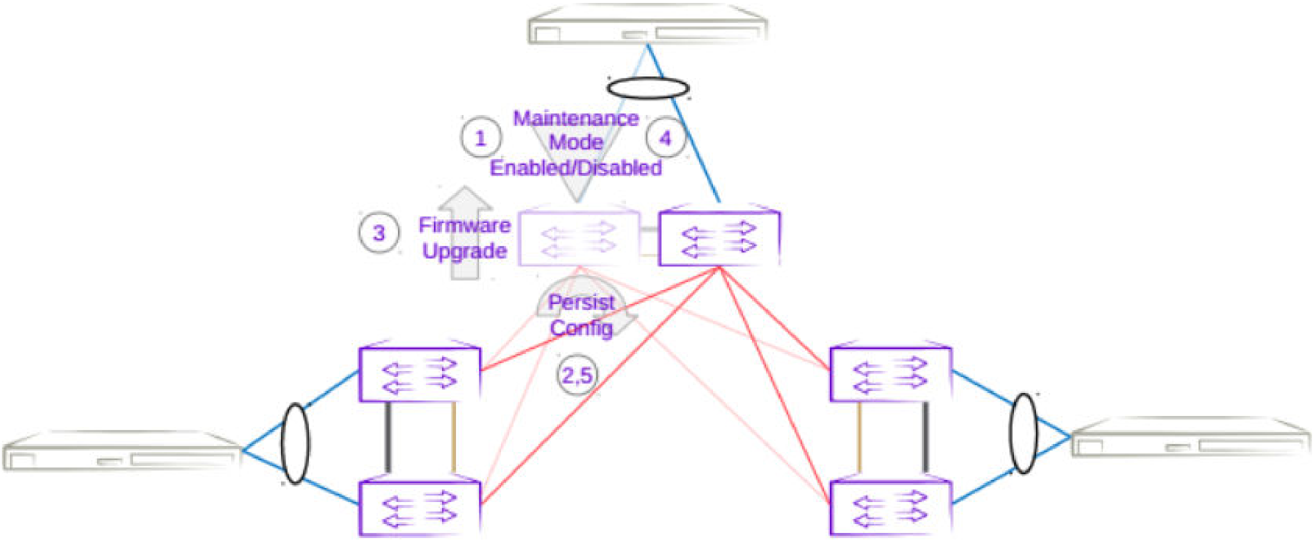


Figure 31: Three rack centralized first MCT leaf firmware upgrade with maintenance mode

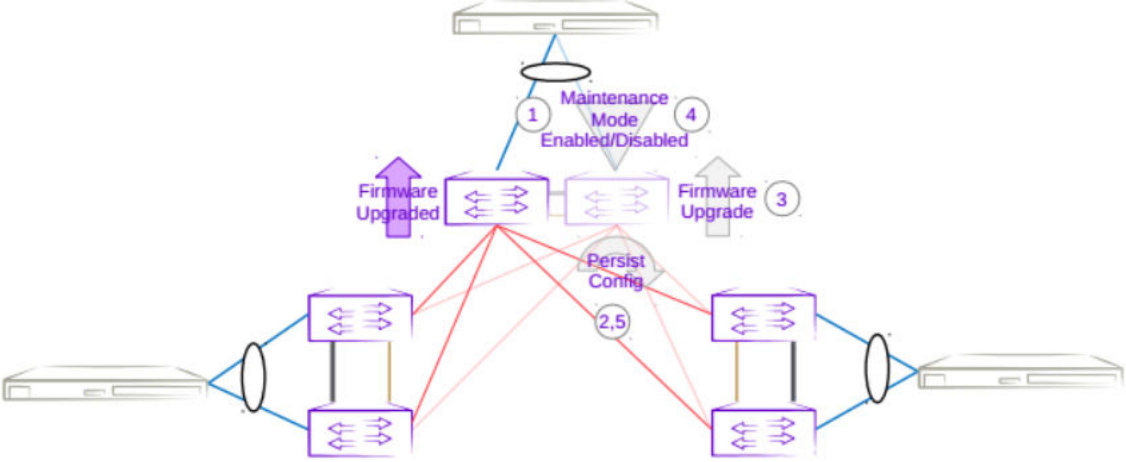


Figure 32: Three rack centralized Second MCT leaf firmware upgrade with maintenance mode

Upgrade Three Rack Ring in Non-Clos

1. Enabling maintenance mode on one of the leafs in a three rack ring MCT leaf pair follows the same behavior as the MCT leaf pair in Clos topology. The only difference is the iBGP L3 backup link between MCT leaf pairs. Maintenace mode will result in the traffic being redirected to the peer MCT leaf.
2. The running-configuration is saved on the first MCT leaf to preserve all current configurations including the maintenance mode enable configuration.
3. The firmware on the MCT leaf can now be upgraded and rebooted for firmware activation without traffic loss.
4. After the firmware is upgraded, the maintenance mode is disabled to allow traffic again through the upgraded MCT leaf.
5. The running-config is saved again to ensure the maintenance mode config remains disabled.

   The same process can be carried out on the second MCT leaf to upgrade the firmware without traffic loss.

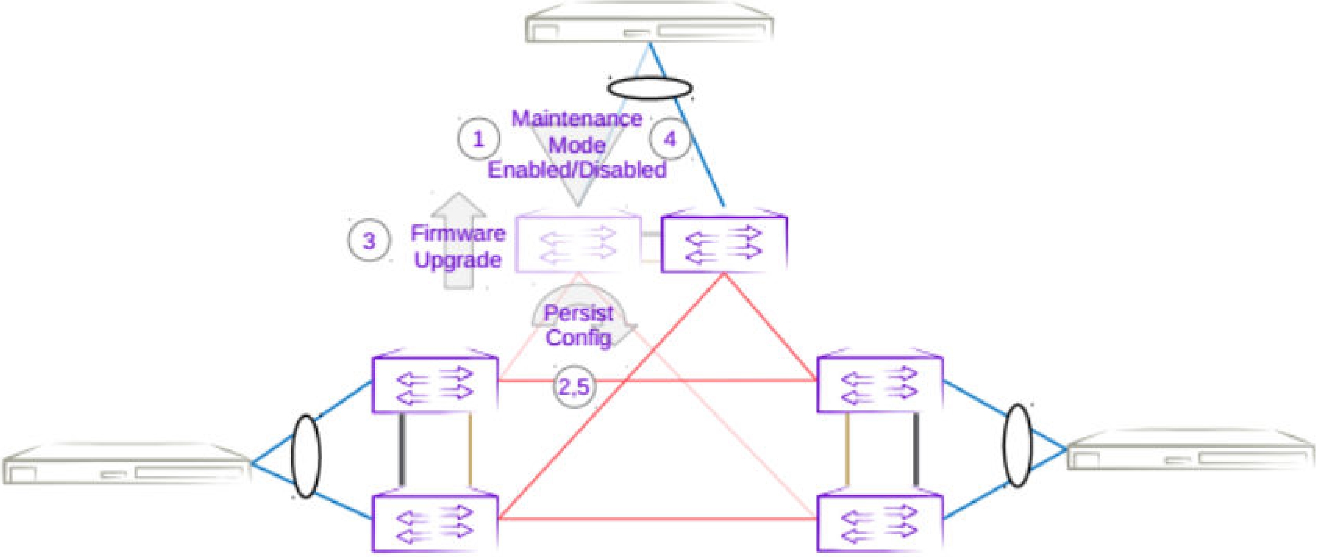


Figure 33: Three rack ring first MCT leaf firmware upgrade with maintenance mode

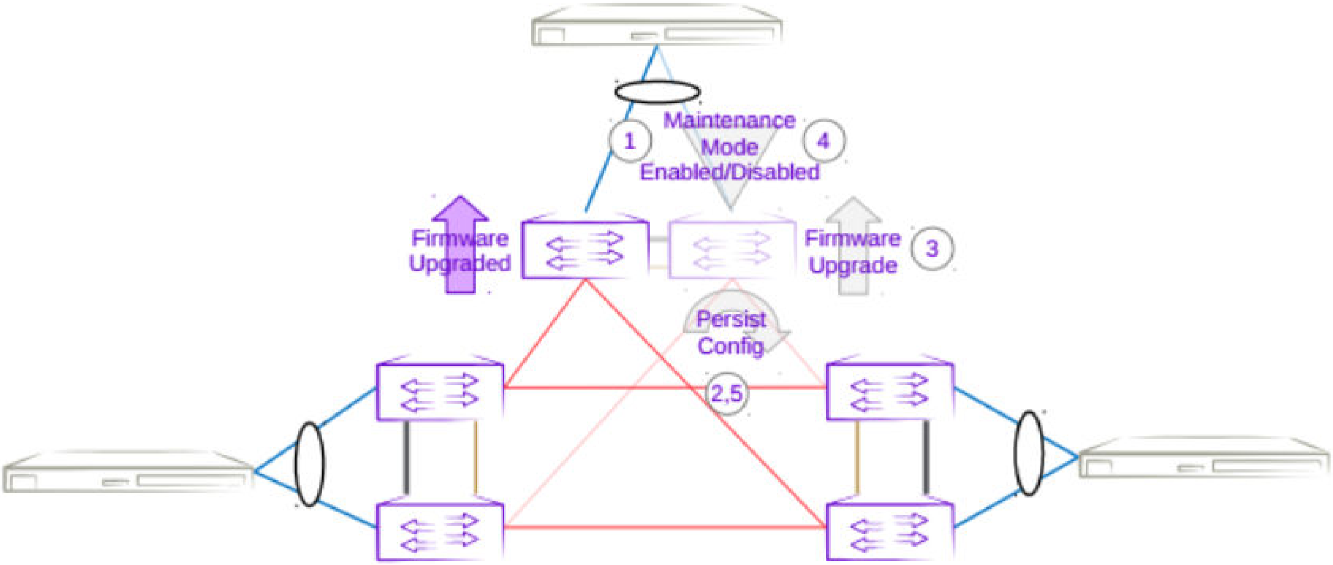


Figure 34: Three rack ring second MCT leaf firmware upgrade with maintenance mode

Traffic Loss

Single Leaf

Traffic loss is expected when upgrading a single leaf which is not in an MCT pair. Since there are no alternate paths for the single leaf, maintenance mode will not be enabled. Only the configuration will be persisted, and a firmware upgrade will be carried out. A traffic loss warning will be flagged when upgrading a single non-MCT leaf.

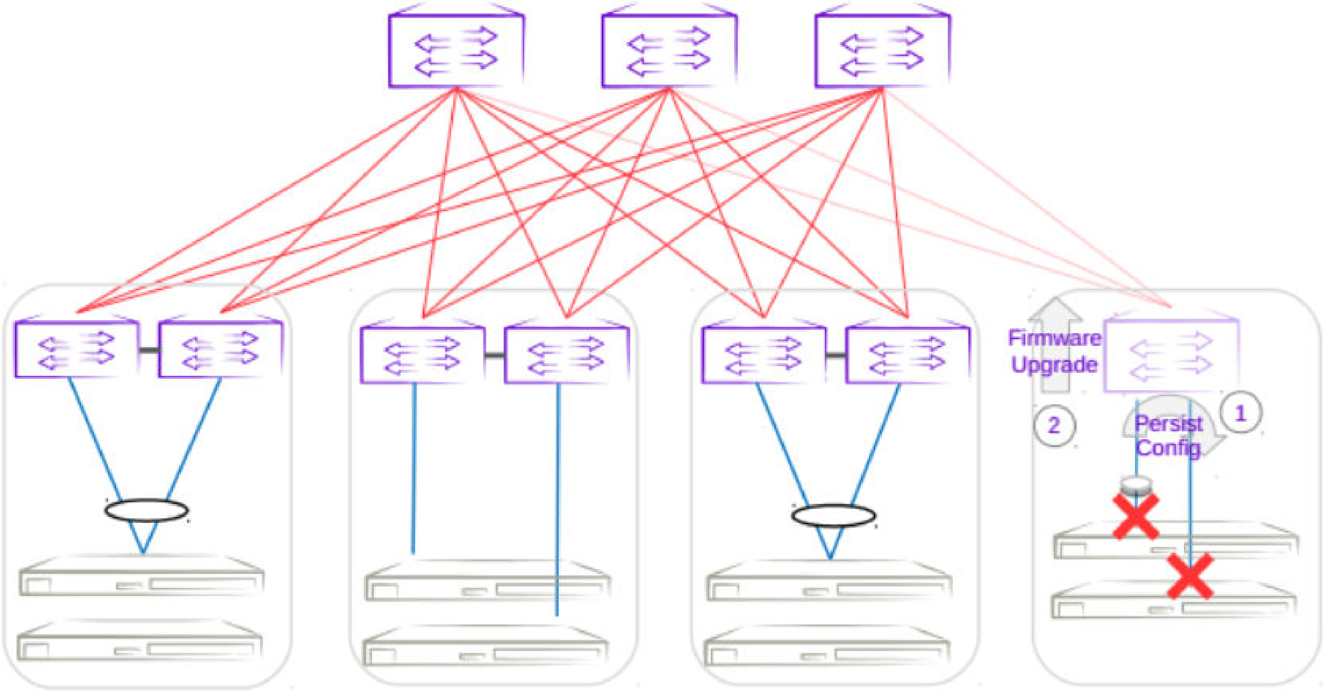


Figure 35: Single leaf traffic loss

Single-Homed Server

Traffic loss is also expected for any singled-homed server. Detecting single-homed servers are not in the scope of this feature so a generic warning will be provided at the start of a firmware download.

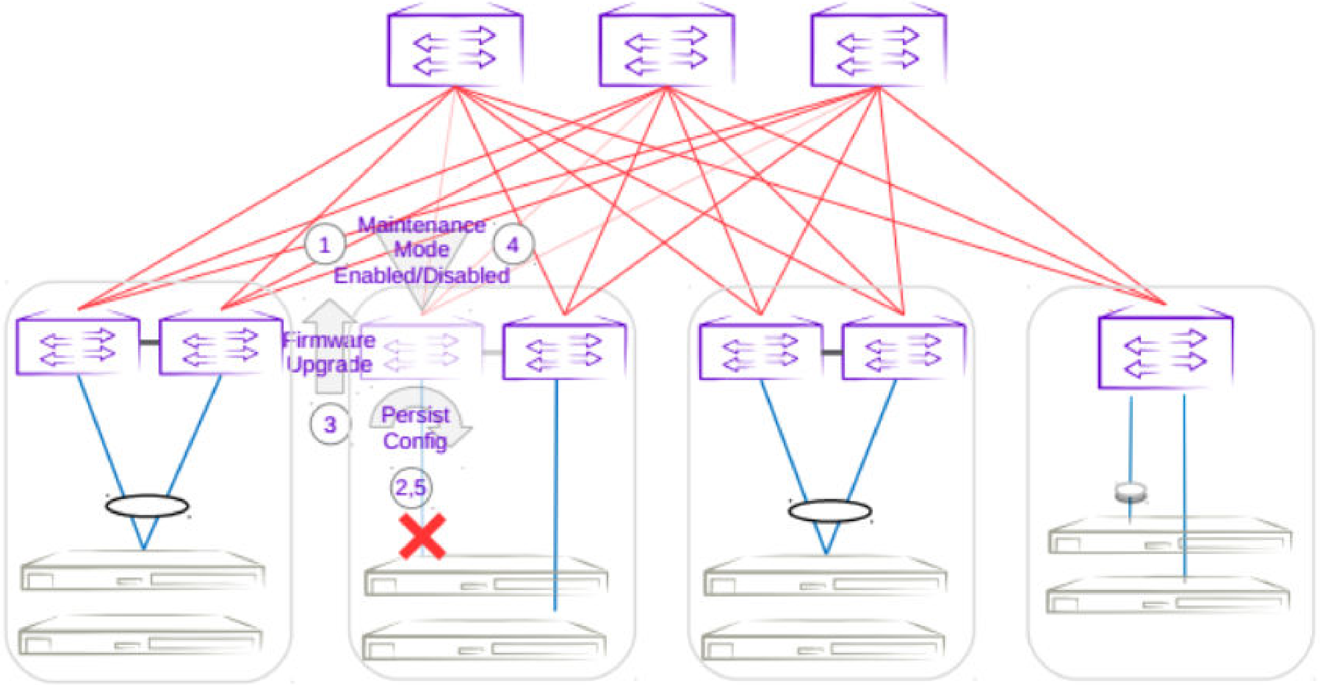


Figure 36: Single-homed server traffic loss

Non-Redundant Spine or Super-Spine

This is not a typical deployment, but traffic loss is expected in this scenario. Since no alternate paths exist for non-redundant switches, maintenance mode will not be enabled for this case. A traffic loss warning will be flagged when upgrading non-redundant switches.

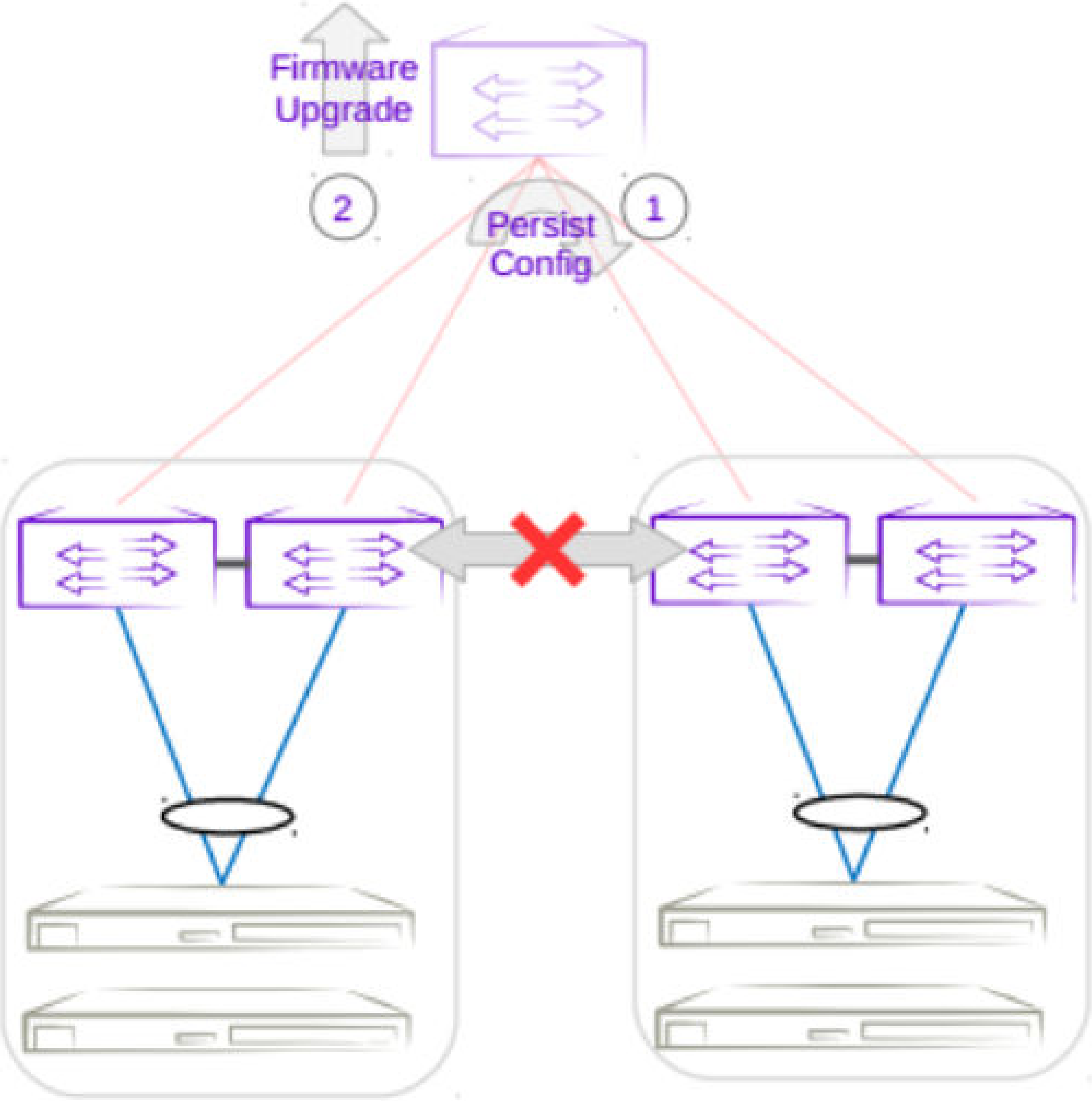


Figure 37: Non-redundant spine traffic loss

Firmware Download with Maintenance Mode Operations

The following operations are performed per device when a firmware download is executed.

1. Firmware Sanity Check
2. Maintenance Mode Enable
3. Persist Config
4. Firmware Download
5. Maintenance Mode Disable
6. Persist Config
7. Firmware Commit

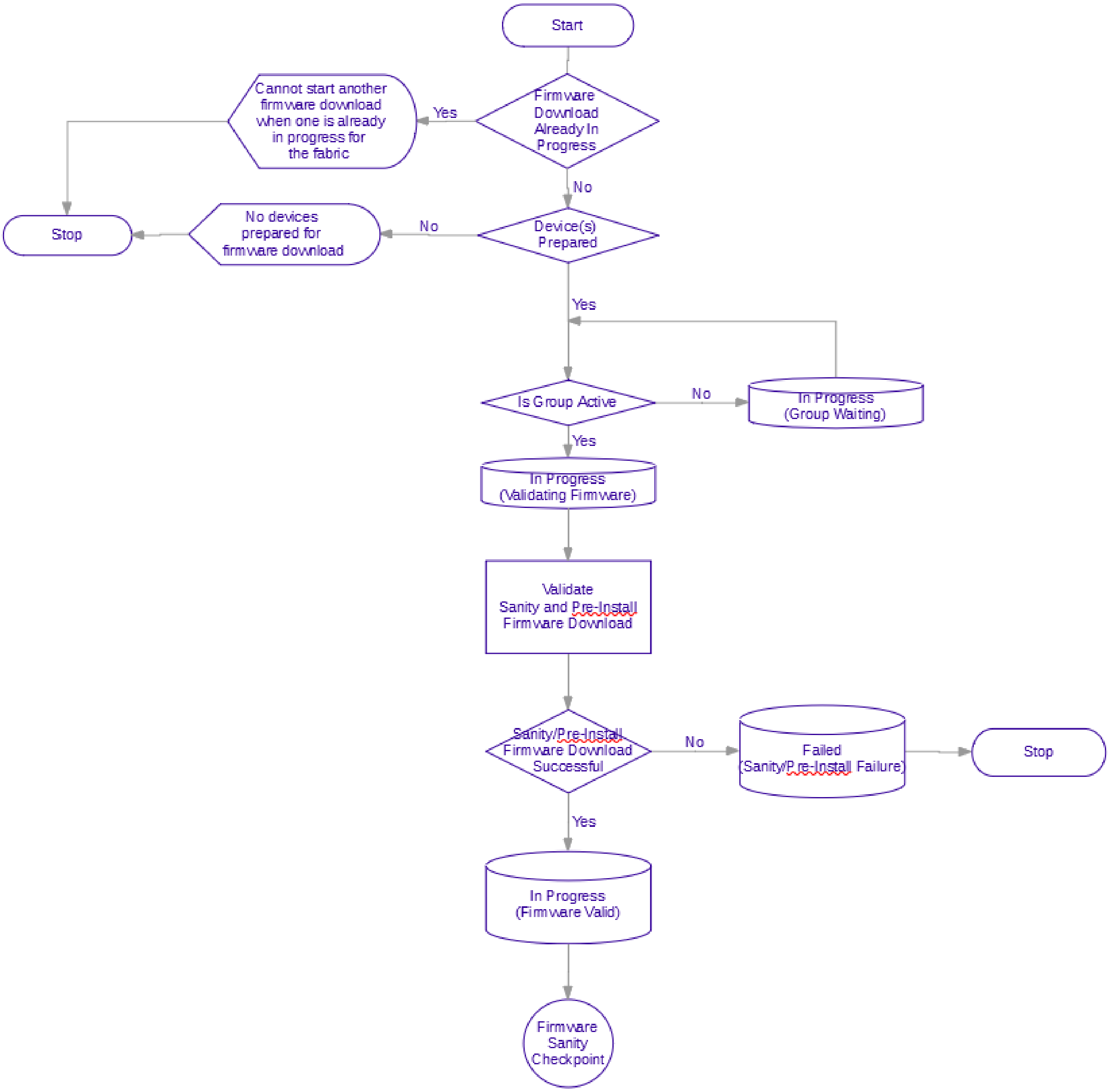

Start
Firmware Download Already In Progress
Yes
Cannot start another firmware download when one is already in progress for the fabric
No
Device(s) Prepared
No
No devices prepared for firmware download
Stop
Yes
Is Group Active
No
In Progress (Group Waiting)
Yes
In Progress (Validating Firmware)
Validate Sanity and Pre-Install Firmware Download
Sanity/Pre-Install Firmware Download Successful
No
Failed (Sanity/Pre-Install Failure)
Stop
Yes
In Progress (Firmware Valid)
Firmware Sanity Checkpoint

Figure 38: Firmware sanity check flowchart

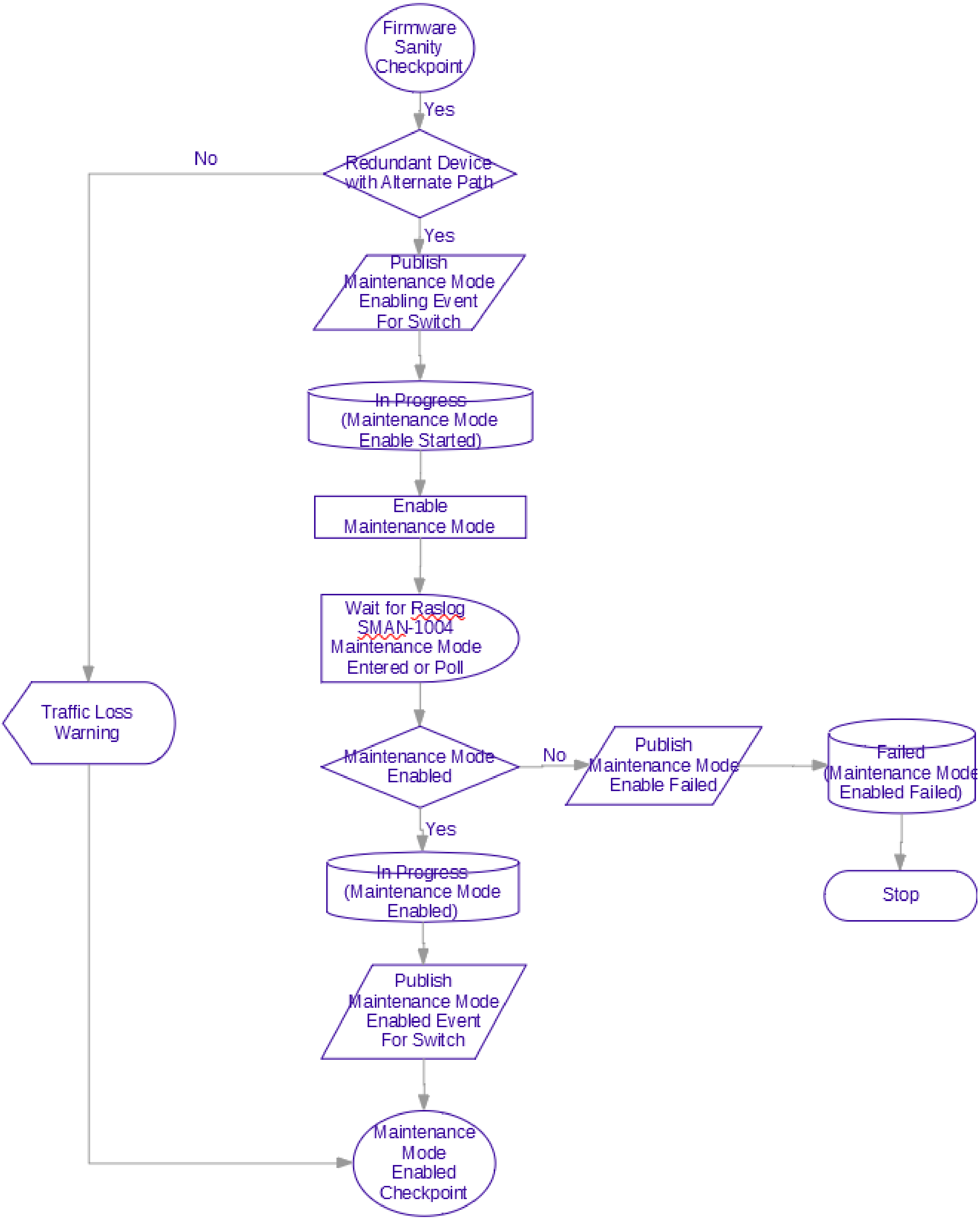

Firmware Sanity Checkpoint
Yes
Redundant Device with Alternate Path
No
Yes
Publish Maintenance Mode Enabling Event For Switch
In Progress (Maintenance Mode Enable Started)
Enable Maintenance Mode
Wait for Raslog SMAN-1004 Maintenance Mode Entered or Poll
Traffic Loss Warning
Maintenance Mode Enabled
No
Publish Maintenance Mode Enable Failed
Failed (Maintenance Mode Enabled Failed)
Stop
Yes
In Progress (Maintenance Mode Enabled)
Publish Maintenance Mode Enabled Event For Switch
Maintenance Mode Enabled Checkpoint

Figure 39: Maintenance mode enable flowchart

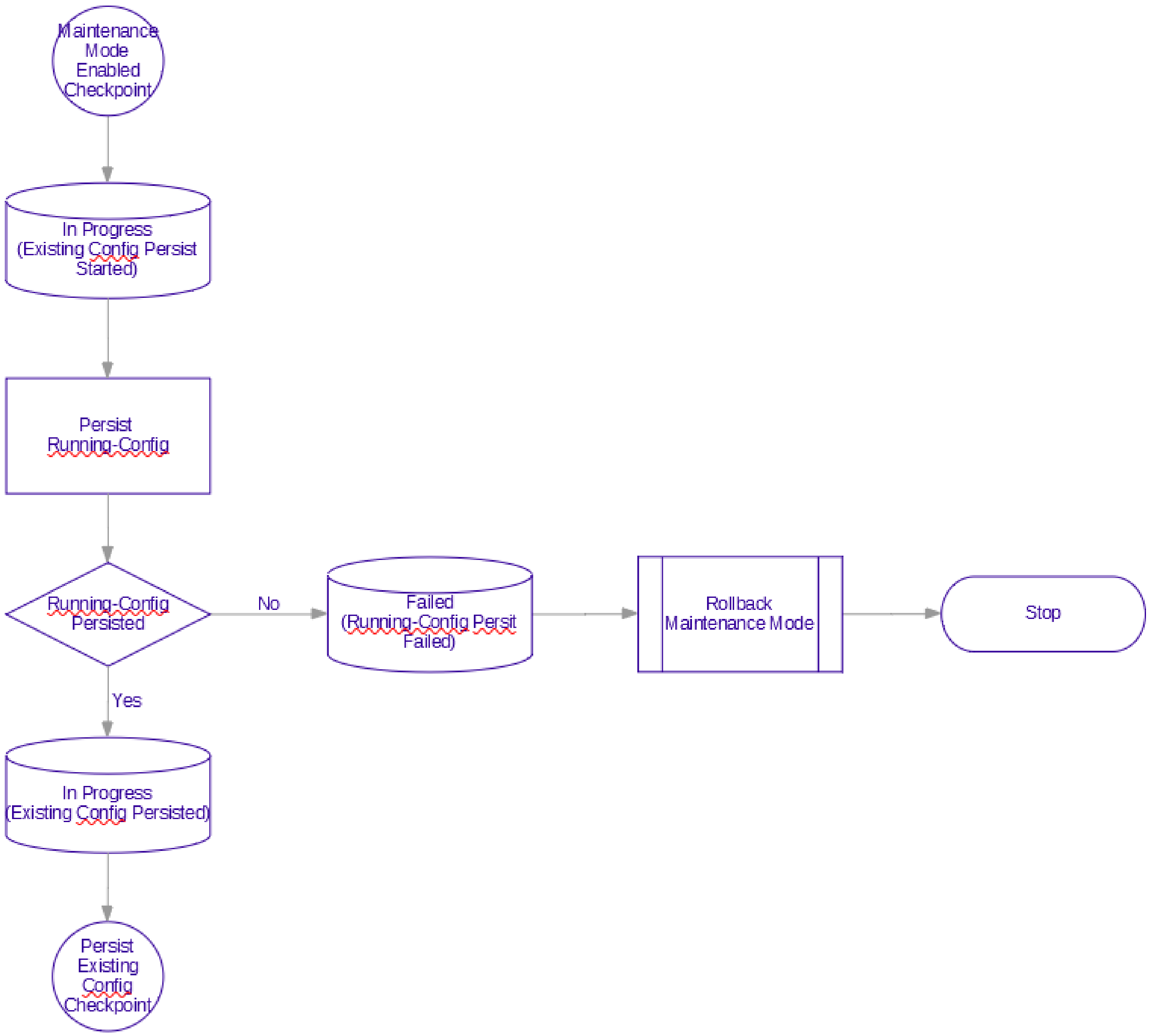

Maintenance Mode Enabled Checkpoint
In Progress (Existing Config Persist Started)
Persist Running-Config
Running-Config Persisted
No
Failed (Running-Config Persit Failed)
Rollback Maintenance Mode
Stop
Yes
In Progress (Existing Config Persisted)
Persist Existing Config Checkpoint

Figure 40: Persist existing config flowchart

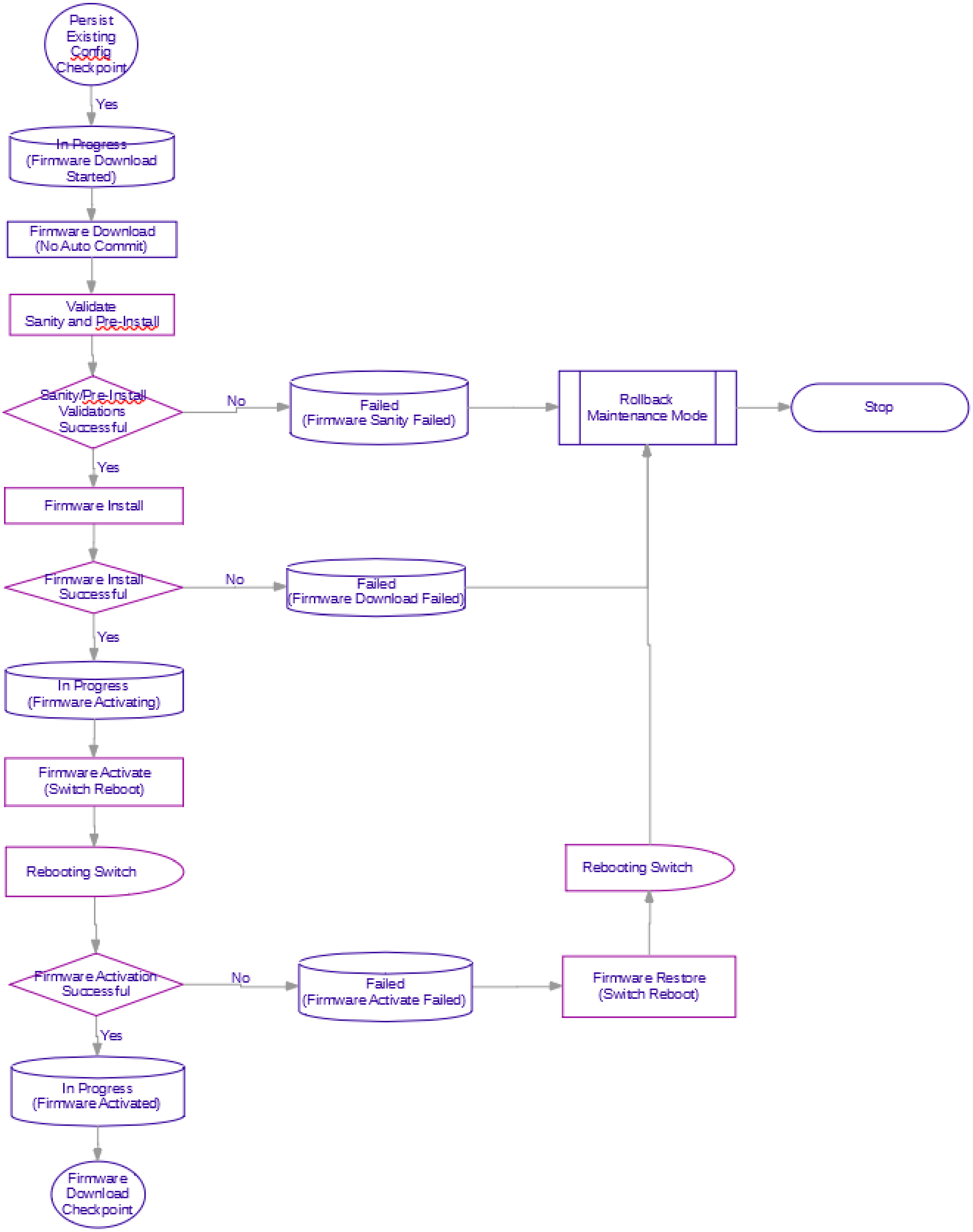

Persist Existing Config Checkpoint
Yes
In Progress (Firmware Download Started)
Firmware Download (No Auto Commit)
Validate Sanity and Pre-Install
Sanity/Pre-Install Validations Successful
No
Failed (Firmware Sanity Failed)
Rollback Maintenance Mode
Stop
Yes
Firmware Install
Firmware Install Successful
No
Failed (Firmware Download Failed)
Yes
In Progress (Firmware Activating)
Firmware Activate (Switch Reboot)
Rebooting Switch
Rebooting Switch
Firmware Activation Successful
No
Failed (Firmware Activate Failed)
Firmware Restore (Switch Reboot)
Yes
In Progress (Firmware Activated)
Firmware Download Checkpoint

Figure 41: Firmware download flowchart

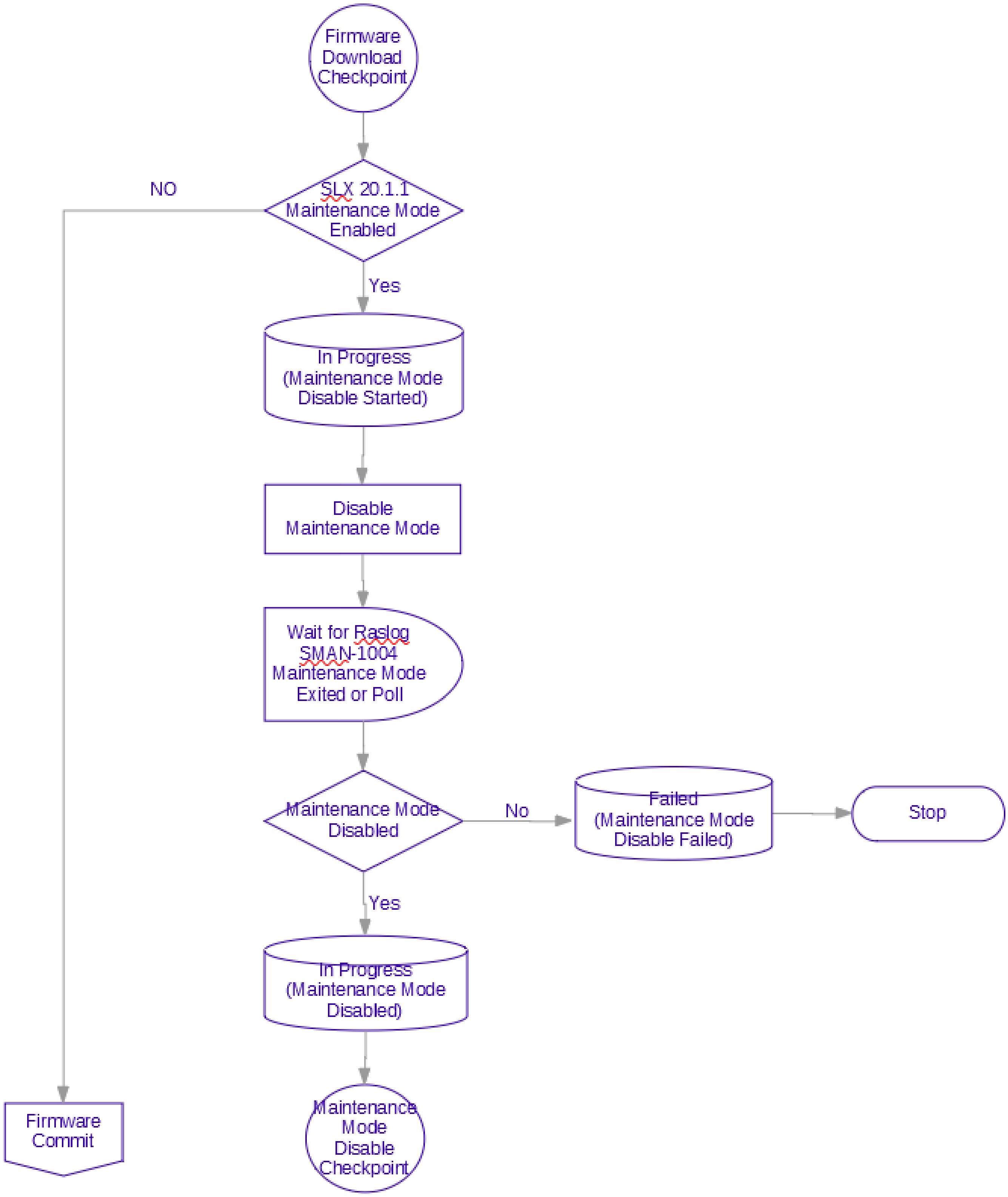

Firmware Download Checkpoint
SLX 20.1.1 Maintenance Mode Enabled
NO
Yes
In Progress (Maintenance Mode Disable Started)
Disable Maintenance Mode
Wait for Raslog SMAN-1004 Maintenance Mode Exited or Poll
Maintenance Mode Disabled
No
Failed (Maintenance Mode Disable Failed)
Stop
Yes
In Progress (Maintenance Mode Disabled)
Firmware Commit
Maintenance Mode Disable Checkpoint

Figure 42: Maintenance mode disable flowchart

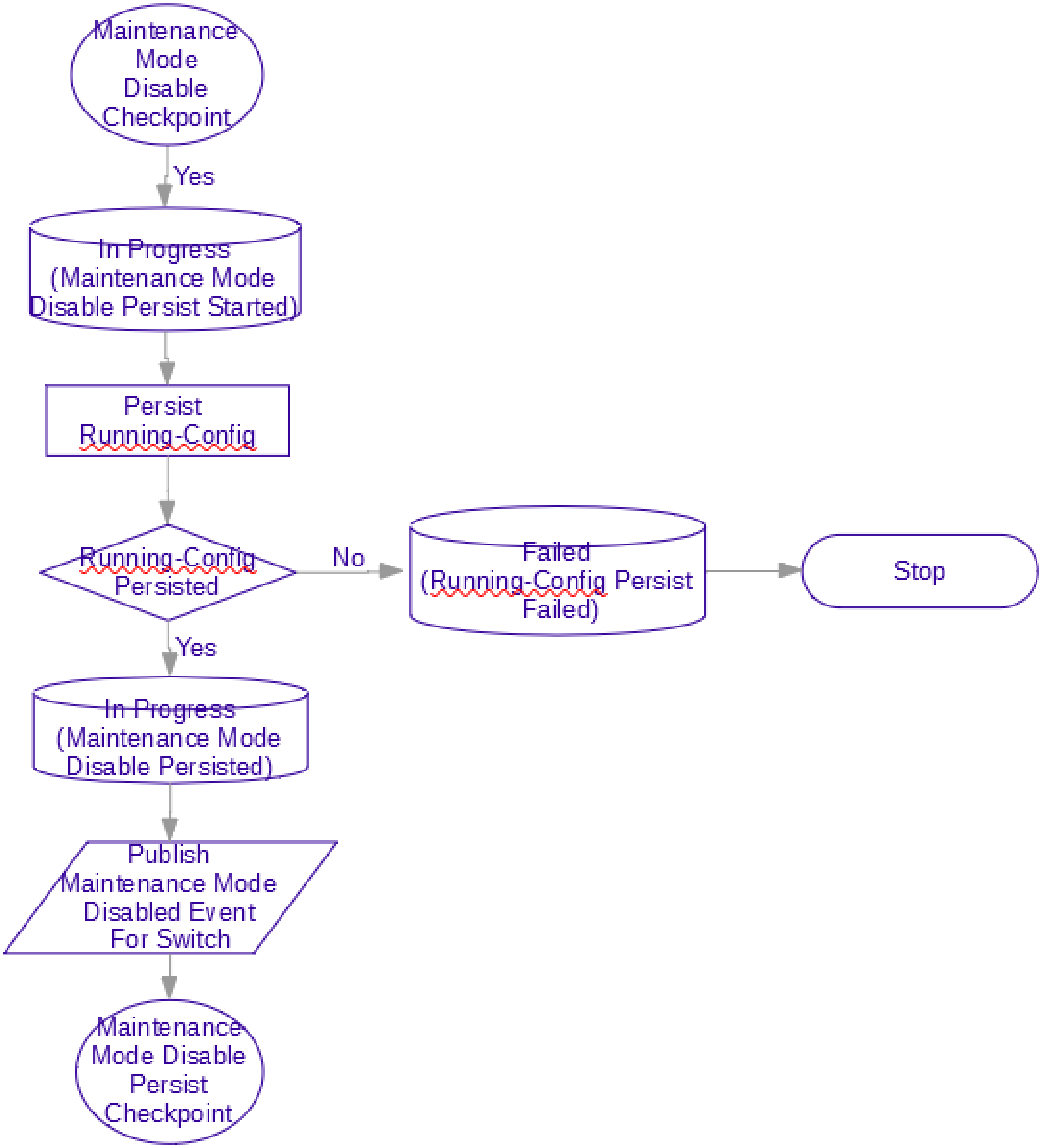

Maintenance Mode Disable Checkpoint
Yes
In Progress (Maintenance Mode Disable Persist Started)
Persist Running-Config
Running-Config Persisted
No
Failed (Running-Config Persist Failed)
Stop
Yes
In Progress (Maintenance Mode Disable Persisted)
Publish Maintenance Mode Disabled Event For Switch
Maintenance Mode Disable Persist Checkpoint

Figure 43: Persist maintenance mode disable config flowchart

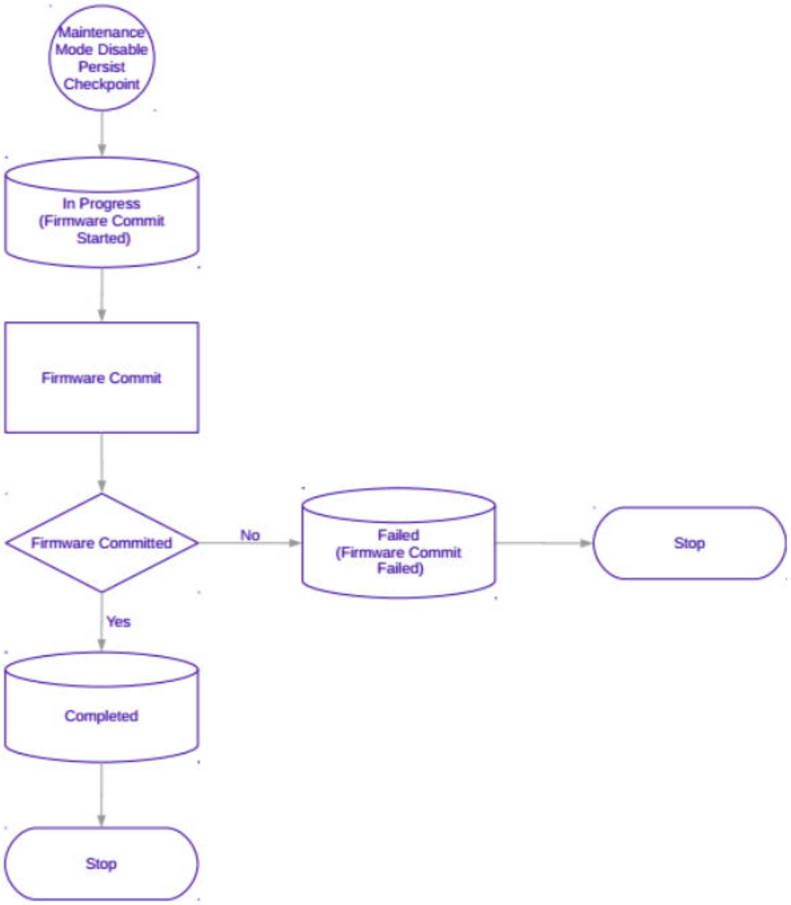

Maintenance Mode Disable Persist Checkpoint
In Progress (Firmware Commit Started)
Firmware Commit
Firmware Committed
No
Failed (Firmware Commit Failed)
Stop
Yes
Completed
Stop

Figure 44: Firmware commit flowchart

Foundation and Ecosystem Services Interaction

When maintenance mode is enabled, configuration changes to the switch are not allowed. This feature will notify other services by publishing the following messages.

Message	Details	Maintenance Mode State
Maintenance Mode Fabric Enabling	Published just prior to issuing the maintenance mode on one or more devices in the fabric when no devices had maintenance mode enabled previously.	At least one device in the fabric is enabling or has maintenance mode enabled.
Maintenance Mode Fabric Disabled	Published when maintenance mode disable (no enable) operation has completed on devices in the fabric and all devices in the fabric have maintenance mode disabled.	All devices in the fabric have maintenance mode disabled.
Maintenance Mode Device Enabling	Published just prior to issuing the maintenance mode enable configuration to the switch.	Disabled

Message	Details	Maintenance Mode State
Maintenance Mode Device Enable Failed	Published when maintenance mode enable operation on the switch has failed or timed out.	Disabled
Maintenance Mode Device Enable Success	Published when maintenance mode enable operation on the switch is successful.	Enabled
Maintenance Mode Device Disabling	Published just prior to issuing the maintenance mode disable (no enable) configuration to the switch.	Enabled
Maintenance Mode Device Disable Failed	Published when maintenance mode disable (no enable) operation on the switch has failed or timed out.	Enabled
Maintenance Mode Device Disable Success	Published when maintenance mode disable (no enable) operation on the switch is successful.	Disabled

Persisting SLX Configurations

SLX 20.1.1 Behavior

In SLX 20.1.1, configuration management process maintains two databases.

- Running-DB maintains the running configuration.
- Startup-DB maintains the persisted configuration which is created or updated when the **copy running-config startup-config** command is executed.

SLX OS maintains a startup file which is an ASCII file saved in flash when running-config is copied to startup-config. The startup file can also be updated by any ASCII config from outside the switch. By default, when system reboots, the SLX OS replays the configuration from the Startup database. The user can modify the Startup file to replay the configuration if required. It is also done in some exception cases like firmware upgrade or DB corruption.

Maintenance mode can be enabled by configuring **enable** under system-maintenance configuration mode. If the configuration is persisted, the switch needs to be in maintenance mode before reloading for it to come back in maintenance mode.

SLX 20.1.2 Enhancement

In SLX 20.1.2, all the configurations are stored in one database which also persists.

- The **show running-config** command always fetches the configuration from the database.
- The **copy running-config startup-config** command keeps a copy of the startup-file in flash.
- Checkpoint creations for rollback operations depend on the database.
- After a upgrade or downgrade, replaying the startup-file resumes the database cleanup operations.

Maintenance mode can be enabled by configuring **enable-on-reboot** under system-maintenance configuration mode. After the reboot, the switch comes back up in maintenance mode and remains operational.

```
SLX(config-system-maintenance)# enable-on-reboot
SLX(config-system-maintenance)# [no] enable-on-reboot
```

The **system maintenance turn-off** command brings the system out of maintenance mode.

Switch Health Management

Switch Health Management (SHM) performs drift and reconciliation services, restoring fabric related configurations.

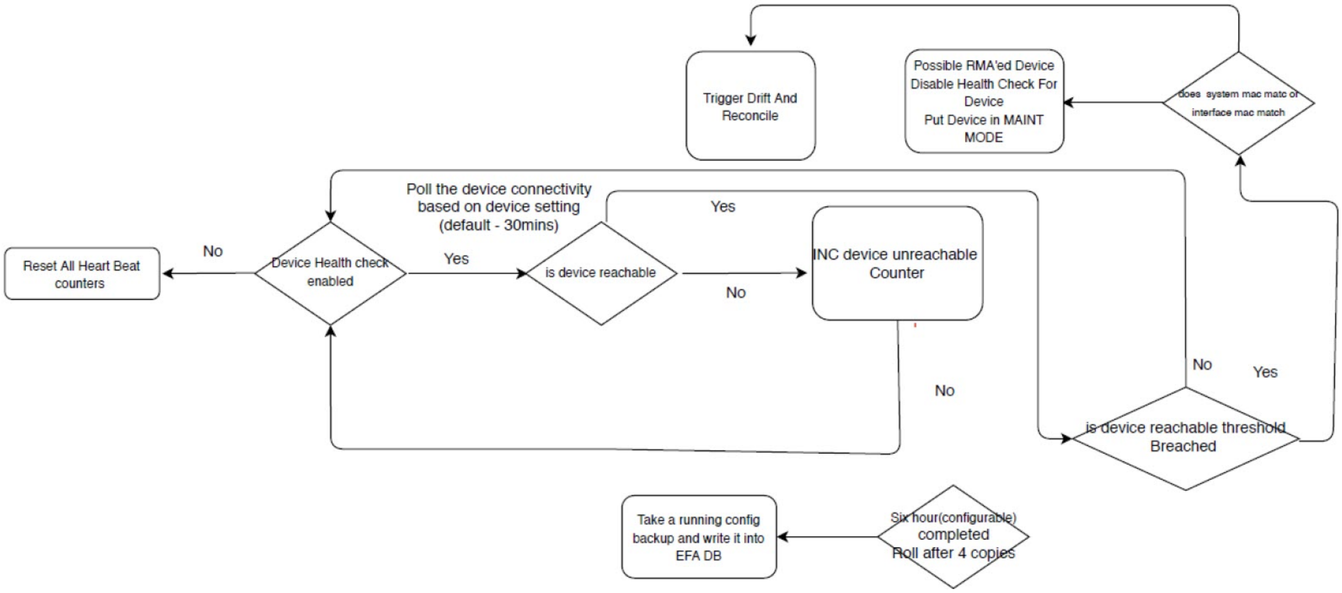

Trigger Drift And Reconcile
Possible RMA'ed Device Disable Health Check For Device Put Device in MAINT MODE
does system mac matc or interface mac match
Poll the device connectivity based on device setting (default - 30mins)
Yes
Reset All Heart Beat counters
No
Device Health check enabled
Yes
is device reachable
No
INC device unreachable Counter
No
No
Yes
is device reachable threshold Breached
Take a running config backup and write it into EFA DB
Six hour(configurable) completed Roll after 4 copies

Figure 45: Switch Health Management work flow

Monitor Switch Health

The switches registered with EFA can be monitored for connectivity issues. If connectivity violates pre-defined thresholds, EFA starts drift and reconciliation.

1. Enable switch health check.

   ```
   # efa inventory device setting update --ip 10.24.14.133 --health-check-enable yes
   ```

2. Configure health check interval.

   ```
   # efa inventory device setting update --ip 10.24.14.133 --health-check-interval 30mins
   ```

3. Configure health check threshold.

   ```
   # efa inventory device setting update --ip 10.24.14.133 --health-check-heartbeat-miss-
   threshold 2
   ```

4. View device health status.

   ```
   # efa inventory device health status --ip 10.24.14.133
   ```

5. (Optional) Disable switch health check.

   ```
   # efa inventory device setting update --ip 10.24.14.133 --health-check-enable no
   ```

Switch Configuration Backup and Replay

Switch Configuration backup and replay enables backup of the switch configuration based on inventory device setting or user executed REST/CLI.

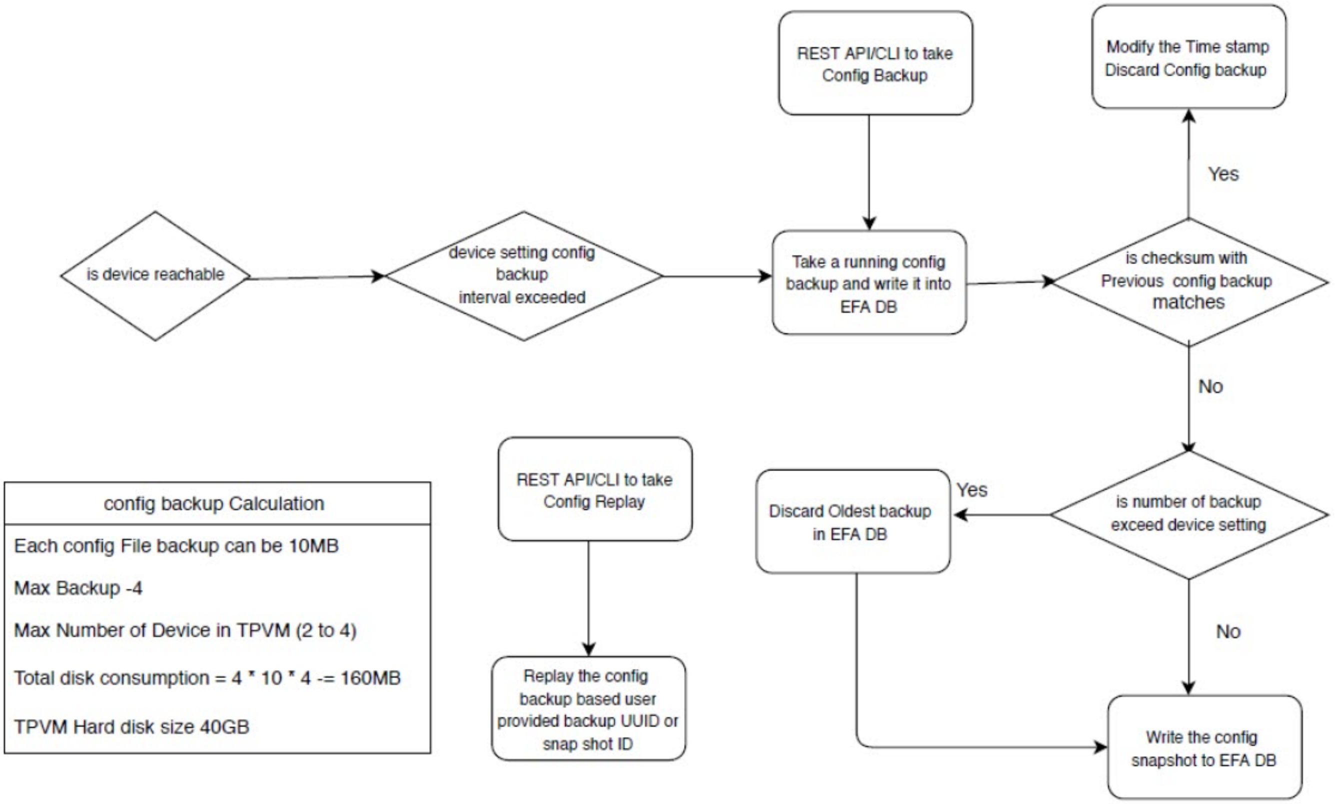

REST API/CLI to take Config Backup
Modify the Time stamp Discard Config backup
Yes
is device reachable
device setting config backup interval exceeded
Take a running config backup and write it into EFA DB
is checksum with Previous config backup matches
No
config backup Calculation
Each config File backup can be 10MB
Max Backup -4
Max Number of Device in TPVM (2 to 4)
Total disk consumption = 4 * 10 * 4 -= 160MB
TPVM Hard disk size 40GB
REST API/CLI to take Config Replay
Replay the config backup based user provided backup UUID or snap shot ID
Discard Oldest backup in EFA DB
Yes
is number of backup exceed device setting
No
Write the config snapshot to EFA DB

Figure 46: Switch Config Backup and Config Replay workflow

Configure Backup and Replay

1. Enable periodic config-backup.

```
# efa inventory device setting update --ip 10.24.14.133 --config-backup-periodic-
enable yes
```

2. Configure device backup.

```
efa inventory device setting update --ip 10.24.14.133 --config-backup-interval 30m
[3m-1800m, default 1440m]
# efa inventory device setting update --ip 10.24.14.133 --number-of-config-backups 2
[2-20, default 4]
# efa inventory config-backup execute --ip 10.24.14.133
```

3. View config-backup history.

```
# efa inventory config-backup history --ip 10.24.14.133
# efa inventory config-backup detail --uuid 1111-1111-1111 --show-config
# efa inventory config-backup detail --uuid 1111-1111-1111 --show-config --file-dump
<filename>
```

4. Delete config-backup.

```
# efa inventory config-backup delete --key 10.24.14.133
# efa inventory config-backup delete --key 1111-1111-111
```

5. Configure device replay.

```
# efa inventory config-replay execute --ip 10.24.14.133 --uuid 1111-1111-111
```

6. View config-replay history.

```
# efa inventory config-replay history --ip 10.24.14.133
# efa inventory config-replay detail --uuid 1111-1111-1111
```

7. (Optional) Delete config-replay.

```
# efa inventory config-replay delete --key 10.24.14.133
# efa inventory config-replay delete --key 1111-1111-111
```

Drift Identification and Reconcile

EFA provides APIs to initiate Drift and Reconcile requests. Drift Identification and Reconcile support is provided at device level. The unit of comparison is a single device whose configuration is compared with EFA and reconciled in case of a drift.

Drift Identification and reconcile is used during the following operations:

- Switch replacement
- After the reboot of a device in maintenance mode

Drift and Reconciliation Engine

The APIs for Drift and Reconcile perform the following operations:

Note
If **`maintenance-mode-enable on reboot`** is not set on the devices, Data Consistency is not guaranteed and Drift And Reconciliation operation is skipped.

1. Raslog received from the switch starts the state engine for reconciliation of the device.
   a. Initiate reconciliation of the Fabric Service
   b. Initiate reconciliation of the Tenant Service
2. Identify the drift in configuration by comparing the fabric configurations in Fabric Service with configurations in Asset service. Fabric service performs reconciliation and pushes the intended configuration from fabric to the device.
3. Identify the drift in configuration by comparing the Tenant configurations in Tenant Service with configurations in Asset service. Tenant service performs reconciliation and pushes the intended configuration from fabric to the device.

The reconcileAPI does not perform reconciliation on the device. The reconcileAPI only identifies the configuration drift and displays the information. This API can also initiate device discovery before starting the reconcile engine.

To improve performance, the drift computation is done in multiple go-routines and bulk switch configurations per device as applicable.

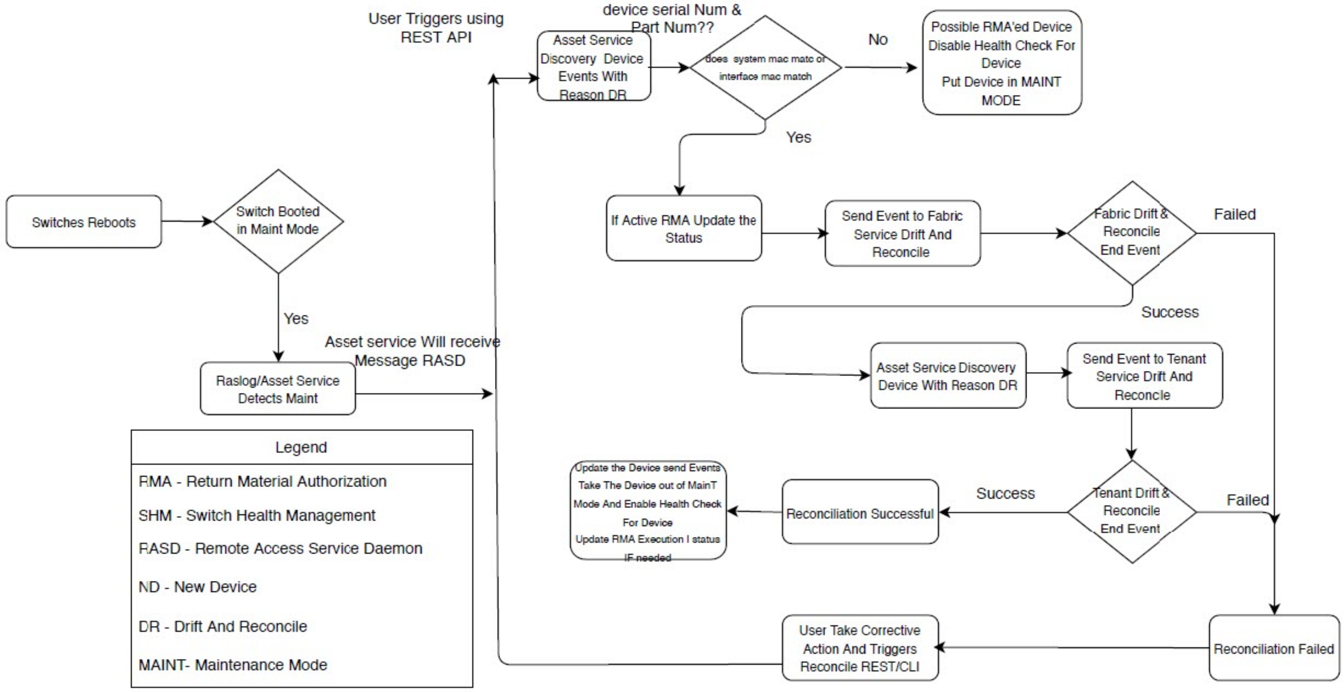

User Triggers using REST API
device serial Num & Part Num??
Asset Service Discovery Device Events With Reason DR
does system mac matc or interface mac match
No
Possible RMA'ed Device Disable Health Check For Device Put Device in MAINT MODE
Yes
Switches Reboots
Switch Booted in Maint Mode
If Active RMA Update the Status
Send Event to Fabric Service Drift And Reconcile
Fabric Drift & Reconcile End Event
Failed
Success
Yes
Asset service Will receive Message RASD
Raslog/Asset Service Detects Maint
Asset Service Discovery Device With Reason DR
Send Event to Tenant Service Drift And Reconcle
Legend
RMA - Return Material Authorization
SHM - Switch Health Management
RASD - Remote Access Service Daemon
ND - New Device
DR - Drift And Reconcile
MAINT- Maintenance Mode
Update the Device send Events Take The Device out of MainT Mode And Enable Health Check For Device Update RMA Execution I status IF needed
Reconciliation Successful
Success
Tenant Drift & Reconcile End Event
Failed
User Take Corrective Action And Triggers Reconcile REST/CLI
Reconciliation Failed

Figure 47: Drift and Reconcile workflow

Drift and Reconcile Limitations in Tenant Service

Drift and Reconcile must be triggered after Inventory Service completes the event handling which are generated by Tenant-Service epg operations.

The following VLAN based EPG example shows how events are handled by the Inventory Service.

1. Create an VLAN based EPG on the device, D1 with `ctag-range 100`.
2. After successful epg creation, events containing new `epg-config` are published to the message bus.

   Note
   If the vlan(100) is manually deleted and the Inventory service cannot find the vlan(100) on the device D1, delete event for vlan(100) is not generated.

3. Inventory service processes the events, and updates the database through device discovery.

Return Material Authorization

Return Material Authorization (RMA) allows replacement of a faulty switch with a new switch with the same configuration.

RMA process comprises of the following steps:

1. A faulty switch is replaced with a new switch with same configuration.
2. EFA engine initiates the RMA process. The RMA process can be initiated manually using REST/CLI.
3.

4. EFA engine uses the config-backup taken by Switch Health Management and initiates the config-replay on the replacement switch.
5. After config-replay is complete, EFAengine uses Drift and Reconcile to bring in data consistency.

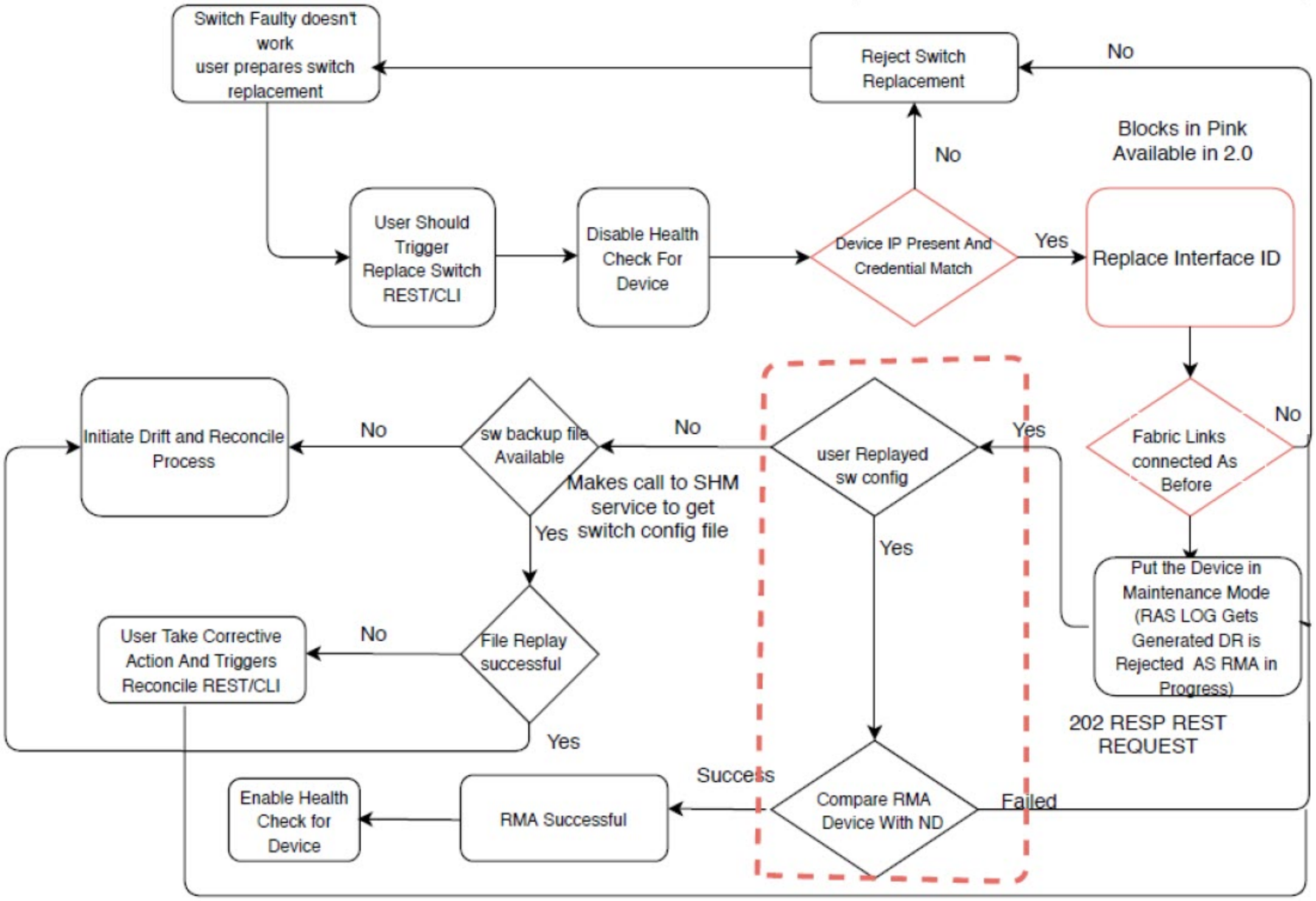

Switch Faulty doesn't work user prepares switch replacement
Reject Switch Replacement
No
Blocks in Pink Available in 2.0
No
User Should Trigger Replace Switch REST/CLI
Disable Health Check For Device
Device IP Present And Credential Match
Yes
Replace Interface ID
Initiate Drift and Reconcile Process
No
sw backup file Available
No
user Replayed sw config
Yes
Fabric Links connected As Before
No
Makes call to SHM service to get switch config file
Yes
Yes
Put the Device in Maintenance Mode (RAS LOG Gets Generated DR is Rejected AS RMA in Progress)
User Take Corrective Action And Triggers Reconcile REST/CLI
No
File Replay successful
Yes
202 RESP REST REQUEST
Success
Enable Health Check for Device
RMA Successful
Compare RMA Device With ND
Failed

Figure 48: RMA workflow

Replace a Faulty Switch

1. Obtain configuration backup of the old device.

   Refer to Configure Backup and Replay on page 153 for more information.

   ```
   # efa inventory config-backup execute --ip 10.24.14.133
   ```
2. Initiate RMA process.

   ```
   # efa inventory rma execute --ip 10.24.14.133 --config-backup-id 1111-1111-111
   ```
3. Initiate drift reconcile to ensure data consistency.

   ```
   # efa inventory drift-reconcile execute --ip 10.24.14.133 --reconcile
   ```
4. View RMA history and detail.

   ```
   # efa inventory rma history -ip 10.24.14.133
   # efa inventory rma detail -uuid 123e4567-e89b-12d3-a456-426614174000
   ```

5. View drift reconcile history and detail.

   ```
   # efa inventory drift-reconcile history --device-ip 10.24.14.133
   # efa inventory drift-reconcile detail --uuid 1111-1111-1111
   ```

6. (Optional) Delete RMA record.

   ```
   # efa inventory rma delete –ip 10.24.14.133
   ```

7. (Optional) Disable drift reconcile.

   ```
   # efa inventory drift-reconcile delete --key 10.24.14.133
   # efa inventory drift-reconcile delete --key 1111-1111-1111
   ```

SLX Device Configuration

You can perform several configuration tasks on the SLX devices that EFA manages.

Compare a Device

You can view the configurations on the device that are out of sync with the configurations in the Asset service.

This helper utility displays a summary of the information in the Asset database.

View a summary of the information in the Asset database.

```
efa inventory device compare --ip <IP address of the device>
```

Enable Maintenance Mode on SLX Devices

You can enable maintenance mode on the SLX devices that EFA manages.

By default, EFA performs Drift and Reconcile actions on the SLX devices that are in maintenance mode, taking those devices out of maintenance mode after a designated interval. For more information about Drift and Reconcile, see Drift Identification and Reconcile on page 154.

You can enable maintenance mode on SLX devices without triggering Drift and Reconcile. Take the following steps.

1. Disable syslog.

   ```
   efa inventory device execute-cli --ip 10.18.120.187
   --command "no logging syslog-server 10.18.120.140 use-vrf mgmt-vrf" --config
   ```

2. Enable maintenance mode.

   ```
   efa inventory device setting update --maint-mode-enable Yes --ip 10.18.120.187
   ```

   The device remains in maintenance mode until you disable the mode.

3. Disable maintenance mode.

   a. Enable syslog.

      ```
      efa inventory device execute-cli --ip 10.18.120.187
      --command "logging syslog-server 10.18.120.140 use-vrf mgmt-vrf" --config
      ```

   b. Run Drift and Reconcile.

      ```
      efa inventory drift-reconcile execute --ip 10.18.120.187 –reconcile
      ```

      The Drift and Reconcile process takes the device out of maintenance mode.

Configure Physical Port Speed

You can configure the speed for receiving and transmitting data on a physical port.

In SLX-OS, you can use the **show interface ethernet** command to see the speed of the Ethernet interfaces on your device.

Run the **efa inventory device interface set-speed** command.

```
# efa inventory device interface set-speed --ip 10.x.x.x --if-type eth
--if-name 0/31,0/32 --speed 10Gbps

Port Speed Updated Successfully
+----+------+----------------+------------+
| ID | Name | Interface Type | Port Speed |
+----+------+----------------+------------+
| 51 | 0/32 | ethernet       | 10Gbps     |
+----+------+----------------+------------+
| 15 | 0/31 | ethernet       | 10Gbps     |
+----+------+----------------+------------+
Interface Details
--- Time Elapsed: 18.0329535s ---
```

This example sets the speed to 10 Gbps for two ports.

Configure Breakout Ports

You can break a port into multiple interfaces, such as breaking one 40G port into four 10G ports. You can also revert the breakout.

In SLX-OS, you can use the **show running-config hardware** command to see whether breakout mode is configured for a device.

You can break a port into the following modes: one 10g port, one 25g port, one 100g port, two 40g ports, two 50g ports, four 10g ports, and four 25g ports.

The breakout interfaces you create are identified by the name of the original interface followed by a suffix.

When you run revert a breakout, the breakout interfaces are deconfigured and deleted. The original Ethernet interface in the default configuration is created automatically.

1. To break a port into multiple ports, run the following command.

```
# efa inventory device interface set-breakout --ip 10.x.x.x
--if-type eth --if-name 0/52 --mode 4x10g

Breakout Created Successfully
+-----+--------+----------------+
| ID  | Name   | Interface Type |
+-----+--------+----------------+
| 975 | 0/52:3 | ethernet       |
+-----+--------+----------------+
| 976 | 0/52:1 | ethernet       |
+-----+--------+----------------+
| 977 | 0/52:4 | ethernet       |
+-----+--------+----------------+
| 978 | 0/52:2 | ethernet       |
+-----+--------+----------------+
Interface Details
--- Time Elapsed: 3m43.7323188s ---
```

This example breaks interface 0/52 into four 10g ports.

2. To revert the breakout of multiple ports to the original configuration, run the following command.

   ```
   # efa inventory device interface unset-breakout --ip 10.x.x.x
   --if-type eth --if-name 0/52
   ```

   This example removes breakout mode on interface 0/52.

Configure MTU for Physical Ports

You can configure the MTU (maximum transmission unit) at the physical port level for Layer 2, IPv4, and IPv6.

In SLX-OS, you can use the **show interface ethernet** command to see the MTU configuration for an interface.

Run the **efa inventory device interface set-mtu** command.

```
# efa inventory device interface set-mtu --ip 10.x.x.x
--if-type eth --if-name 0/11,0/12 --mtu 1600 --ip-mtu 700 --ipv6-mtu 1800

Interface MTU Successfully Updated
+-----+------+----------------+------+--------+----------+
| ID  | Name | Interface Type | MTU  | IP MTU | IPv6 MTU |
+-----+------+----------------+------+--------+----------+
| 99  | 0/11 | ethernet       | 1600 | 1700   | 1800     |
+-----+------+----------------+------+--------+----------+
| 102 | 0/12 | ethernet       | 1600 | 1700   | 1800     |
+-----+------+----------------+------+--------+----------+
Interface MTU Details
--- Time Elapsed: 42.769852714s ---
```

This example configures the MTU for two Ethernet interfaces for IP address 10.x.x.x.

Change a Physical Port State

You can bring a physical port administratively up or down.

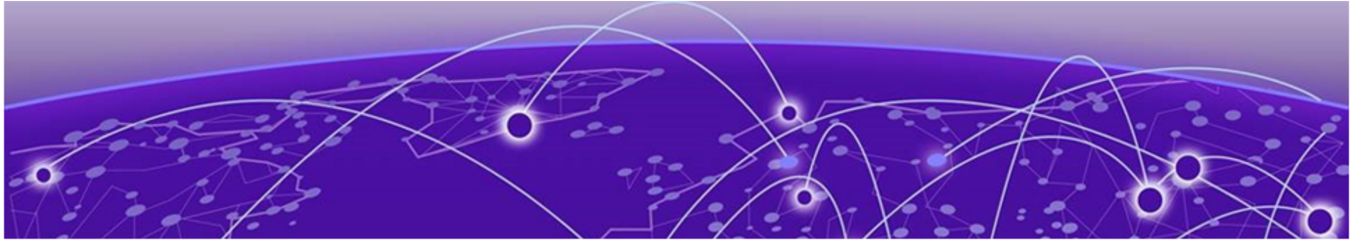
In SLX-OS, you can use the **show interface ethernet** command to see the status of the Ethernet interfaces on your device.

Run the **efa inventory device interface set-admin-state** command.

```
# efa inventory device interface set-admin-state
--ip 10.x.x.x --if-type eth --if-name 0/31,0/32 --state up

Admin-State Updated Successfully
+----+------+----------------+--------------+
| ID | Name | Interface Type | Admin Status |
+----+------+----------------+--------------+
| 51 | 0/32 | ethernet       | up           |
+----+------+----------------+--------------+
| 15 | 0/31 | ethernet       | up           |
+----+------+----------------+--------------+
Interface Details
--- Time Elapsed: 18.235185378s ---
```

This example brings two Ethernet ports into the up state.



EFA Event Management

RASlog Service

The RASlog Service is aware of all devices that are registered with the services in EFA and processes events only from those devices. Messages from other devices are dropped.

The RASlog Service performs the following functions:

- Acts as a syslog server to process syslog messages from devices
- Acts as an SNMP trap receiver to process traps from devices

With the RASlog Service, EFA receives events from network devices and the Inventory service learns of relevant changes. The Inventory Service can fetch the current state of network topology and update Fabric and Tenant services.

RASlog Operations

EFA is registered as a syslog recipient on the devices as part of the device registration. If there are any changes to the link after Fabric or Tenant formation, the RASlog service receives the syslog message.

The sequence of RASlog operations is as follows:

1. The RASlog Service processes the syslog message and notifies all services through message-bus.
2. The Inventory Service receives the RASlog Service message and updates relevant asset details in the database.
3. The Inventory Service notifies Fabric and Tenant Services of any changes in the configurations.
4. Fabric and Tenant Services review the state changes and display information about any pending configurations.

   You can choose to update Fabric or Tenants for the current state.
5. When a device is deleted from the Inventory Service, EFA is unregistered as a syslog recipient from the device. If unregistration of EFA fails, deletion still proceeds.
6. The RASlog Service listens to Device Registration and Device Deletion messages to ensure that messages from registered devices are not dropped.

RASlog Service Examples

Container Details

The RASlog Service listens to Syslog messages on port 514 for both TCP and UDP based packets. Name of the RASlog Service container is `goraslog-service`.

```
root@sam-kub-master:~/GoDCApp/scripts/single-node-deployment# docker-compose ps
Name Command State Ports
------------------------------------------------------------------------------------------
------------------------------------------------------
gofabric-service-v2.0.1 bin/ash -c /opt/fabric/fabric Up (healthy) 0.0.0.0:8081->8081/tcp
goinventory-service-v2.0.1 bin/ash -c /opt/inventory/ ... Up (healthy) 0.0.0.0:8082->8082/
tcp
goraslog-service-v2.0.1 bin/ash -c /opt/raslog/syslog Up 0.0.0.0:514->514/tcp,
0.0.0.0:514->514/udp
goswitch-service-v2.0.1 bin/ash -c /opt/goswitch/g ... Up 0.0.0.0:8084->8084/tcp
gotenant-service-v2.0.1 bin/ash -c /opt/ts/ts-server Up (healthy) 0.0.0.0:8083->8083/tcp
kong-api-gateway-v2.0.1 /docker-entrypoint.sh kong ... Up (healthy) 0.0.0.0:8000->8000/
tcp, 8001/tcp, 0.0.0.0:8002->8002/tcp, 8443/tcp,
8444/tcp
konga-v2.0.1 /app/start.sh Up 0.0.0.0:1337->1337/tcp
postgres-database-v2.0.1 docker-entrypoint.sh postg ... Up (healthy) 0.0.0.0:5432->5432/
tcp
rabbitmq-v2.0.1 docker-entrypoint.sh rabbi ... Up 15671/tcp, 0.0.0.0:15672->15672/tcp,
25672/tcp, 4369/tcp, 5671/tcp,
0.0.0.0:5672->5672/tcp
```

Asset Service Audit Trail

When device update is triggered due to RASlogEvents, the `device_collection_audit` table in Inventory Service is updated.

```
select * from device_collection_audit;
id | device_id | start_time | update_time | status | component | reason |
time_taken_seconds
----+-----------+-------------------------------+-------------------------------+---------
+-----------+-------------------+--------------------
1 | 2 | 2019-08-14 17:19:38.931296-07 | 2019-08-14 17:19:41.491968-07 | Success | |
RaslogEventReason | 2
2 | 2 | 2019-08-14 17:20:18.931769-07 | 2019-08-14 17:20:21.719485-07 | Success | |
RaslogEventReason | 2
```

Support-Save and Logging

The logs from Raslog Service are available in the `/var/log/dcapp/raslog/raslog-server.log` file. This is captured as part of the `supportsave` zip file. The logs are stored for a maximum of 30 days and limited to 10 file with a file size of 100 MB each.

```
root@sam-kub-master:/var/log/dcapp# tree

├── dca-client.log
├── fabric
│ └── fabric.log
├── goswitch
│ ├── fabric
│ │ └── goSwitch.log
│ ├── goswitch
│ │ └── goSwitch.log
│ ├── inventory
│ │ └── goSwitch.log
│ └── ts
│ └── goSwitch.log
```

```
├── installer
│ └── installer_201908141711.log
├── inventory
└── inventory-server.log
├── postgresql
│ ├── 000000010000000000000001
│ ├── archive_status
│ ├── postgresql-2019-08-14_171319.log
│ └── postgresql-2019-08-15_000000.log
├── rabbitmq
│ ├── log
│ │ ├── crash.log
│ │ └── crash.log.0
│ ├── rabbitmq1.log
│ └── rabbit@rabbit1_upgrade.log
├── raslog
│ └── raslog-server.log
└── ts
└── ts.log
```

Notification Service

Notifications are categorized as alerts, tasks, and events.

- Notifications sent from EFA are derived from the syslog events received from the devices that EFA manages.
- Alerts are notifications that services in EFA send for unexpected conditions. They include the following:
  - Loss of switch connectivity
  - Failure to configure the Fabric, tenant, or EPG on the device
  - Failure to perform operations such as port up or port down, set speeds, and breakout mode
  - Firmware download failure
  - A device exiting maintenance mode
- Tasks are user-driven operations or timer-based tasks and consist of the following:
  - Device registration
  - Device update
  - Device timer collection completed
  - Adding devices to fabric
  - Fabric creation and deletion
  - Fabric setting update
  - Fabric configuration
  - Tenant creation, update, or deletion
  - EPG creation, update, or deletion

Notification Workflow

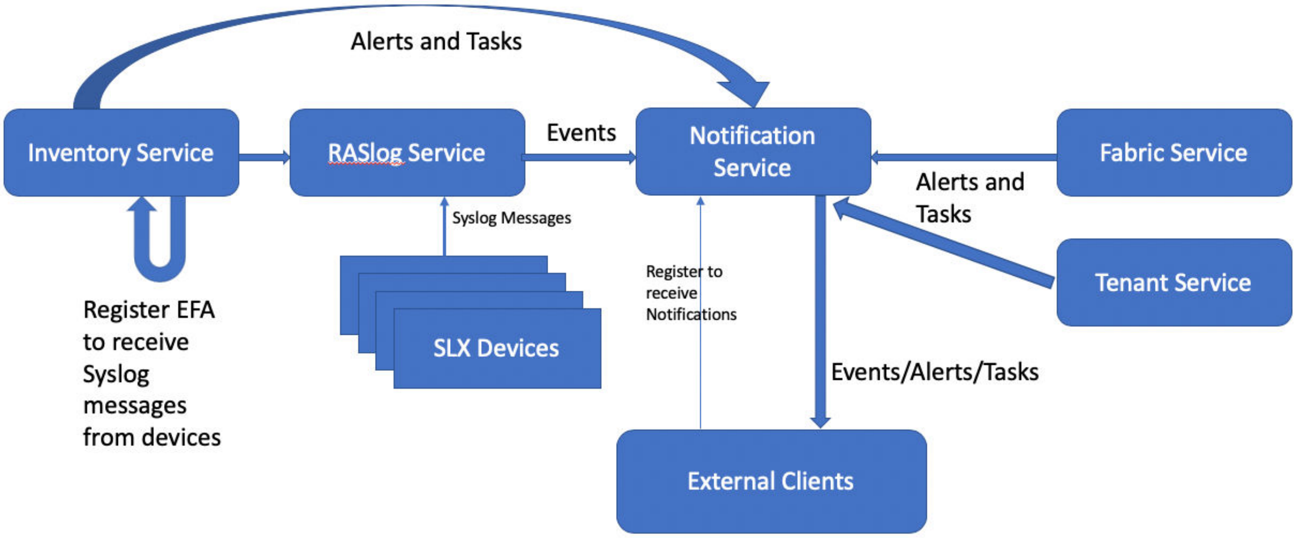


Figure 49: Notification Services Workflow

The sequence of operation is as follows:

- When users register devices in EFA, EFA will register itself as a syslog receiver on each of the device. This way we can filter out events from devices that EFA is managing.
- The RASLog service in EFA will start receiving Syslog messages from the devices.
- The messages are processed and converted to JSON format that is easier to search/sort and perform operations on.
- External Clients register with EFA to receive notifications via webhook. REST APIs will register/ unregister and list all the current subscribed notification receivers.
- Whenever the RASlog service receives an syslog message from the devices, it's notified via webhook to subscribers with the payload in the body of the message.
- Other services within EFA will publish their tasks (user operations) which will also be published as webhook from the notification service.
- Any alerts (failures and error conditions) is notified to subscribers via the notification service.

Notification Subscribers

Users can set up the following subscription type:

- Webhook – REST api based. It is a POST operation with the notification payload in the body of the http(s) call.

Notification Payload

The following is an example of a notification payload.

```
{"level":"info","msg":"Received an event
RAS.Event_Created.","time":"2020-05-28T04:53:04-07:00"}
{"level":"info","msg":"Handling RaslogEvent: \u0026{EventHeader:
{EventID:RAS.Event_Created PublishTime:2020-05-28T04:53:04-07:00 Auth:{Basic:\u003cnil
\u003e Token:\u003cnil\u003e}} DeviceIP:10.20.61.104 Message:admin desc=Userssh
desc=connection methodThu May 28 12:09:04 2020 desc=last_successful_login_time BOMLogin
information: User admin via ssh Last Successful Login Time : Thu May 28 12:09:04 2020.
MessageID:SEC-1206 Module: Severity:INFO Hostname:SLX Username: Timestamp:
2020-05-28T12:10:24.877106}","time":"2020-05-28T04:53:04-07:00"}
{"level":"info","msg":"Received an event
RAS.Event_Created.","time":"2020-05-28T04:53:05-07:00"}
{"level":"info","msg":"Handling RaslogEvent: \u0026{EventHeader:
{EventID:RAS.Event_Created PublishTime:2020-05-28T04:53:05-07:00 Auth:{Basic:\u003cnil
\u003e Token:\u003cnil\u003e}} DeviceIP:10.20.61.104 Message:admin
desc=User134.141.25.192 desc=IP address BOMLogin information: User admin Login successful
via TELNET/SSH/RSH. IP Addr: 134.141.25.192. MessageID:SEC-1203 Module: Severity:INFO
Hostname:SLX Username: Timestamp:
2020-05-28T12:10:25.880126}","time":"2020-05-28T04:53:05-07:00"}
{"level":"info","msg":"Received an event
RAS.Event_Created.","time":"2020-05-28T04:53:05-07:00"}
{"level":"info","msg":"Handling RaslogEvent: \u0026{EventHeader:
{EventID:RAS.Event_Created PublishTime:2020-05-28T04:53:05-07:00 Auth:{Basic:\u003cnil
\u003e Token:\u003cnil\u003e}} DeviceIP:10.20.61.104 Message:login desc=Event NameREMOTE,
IP Addr: 134.141.25.192 desc=connection method and IP Address BOMEvent: login, Status:
success, Info: Successful login attempt via REMOTE, IP Addr: 134.141.25.192.
MessageID:SEC-3020 Module: Severity:INFO Hostname:SLX Username:admin Timestamp:
2020-05-28T12:10:25.881575}","time":"2020-05-28T04:53:05-07:00"}



{"level":"info","msg":"Received an event
APP_NTF.Task_Event.","time":"2020-05-28T04:40:04-07:00"}
{"level":"info","msg":"Handling TaskNotificationEvent: \u0026{EventHeader:
{EventID:APP_NTF.Task_Event PublishTime:2020-05-28T04:40:04-07:00 Auth:{Basic:\u003cnil
\u003e Token:\u003cnil\u003e}} AppName:fabric TaskName:EFA-000001 Scope:user
Username:root Status:started DeviceIP: Message:Fabric create request
received :request={\"FabricDescription\":\"\",\"FabricName\":\"test\",\"FabricStage\":
3,\"Type\":\"clos\"}}","time":"2020-05-28T04:40:04-07:00"}
{"level":"info","msg":"Received an event
APP_NTF.Task_Event.","time":"2020-05-28T04:40:04-07:00"}
{"level":"info","msg":"Handling TaskNotificationEvent: \u0026{EventHeader:
{EventID:APP_NTF.Task_Event PublishTime:2020-05-28T04:40:04-07:00 Auth:{Basic:\u003cnil
\u003e Token:\u003cnil\u003e}} AppName:fabric TaskName:EFA-000002 Scope:user
Username:root Status:succeeded DeviceIP: Message:Fabric create request
success :request={\"FabricDescription\":\"\",\"FabricName\":\"test\",\"FabricStage\":
3,\"Type\":\"clos\"}}","time":"2020-05-28T04:40:04-07:00"}


{"level":"info","msg":"Received an event
RAS.Event_Created.","time":"2020-05-28T04:41:32-07:00"}
{"level":"info","msg":"Handling RaslogEvent: \u0026{EventHeader:
{EventID:RAS.Event_Created PublishTime:2020-05-28T04:41:32-07:00 Auth:{Basic:\u003cnil
\u003e Token:\u003cnil\u003e}} DeviceIP:10.20.61.104 Message:Ethernet 0/47
desc=InterfaceName BOM Interface Ethernet 0/47 is protocol down. MessageID:NSM-1002
Module:Interface Severity:INFO Hostname:SLX Username: Timestamp:
2020-05-28T11:58:53.538052}","time":"2020-05-28T04:41:32-07:00"}
{"level":"info","msg":"Received an event
RAS.Event_Created.","time":"2020-05-28T04:41:32-07:00"}
{"level":"info","msg":"Handling RaslogEvent: \u0026{EventHeader:
```

```
{EventID:RAS.Event_Created PublishTime:2020-05-28T04:41:32-07:00 Auth:{Basic:\u003cnil
\u003e Token:\u003cnil\u003e}} DeviceIP:10.20.61.104 Message:Ethernet 0/47
desc=InterfaceName BOM Interface Ethernet 0/47 is link down. MessageID:NSM-1003
Module:Interface Severity:INFO Hostname:SLX Username: Timestamp:
2020-05-28T11:58:53.538078}","time":"2020-05-28T04:41:32-07:00"}


{"level":"info","msg":"Received an event
RAS.Event_Created.","time":"2020-05-28T04:57:02-07:00"}
{"level":"info","msg":"Handling RaslogEvent: \u0026{EventHeader:
{EventID:RAS.Event_Created PublishTime:2020-05-28T04:57:02-07:00 Auth:{Basic:\u003cnil
\u003e Token:\u003cnil\u003e}} DeviceIP:10.20.61.104 Message:Port-channel 26
desc=InterfaceName BOM Port-channel 26 is created. MessageID:NSM-1004 Module:
Severity:INFO Hostname:SLX Username: Timestamp:
2020-05-28T12:14:22.133559}","time":"2020-05-28T04:57:02-07:00"}
{"level":"info","msg":"Received an event
RAS.Event_Created.","time":"2020-05-28T04:57:02-07:00"}
{"level":"info","msg":"Handling RaslogEvent: \u0026{EventHeader:
{EventID:RAS.Event_Created PublishTime:2020-05-28T04:57:02-07:00 Auth:{Basic:\u003cnil
\u003e Token:\u003cnil\u003e}} DeviceIP:10.20.61.104 Message:database commit transaction
desc=Event NameSucceeded desc=Command statusconfigure config interface Port-channel 26
desc=ConfD hpath string BOMEvent: database commit transaction, Status: Succeeded, User
command: configure config interface Port-channel 26. MessageID:DCM-1006 Module:
Severity:INFO Hostname:SLX Username:admin Timestamp:
2020-05-28T12:14:22.224708}","time":"2020-05-28T04:57:02-07:00"}


{"level":"info","msg":"Received an event
RAS.Event_Created.","time":"2020-05-28T05:00:47-07:00"}
{"level":"info","msg":"Handling RaslogEvent: \u0026{EventHeader:
{EventID:RAS.Event_Created PublishTime:2020-05-28T05:00:47-07:00 Auth:{Basic:\u003cnil
\u003e Token:\u003cnil\u003e}} DeviceIP:10.20.61.104 Message:Ethernet 0/40
desc=InterfaceName BOM Interface Ethernet 0/40 is administratively down.
MessageID:NSM-1020 Module:Interface Severity:INFO Hostname:SLX Username: Timestamp:
2020-05-28T12:18:07.943449}","time":"2020-05-28T05:00:47-07:00"}
{"level":"info","msg":"Received an event
RAS.Event_Created.","time":"2020-05-28T05:00:47-07:00"}
{"level":"info","msg":"Handling RaslogEvent: \u0026{EventHeader:
{EventID:RAS.Event_Created PublishTime:2020-05-28T05:00:47-07:00 Auth:{Basic:\u003cnil
\u003e Token:\u003cnil\u003e}} DeviceIP:10.20.61.104 Message:Ethernet 0/40
desc=InterfaceName BOM Interface Ethernet 0/40 is link down. MessageID:NSM-1003
Module:Interface Severity:INFO Hostname:SLX Username: Timestamp:
2020-05-28T12:18:07.946265}","time":"2020-05-28T05:00:47-07:00"}
{"level":"info","msg":"Received an event
RAS.Event_Created.","time":"2020-05-28T05:00:48-07:00"}
{"level":"info","msg":"Handling RaslogEvent: \u0026{EventHeader:
{EventID:RAS.Event_Created PublishTime:2020-05-28T05:00:48-07:00 Auth:{Basic:\u003cnil
\u003e Token:\u003cnil\u003e}} DeviceIP:10.20.61.104 Message:database commit transaction
desc=Event NameSucceeded desc=Command statusconfigure conf-if-eth-0/40 shutdown
desc=ConfD hpath string BOMEvent: database commit transaction, Status: Succeeded, User
command: configure conf-if-eth-0/40 shutdown. MessageID:DCM-1006 Module: Severity:INFO
Hostname:SLX Username:admin Timestamp:
2020-05-28T12:18:07.973754}","time":"2020-05-28T05:00:48-07:00"}
```

Types of notifications can be Event, Alert or Task.