Extreme Fabric Automation Administration Guide

2.7.1

9037424-01 Rev AA
June 2022

Copyright © 2022 Extreme Networks, Inc. All rights reserved.

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/

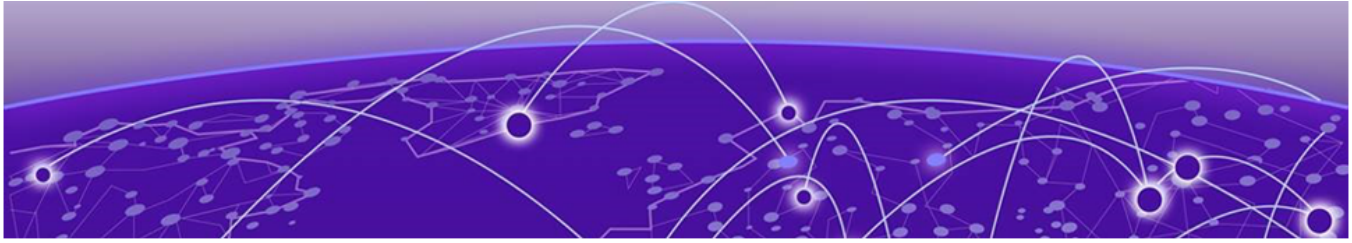


Table of Contents

Preface **9**
- Text Conventions 9
- Documentation and Training 10
- Help and Support 11
  - Subscribe to Product Announcements 11
- Send Feedback 11

About this Document **13**
- What's New in this Document 13

Extreme Fabric Automation **14**
- Introduction to Extreme Fabric Automation 14
  - CLI and API 15
  - Deployment 15
- EFA Microservices 16
  - Fabric Service 16
  - Tenant Service 17
  - Inventory Service 17
  - Notification Service 17
  - RASlog Service 17
  - Security Service 18
  - SNMP Service 18
  - Policy Service 18
  - Ecosystem Integration Services 18
- REST API Documentation for EFA 19
  - 20
  - Access Northbound API 20

EFA System Management **21**
- Verify the Running System and Services 21
- Log in to EFA 23
- EFA Certificate Management 23
  - Northbound Interface Certificates 24
  - Device Certificates 24
  - Certificate Expiry and Renewal 26
  - Certificate Troubleshooting 28
- Monitoring EFA Status 29
- Verifying EFA System Health 29
  - SLX Device Health 29
  - EFA Services Health 30
  - RabbitMQ Liveness 30
  - EFA System Health for High-availability Deployments 30
  - Node Health 30

EFA System Backup and Restoration....30
Manual Backup and Restore....30
Periodic Backups and Configuration....31
Backups During Upgrades....31
Logs....31
Back up and Restore the EFA System....32
Change the Host Name or IP Address....33
Display EFA Running Configurations....34
Audit Trail Logging....35
Transfer of Audit Trail Data....35
Logging and Log Files....36
Data Consistency....36
Overview....36
Limitations....37
Periodic Device Discovery....38
Persistent Configuration....38
Drift and Reconcile....39
Network Elements....40
Idempotent Operations....49
Rollback Scenarios for Data Consistency....50
EFA High Availability Failover Scenarios....51
SLX Device Failure....51
SLX Device Failure on the Active K3s Agent Node....53
SLX Device Failure on the Standby K3s Agent Node....53
TPVM Failure....53
Two-node Failure....53
Multiple Management IP Networks....53
Overview....53
Assumptions....54
Add and Delete Management Routes....55
Add and Delete Management Subinterfaces....56
Configure Static IP Addresses for Management Subinterfaces....57
Change the Default Gateway of a TPVM....57

Fabric Infrastructure Provisioning....59

Fabric Service Overview....59
IP Fabric and Clos Orchestration Overview....60
SLX Device Prerequisites for Fabric Service....60
Clos Overview....61
3-Stage Clos....61
5-Stage Clos....61
Configure a 3-Stage Clos Fabric....62
Configure a 5-Stage Clos Fabric....64
Overview of Day-0 Operations for a Small Data Center Fabric....66
Supported Small Data Center Topologies....66
Configure a Small Data Center Fabric....68
Dynamic ICL in small data center....69
IP Multicast Fabric Provisioning....69
IP Multicast Fabric Overview....69
Bidirectional Forwarding Detection....70

Fabric Settings to Update BGP MD5 Password, BGP Dynamic Peer Listen Limit, and Single Rack Deployment ... 72
Configure an IP Multicast Fabric ... 74
Device Configuration ... 75
Configure Drift and Reconcile on Multicast Fabric ... 75
View Fabric Details ... 76
Edit Fabric Settings ... 77
Edit Fabric IP Range Settings ... 77
Edit Fabric BFD Settings ... 78
Fabric Settings in Active Fabric: Small Data Center Fabric ... 79
Fabric Settings in Active Fabric: Clos Fabric ... 80
Fabric Event Handling ... 80
Northbound IPv6 support ... 81
Verify the Upgrade ... 81
Verify the IPv6 Address ... 82

Tenant Services Provisioning ... 83
Tenant Services Provisioning Overview ... 83
Tenant ... 85
VLAN-based Tenant ... 85
Bridge domain-based Tenant ... 85
Scalability ... 85
Event handling ... 85
Provisioning a Tenant ... 86
Create a Tenant ... 86
Create a Port Channel ... 86
Create the Tenant VRF ... 86
Create the Tenant Endpoint Group ... 87
Create the BGP Peer Group ... 87
Create the BGP Peer ... 88
Clos Fabric with Non-auto VNI Maps ... 88
Clos Fabric with Auto VNI Map ... 89
Multi Tenancy ... 89
Layer 3 Network Services ... 93
Configure Description on Port Channel ... 93
Configure Minimum Link Count on Port Channel ... 94
Configure Backup Routing on Tenant VRF ... 96
Route-Distinguisher (RD) allocation independent of Route-Target (RT) ... 98
Configure Static VRF Route ... 100
Configure BFD on Static VRF Route ... 101
Configure Local ASN on Tenant VRF ... 101
Enable Graceful Restart on Tenant VRF ... 108
Configure Resilient Hashing on Tenant VRF ... 108
Enable Graceful Restart on Tenant VRF ... 109
Configure MaxPaths on Tenant VRF ... 110
Configure Redistribute Attribute on Tenant VRF ... 111
Configure Advertise Network and Static-Network on Tenant VRF ... 112
Configure Aggregate Address on Tenant VRF ... 113
EPG: Network Property Description ... 115
Update anycast-ip on an Existing Tenant Network ... 118

Configure Multiple Anycast IP....120
Configure IPv6 ND on a Tenant Network....123
Configure the BFD Session Type for an Endpoint Group....126
Configure CEP Cluster Tracking for Endpoint Groups....126
Suppress Address Resolution Protocol and Neighbor Discovery on VLAN or Bridge Domain....127
Configure Local IP for Endpoint Group....131
EPG: Network Property: IP MTU....133
BGP as a Service....133
IPv6 Support....156
Software BFD Session Support on CEP....156
Exclusion of VLANs and Bridge from Cluster Instance....164
Sharing Resources Across Tenants using Shared Tenant....165
Distributed and Centralized Routing....179
Route-Distinguisher (RD) allocation independent of Route-Target (RT)....188
BFD timers for router-bgp BFD and static-route BFD sessions....190
Administered Partial Success....191
Overview....191
Tips and considerations....192
Behavior changes during "admin down" state....192
Behavior changes during "admin up" state....193
Administratively Manage a Device State....194
APS Behavior of Tenant Configuration....194
In-flight Transaction Recovery....206
Overview....206
Examples....208
Traffic Mirroring....208
In-band Traffic Mirroring....210
Out-of-band Traffic Mirroring....211
Support Matrix....213
Configure Port-Based Mirroring in a Multi-Tenant Architecture....214
Configure Flow-Based Mirroring in a Multi-Tenant Architecture....217
Configure VLAN-Based Mirroring in a Multi-Tenant Architecture....221
Configure ICL Port Mirroring in a Multi-Tenant Architecture....225
Scalability....229
Scaled REST Request Timeout....229
Scaled DRC Timeout....229

Policy Services Provisioning....230
Policy Services Provisioning Overview....230
Database, REST API and inter-service communication....230
Inventory Service interactions....230
Prefix List....230
Configure IP Prefix List on Devices....231
Drift and Reconcile (DRC) and Idempotency for IP prefix list configuration....232
Route Map....233
....233
Configure Route Map on devices....233
Drift and Reconcile (DRC) and Idempotency for route map configuration....236
Event handling for IP prefix list and route map....237

Inventory service....237
Policy service....237

EFA Device Management....238
Device Image Management....238
Limitations....238
Supported devices....239
Hitless Firmware Upgrade....239
Firmware Download....246
Fabric-wide Firmware Download....250
Roll Back Device Firmware....258
Traffic Loss Scenarios....258
Device Health Management....260
Monitor Device Health....261
Device Configuration Backup and Replay....261
Configure Backup and Replay....262
Return Material Authorization....263
Replace a Faulty Device....264
SLX Device Configuration....265
Compare a Device....265
Enable Maintenance Mode on SLX Devices....266
Configure Physical Port Speed....266
Configure Breakout Ports....267
Configure MTU at the Interface or System Level....270
Change the Admin Status of an Interface....272
Configure NTP at Device and Fabric Levels....273
Configure RME on Interface....274
Device Configuration Synchronization....275
SLX Configuration Backup....275

EFA Event Management....279
RASlog Service....279
RASlog Operations....279
Notification Service....280
Overview....280
Notification methods....280
Notification workflow....281
Notification example....281
EFA as SNMP Proxy....282
Drift and Reconcile (DRC) and Idempotency for SNMP....284

Known Limitations....287
....0
Known Limitations in EFA....287
Provided VNI: 1006 already consumed in fabric....287
VRF delete from EPG and re-adding VRF to EPG fails intermittently....287
vrf_route_target_mapping error while creating EPG (after VRF delete)....288
Inventory device update fails to change device table router-ip field....288
REST operations are not retried (as applicable) during the service boot....288
RBAC: EFA shows "export EFA_TOKEN" command suggestion when a tenant user logs in....289

EFA CLI or REST request with scale config takes longer than 15 minutes fails..... 290

Preface

Read the following topics to learn about:

- The meanings of text formats used in this document.
- Where you can find additional information and help.
- How to reach us with questions and comments.

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
	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 is displayed 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/.

Help and Support

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 Product Announcements

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

1. Go to The Hub.
2. In the list of categories, expand the **Product Announcements** list.
3. Select a product for which you would like to receive notifications.
4. Select **Subscribe**.
5. To select additional products, return to the **Product Announcements** list and repeat steps 3 and 4.

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

Send Feedback

The Information Development team at Extreme Networks has made every effort to ensure that this document is accurate, complete, and easy to use. We strive to improve our documentation to help you in your work, so we want to hear from you. We welcome all feedback, but we especially want to know about:

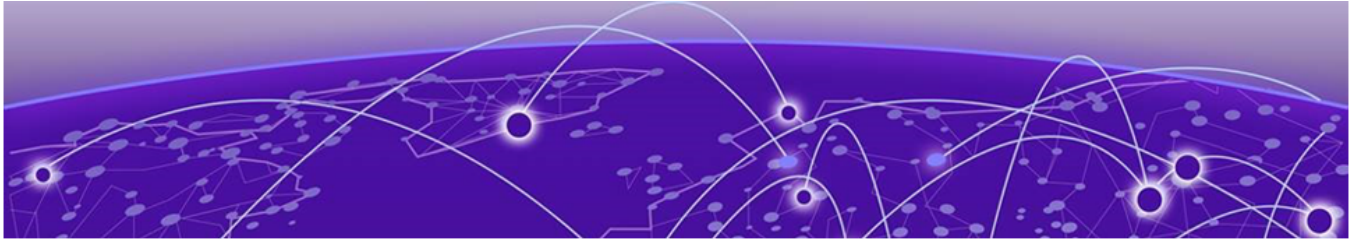
- Content errors, or confusing or conflicting information.

- Improvements that would help you find relevant information.
- Broken links or usability issues.

To send feedback, do either of the following:

- 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

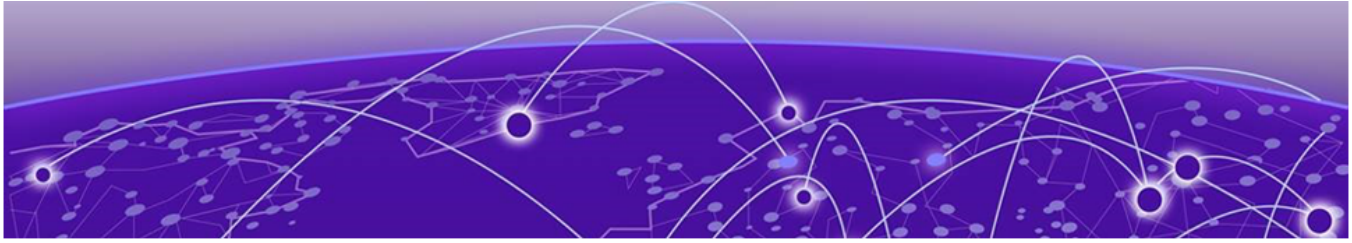
What's New in this Document on page 13

What's New in this Document

The following table describes information added to this guide for the Extreme Fabric Automation 2.7.1 software release.

Table 4: Summary of changes

Feature	Description	Link
Northbound API access	Describes how to access northbound API in EFA.	Access Northbound API on page 20



Extreme Fabric Automation

Introduction to Extreme Fabric Automation on page 14
EFA Microservices on page 16
REST API Documentation for EFA on page 19

Introduction to Extreme Fabric Automation

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

EFA automates and orchestrates SLX IP fabrics and tenant networks, with support for the following:

- Building and managing small data center (non-Clos) fabrics and 3-stage and 5-stage IP Clos fabrics
- Managing tenant-aware Layer 2 and Layer 3 networks
- Integrating Virtual Management ecosystem platforms, such as VMWare vCenter, OpenStack, and Microsoft SCVMM
- Providing a single point of configuration for your entire fabric

EFA consists of core K3s containerized services that interact with each other and with other infrastructure services to provide the core functions of fabric and tenant network automation. For more information, see EFA Microservices on page 16.

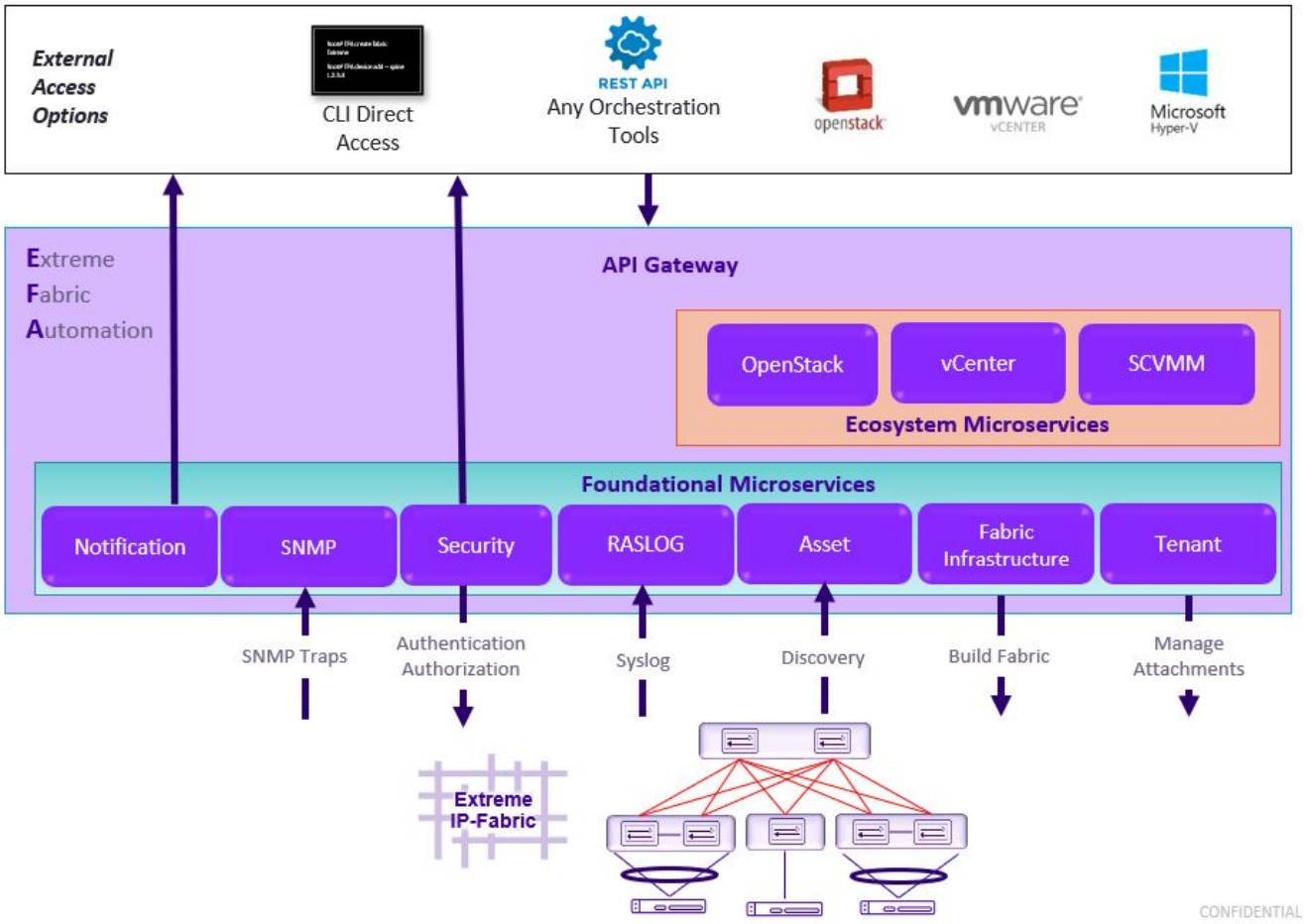


Figure 1: EFA orchestration

CLI and API

Using the built-in command and OpenAPI-based REST APIs, you can discover physical and logical assets, build and manage fabrics, manage the EFA system, and configure security. For more information, see the *Extreme Fabric Automation Command Reference, 2.7.0* and REST API Documentation for EFA on page 19.

Deployment

For more information about deployment scenarios, see the *Extreme Fabric Automation Deployment Guide, 2.7.0*.

EFA on TPVM

TPVM (Third-Party Virtual Machine) is a guest VM that resides on Extreme SLX devices. You can run EFA from the SLX 9150, SLX 9250, or SLX 9740 TPVM. In this context, EFA leverages the K3S Kubernetes cluster as an underlying infrastructure for the EFA services deployment. The K3S cluster is a single instance and an important component for supporting high availability. A maximum of 24 devices is supported, either 24 devices in one fabric or 24 devices across multiple fabrics.

EFA 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 in the data center.

EFA for high availability

A high-availability cluster is a group of servers that provide continuous up time, or at least minimum down time, for the applications on the servers in the group. If an application on one server fails, another server in the cluster maintains the availability of the service or application. You can install EFA on a two-node cluster, including on TPVM, for high availability.

EFA Microservices

EFA consists of core K3s containerized microservices that interact with each other and with other infrastructure services to provide the core functions of fabric and tenant network automation.

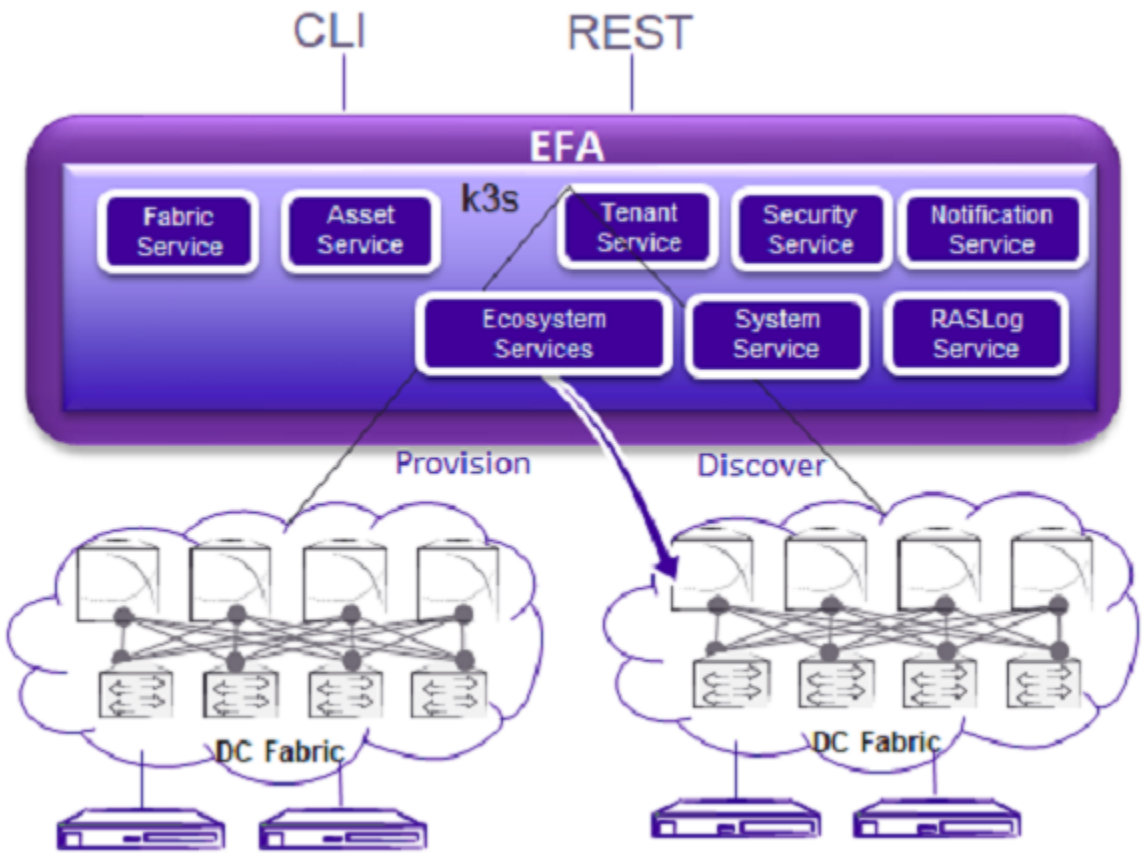


Figure 2: Microservices in the EFA architecture

Fabric Service

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

The Fabric Service features include:

- Support for small data centers (non-Clos)
- Support for 3-stage and 5-stage Clos fabrics
- Support for MCT configuration

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.

Tenant Service

The Tenant Service manages tenants, tenant networks, and endpoints, fully leveraging the knowledge of assets and the underlying fabric. You can use the CLI and REST API for tenant network configuration on Clos and small data center fabrics.

Tenant network configuration includes VLAN, BD, VE, EVPN, VTEP, VRF, and router BGP configuration on fabric devices to provide Layer 2 extension, Layer 3 extension across the fabric, Layer 2 hand-off, and Layer 3 hand-off at the edge of the fabric.

Inventory Service

The Inventory Service acts as an inventory of all the necessary physical and logical assets of the fabric devices. All other EFA services rely on asset data for their configuration automation. 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 to devices. Examples include configuring SNMP parameters or OSPF configurations. This means you can use EFA for SLX-OS commands and push the same configuration to multiple devices.

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 processes syslog messages from devices and forwards notifications to subscribers. For more information, see RASlog Service in the *Extreme Fabric Automation Administration Guide, 2.7.0*.

Security Service

The Security Service consists of authentication and authorization features that enforce a security boundary between northbound clients and downstream operations between EFA and SLX devices. The service also validates users and their credentials through Role-based Access Control (RBAC) and supports local and remote (LDAP) login.

Note

If you configure LDAP server over SSL, and use IP to connect to the server, ensure that the certificate includes the IP as part of SANs for a successful connection.

SNMP Service

The SNMP Service processes SNMP traps from devices and forwards notifications to subscribers. For more information, see EFA as SNMP Proxy in the *Extreme Fabric Automation Administration Guide, 2.7.0*.

Policy Service

Policy Service in EFA manages and configures IP prefix lists and route maps on fabric devices. It subscribes to the inventory service to receive events including device registration, device deletion, and changes to previously identified IP prefix lists and route maps.

Ecosystem Integration Services

EFA provides one-touch integration 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 Service

The vCenter integration provides connectivity between EFA and vCenter using a REST API. EFA does not connect to individual ESXi servers. All integration is done through vCenter. For more information, see the *Extreme Fabric Automation VMware vCenter Integration Guide, 2.7.0*. Integration support includes the following:

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

Hyper-V

The Hyper-V integration supports networking configuration for Hyper-V servers in a datacenter, manual and automated configuration updates when VMs move, and visibility into the VMs and networking resources that are deployed in the Hyper-V setup. For more information, see *Extreme Fabric Automation Hyper-V Integration Guide, 2.7.0*. Integration support includes the following:

- 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 Service

The OpenStack service integrates Extreme OpenStack plugins with the rest of the EFA foundation services in an IP fabric. For more information, see the *Extreme Fabric Automation OpenStack Integration Guide, 2.7.0*. Integration support includes the following:

- Create, read, update, delete (CRUD) operations on networks and ports
- LAG support
- Provider network (default, PT)
- VLAN trunking
- Network operations using single-root I/O virtualization (SR-IOV), physical and virtual functions
- vMotion (virtual machine migration)
- ML2 driver with support for:
  - Network and segment provisioning for non-default physnets
  - DC-owner-based l2 extension for DC gateway
- Topology changes for port-based extension of DC gateway addition and deletion of topology entries and its changes on EFA engpoint groups.
- Single-homed connections to the edge port
- Multi-segment support
- Journaling support for L2 and L3
- L3 service plugin:
  - Routing feature support using VRF
  - Flavor (service provider) support
  - Centralized routing
  - IPv6 support (dual stack)
- Layer 3 flavors
- Neighbor Discovery and Router Advertisement support:
  - IPv6 ND MTU support
  - IPv6 No-Autoconfig support

REST API Documentation for EFA

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

The REST API is specified by OpenAPI and Swagger.

Specific API guides for the EFA services are available on the Extreme Networks website. Select **Extreme Fabric Automation** here: https://www.extremenetworks.com/support/documentation/product-type/software/. And then select the version of EFA you want to work with.

API guides are available for the following services:

- Authorization service
- Fabric service

- Hyper-V service
- Inventory service
- Monitoring service
- Notification service
- OpenStack service
- RASlog service
- RBAC service
- SNMP service
- System service
- Tenant service
- vCenter service
- Policy service

Access Northbound API

This topic describes setting up northbound API access in EFA.

Note
Do not use this process for OpenStack instance.

Fetch the access token using username and password.

```
curl -X POST 'https://10.x.x.x/v1/auth/token/access-token' \
--header 'Content-Type: application/json' \
--data '{
    "username": "xxxxxx",
    "password": "xxxxxx"
}'
```

The response contains the access token.

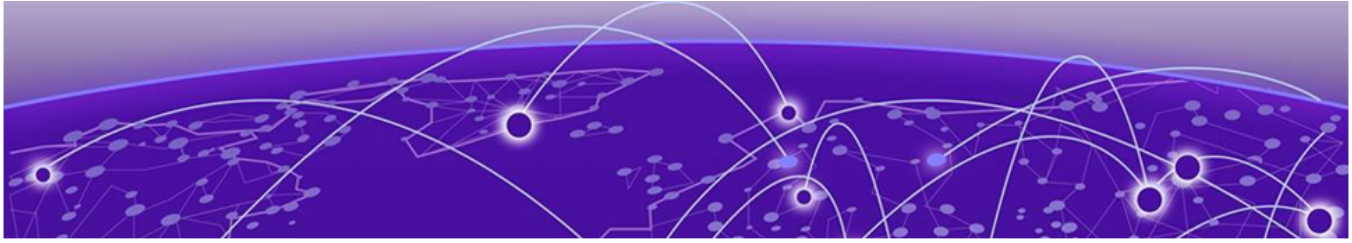
```
{
    "access-token": "eyJhbGciOi… ",
    "token-type": "Bearer",
    "refresh-token": "eyJhbGciOi…"
}
```

Any request to the northbound API must have the access token in the request header.

For example, to fetch all the switches in inventory, run the following command:

```
curl -X GET 'https://10.x.x.x/v1/inventory/switches' \
--header 'Authorization: Bearer eyJhbGciOi…'
```

Note
The expiry for the access token is set to one hour by default. You can increase the expiry by using the **`efa auth settings token update`** command.



EFA System Management

Verify the Running System and Services on page 21
Log in to EFA on page 23
EFA Certificate Management on page 23
Monitoring EFA Status on page 29
Verifying EFA System Health on page 29
EFA System Backup and Restoration on page 30
Change the Host Name or IP Address on page 33
Display EFA Running Configurations on page 34
Audit Trail Logging on page 35
Logging and Log Files on page 36
Data Consistency on page 36
EFA High Availability Failover Scenarios on page 51
Multiple Management IP Networks on page 53

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
=====================================================
               EFA version details
=====================================================
Version : 2.7.0
Build: 30
Time Stamp: 22-04-23:12:37:36
Mode: Secure
Deployment Type: multi-node
```

```
Deployment Platform: TPVM
Virtual IP: 10.20.59.183
Node IPs: 10.20.59.181,10.20.59.182
--- Time Elapsed: 12.253203ms ---

================================================
               EFA Status
================================================
+-----------+---------+--------+--------------+
| Node Name | Role    | Status | IP           |
+-----------+---------+--------+--------------+
| node-1    | active  | up     | 10.20.59.182 |
+-----------+---------+--------+--------------+
| node-2    | standby | up     | 10.20.59.181 |
+-----------+---------+--------+--------------+
--- Time Elapsed: 6.888985085s ---
```

Output varies by type of deployment, such as single-node or multi-node, and the services that are installed.

2. View details of EFA nodes, PODs, and services.
   a. Run the **efa status** command.

      On a multi-node installation:

```
+-----------+---------+--------+---------------+
| Node Name | Role    | Status | IP            |
+-----------+---------+--------+---------------+
| tpvm2     | active  | up     | 10.20.216.242 |
+-----------+---------+--------+---------------+
| tpvm1     | standby | up     | 10.20.216.241 |
+-----------+---------+--------+---------------+
--- Time Elapsed: 4.277420974s ---
```

      On a single-node installation:

```
+-----------+--------+--------+-------------+
| Node Name | Role   | Status | IP          |
+-----------+--------+--------+-------------+
| efa       | active | up     | 10.21.90.43 |
+-----------+--------+--------+-------------+
--- Time Elapsed: 1.461512261s ---
```

      These examples show 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.

```
extreme@node-1:~$ k3s kubectl -n efa get pods

NAME                              READY   STATUS    RESTARTS   AGE
efa-api-docs-m6dgg                1/1     Running   0          19h
gosnmp-service-vmmf7              1/1     Running   0          19h
goopenstack-service-lf45g         1/1     Running   0          19h
rabbitmq-ckvrq                    1/1     Running   0          19h
gorbac-service-2v8qh              1/1     Running   0          19h
gonotification-service-9tjbg      1/1     Running   0          19h
goauth-service-8zltv              1/1     Running   0          19h
goinventory-service-gsxzb         1/1     Running   0          19h
gohyperv-service-ng95v            1/1     Running   0          19h
govcenter-service-rt9cc           1/1     Running   0          19h
gofabric-service-nnv4m            1/1     Running   0          19h
gosystem-service-wxnww            1/1     Running   0          19h
gopolicy-service-hrl7d            1/1     Running   0          19h
```

```
goraslog-service-fqdjm          1/1     Running   0          19h
gotenant-service-v5nq2          1/1     Running   0          19h
```

In a multi-node installation, only the pods on the active node are in "Running" status.

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 using the **efactl restart-service <service-name>** command.
6. Identify the active node that serves as the database for Kubernetes 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 (or the **efa status** command, as an alternative).
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 Certificate Management

EFA requires certificates for the northbound interface and certificates for devices.

For information about EFA certificates and external certificates, see "EFA Certificates" in the *Extreme Fabric Automation Security Guide, 2.7.1*.

.

Northbound Interface Certificates

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> --cacert
cert-filename
```

Important

- If you install your own server certificate to use with the EFA HTTPS server, remember to reinstall the certificate when you upgrade EFA.
- Generate the third-party certificates and keys without a passphrase. Certificate installation may fail if you generate the third-party certificates and keys with passphrase.

Communication with third-party certificates in an EFA installation is enabled on the following ports:

- 443: Secure installation of EFA
- 8078: Monitoring service of EFA

For information about third-party certificates in a multiple management IP network, see Configuration Supporting Multiple Management IP Networks on page 55.

For a multi-node deployment, EFA uses the common name (CN) of the virtual IP address and a Subject Alternate Name containing the virtual IP address and the node IP addresses.

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 multi-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
```

Device Certificates

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

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.
- A syslog certificate is installed on the device.

- 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."
- Managed devices contain the expiration date of an inventory certificate. If a device certificate is within 30 days of expiration, it pushes an event to the notification using the **`get certs`** command.

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.

HTTPS Certificates

When you register a device in EFA, a new certificate is generated for the HTTPS server of SLX device. The certificate is generated with the default CA that EFA contains.

Following is an example of a certificate on SLX after device registration:

```
slx-171# show crypto ca certificates

Certificate Type: https; Trustpoint: none
certificate:
SHA1 Fingerprint=C1:F1:2C:BF:1A:47:7B:46:5D:8F:18:99:0E:58:CF:31:8C:58:5F:CC
Subject: CN=slx-10.x.x.x.extremenetworks.com
Issuer: C=US, ST=CA, O=Extreme Networks, OU=Extreme Fabric Automation Intermediate,
CN=EFA Intermediate CA/emailAddress=support@extremenetworks.com
Not Before: Jan 10 11:12:18 2022 GMT
Not After : Jan 10 11:12:18 2024 GMT
```

To use third-party certificates for HTTPS server on SLX, the CLI command of certificates on the EFA is extended. You need new certificate and key to install on the device. You can use the CLI command only to install certificates on a single device at once.

```
(efa:extreme)extreme@tpvm:/apps/test/certs$ efa certificate device install --ip=10.x.x.x
        --cert-type https --https-certificate server.crt --https-key
      my_server.key

WARNING: This will restart the HTTP service on the devices and services will not be able
to connect
        till the operation is complete. Do you want to proceed [y/n]?

y
+--------------+---------+
| IP Address   | Status  |
| 10.20.61.171 | Success |
+--------------+---------+
--- Time Elapsed: 38.516844258s ---
```

The device must have the new certificates uploaded:

```
slx-171# show crypto ca certificates

Certificate Type: https; Trustpoint: none
certificate:
SHA1 Fingerprint=D8:49:5F:12:AC:FE:BB:CB:95:C2:AC:6B:AF:B6:5B:9E:24:66:59:7D
Subject: CN=10.x.x.x/subjectAltName=IP=10.20.61.171
Issuer: C=US, O=xyz, OU=abcd, CN=INTERIM-CN
```

```
Not Before: Feb 10 11:23:36 2022 GMT
Not After : Jun 25 11:23:36 2023 GMT
```

Syslog CA

This topic describes about the third-party certificates for RASlog service (syslog from SLX).

EFA is shipped with default certificates. These are self-signed and the same certificates are used for listening to the syslog messages received from SLX.

```
$ efa inventory device register --ip=10.x.x.x --username=admin --password=password
+----+-------------+-----------+-------+--------------+----------+---------+--------+
| ID | IP Address | Host Name | Model | Chassis Name | Firmware | Status | Reason |
+----+-------------+-----------+-------+--------------+----------+---------+--------+
| 1 | 10.x.x.x | SLX | 3012 | SLX9250-32C | 20.2.3d | Success | |
+----+-------------+-----------+-------+--------------+----------+---------+--------+
Device Details
--- Time Elapsed: 1m6.570042048s ---
```

The syslog certificate on the device is the default CA the EFA contains.

```
SLX# show crypto ca certificates
syslog CA certificate(Server authentication):
SHA1 Fingerprint=A3:E8:F6:CB:46:F6:43:C5:D1:90:1F:A7:C6:58:93:29:77:6F:2F:8E
Subject: C=US, ST=CA, O=Extreme Networks, OU=Extreme Fabric Automation Intermediate,
CN=EFA Intermediate CA/emailAddress=support@extremenetworks.com
Issuer: C=US, ST=CA, L=SJ, O=Extreme Networks, OU=Extreme Fabric Automation,
CN=efa.extremenetworks.com/emailAddress=support@extremenetworks.com
Not Before: Feb 20 22:25:26 2020 GMT
Not After : Feb 17 22:25:26 2030 GMT
```

The enhancement updates RASlog service to use the custom certificates that EFA servers use. The certificate CLI on EFA contains new parameter, which enables you to upload CA.

```
$ efa certificate server --certificate=my_server_162.pem --key=my_server_162.key --
cacert=ca-chain.pem
Please wait as the certificates are being installed...
Certificates were installed!
--- Time Elapsed: 30.946303683s ---
```

If there are already registered devices, then the syslog certificate is updated on these devices.

```
SLX# show crypto ca certificates
syslog CA certificate(Server authentication):
SHA1 Fingerprint=32:70:EB:91:F4:6D:9C:9F:6E:35:E0:00:20:B8:1A:FF:AF:BA:0D:8A
Subject: C=US, O=xyz, OU=abcd, CN=INTERIM-CN
Issuer: C=US, O=xzy, OU=abcd, CN=ROOT-CN
Not Before: Feb 15 14:56:08 2022 GMT
Not After : Nov 11 14:56:08 2024 GMT
```

If you do not provide any CA certificate, the default certificates of EFA are used.

Certificate Expiry and Renewal

Notifications for EFA Certificate Expiry

EFA is shipped with self-signed certificates. You can also install third-party certificates. When these certificates expire, EFA will not be operational.

This service periodically checks the expiration status of certificates that the system generates and sends notifications when a certificate gets close to its expiration date.

A warning message is displayed if a certificate is going to expire in 30 days from EFA timestamp. If you do not renew the certificates within 7 days of expiry, a warning message is displayed on every login to the EFA CLI.

Following is a sample notification message if the certificate is going to expire in 30 days:

```
{"@time":"2022-04-06T23:59:12.924962 IST","level":"info","msg":"Handling
AppNotificationEvent: \u0026{EventHeader:{EventID:APP_NTF.App_Event
PublishTime:2022-04-06T23:59:12+05:30 Auth:{Basic:\u003cnil\u003e Token:\u003cnil\u003e}
RequestID:} AppName:auth Severity:critical DeviceIP: Message:The certificate for 'EFA'
will expire on '2022-04-08 14:43:43 +0530 IST'.}"}
```

Following is a sample warning message if the certificate is going to expire in 7 days:

```
(efa:extreme)extreme@tpvm:/apps/test/certs$ efa login
Password:
Login successful.
Warning: The certificate for 'EFA' will expire on '2022-04-08 14:43:43 +0530 IST'.
--- Time Elapsed: 5.532391719s ---
```

EFA Certificate Renewal

When you get a certificate expiry notification, run the following command to renew a certificate. This is not applicable if the third-party certificates are installed on the system.

```
(efa:xmcdev)xmcdev@xmcdev-virtual-machine1:/opt/checkouts/efa/efa$ efa certificate server
renew
Certificate renewal is successful
--- Time Elapsed: 33.516064167s ---
```

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 verify 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`

There is a security violation on the switch when EFA (installed on TPVM) logs in and tries to access the swtich with different usernames. You observe the following logs on SLX console:
`1018 AUDIT, 2021/10/14-17:26:57 (GMT), [SEC-3021], INFO, SECURITY, extreme/root/10.20.32.141/ssh/CLI,, SLX, Event: login, Status: failed, Info: Failed login attempt through REMOTE, IP Addr: 10.20.32.141`
`1017 AUDIT, 2021/10/14-17:26:55 (GMT), [SEC-3020], INFO, SECURITY, admin/admin/10.20.32.141/ssh/CLI,, SLX8720-32C, Event: login, Status: success, Info: Successful login attempt via REMOTE, IP Addr: 10.20.32.141`
`1002 AUDIT, 2021/10/14-17:26:41 (GMT), [SEC-3020], INFO, SECURITY, admin/admin/10.20.32.141/ssh/CLI,, SLX8720-32C, Event: login, Status: success, Info: Successful login attempt via NETCONF, IP Addr: 10.20.32.141`
	Try the following:
	• Ensure that you have correctly followed the system restore process.
	• Ensure that all the devices are registered.
	• Ensure that the certificates are installed on the devices to enable secure connections. Run the **`efa certificates device install --ips <ip-adddr> certType [ http|token]`** command to install the HTTPS or OAuth2 certificate on one or more devices..

Monitoring EFA Status

The Monitoring service provides REST API to monitor the status of the various services running in EFA.

The service runs on the host and is exposed on port 8078, which is not the port where the EFA application is running. In a multi-node deployment, this service is available on both nodes and can be accessed through the virtual IP (VIP).

- To start or stop the Monitoring service, run the **`systemctl stop/start/restart monitor.service`** command as a sudo or root user.

  For information, see REST API Documentation for EFA on page 19.
- Use the **`efa status`** command to verify application status.

  For more information, see the **`efa status`** command in the *Extreme Fabric Automation Command Reference, 2.7.0*.

Verifying EFA System Health

This topic describes methods for verifying the health of various EFA services.

SLX Device Health

By default, health check functionality is deactivated when SLX devices are registered. You can verify the status of the functionality with the following EFA command.

```
$ efa inventory device setting show --ip <ip-addr>

+---------------------------------+-------+
| NAME | VALUE |
+---------------------------------+-------+
| Maintenance Mode Enable On | No |
| Reboot | |
+---------------------------------+-------+
| Maintenance Mode Enable | No |
+---------------------------------+-------+
| Health Check Enabled | No |
+---------------------------------+-------+
| Health Check Interval | 6m |
+---------------------------------+-------+
| Health Check Heartbeat Miss | 2 |
| Threshold | |
+---------------------------------+-------+
| Periodic Backup Enabled | Yes |
+---------------------------------+-------+
| Config Backup Interval | 24h |
+---------------------------------+-------+
| Config Backup Count | 4 |
+---------------------------------+-------+
--- Time Elapsed: 270.251797ms ---
```

You can enable health check functionality on the device. And you can configure EFA to regularly back up the device configuration (every 6 minutes by default). For more information, see Configure Backup and Replay on page 262.

If the threshold for missed heartbeats is exceeded, EFA begins the drift and reconcile process after connectivity to the device is re-established. For more information, see Drift and Reconcile on page 39.

EFA Services Health

All services in EFA have internal health REST APIs that Kubernetes uses to restart pods that are deemed unhealthy. The results of a liveness probe determines whether a pod is healthy. Typical values for liveness probes are as follows:

- initialDelaySeconds: 60
- periodSeconds: 10
- timeoutSeconds: 15

RabbitMQ Liveness

The EFA message bus is the workhorse for asynchronous inter-service communication. Therefore, EFA uses the RabbitMQ built-in ping functionality to determine the liveness of the RabbitMQ pod.

As part of a health check, each EFA service also validates its connection to RabbitMQ and attempts to reconnect to RabbitMQ when necessary.

EFA System Health for High-availability Deployments

During installation or upgrade of EFA, a systemd service called `efamonitor` is set up. This service runs validations every minute to check EFA database cluster, glusterFS, and RabbitMQ are functioning correctly.

As needed, the `efamonitor` service remediates the MariaDB Galera cluster and RabbitMQ connection issues, and logs the stats of the system.

Node Health

To ensure that the active and standby nodes are operational, ping checks occur between the nodes. The pings determine whether the active node is up and running. If not, the virtual IP addresses are switched over to the other node.

To ensure that failover does not occur due to a network issue, if a ping to the peer fails, a ping is also attempted to the default gateway. If ping to default gateway fails, ping is attempted to any alternative gateway that may have been provided during installation or upgrade.

If all of the pings fail, keepalived triggers Kubernetes to switch over to the active node and to put the other node in a Fault state.

EFA System Backup and Restoration

The backup process saves EFA data, including the database, certificates, and multi-access network configuration. The process does not back up northbound certificates.

Manual Backup and Restore

The backup process creates a backup tar file. You can select from all saved tar files during the restore process. The tar files are saved to one of the following locations.

- Server: `/var/log/efa/backup`

- TPVM: `/apps/efa_logs/backup`

A backup generated on one EFA system can be restored on another system.

For more information, see Back up and Restore the EFA System on page 32.

You can use the **`efa system backup-list`** command to see the backup files that are available to use in a restore operation. For example:

```
$ efa system backup-list
+----+-----------------------------------+---------+----------------+
| ID    | File                           | Version | Generated By   |
+----+-----------------------------------+---------+----------------+
| 1     | EFA-2020.08.20-20.26.46.tar    | 2.3.0-1 | User           |
+----+-----------------------------------+---------+----------------+
| 2     | EFA-2020.08.20-20.27.29.tar    | 2.3.0-GA| System         |
+----+-----------------------------------+---------+----------------+
--- Time Elapsed: 183.69386ms ---
```

Periodic Backups and Configuration

EFA periodically backs up the system. The periodic backup process creates a backup tar file that is saved to the same location as the manual backup files. When new backup and supportsave files are created, the system deletes saved system-generated backup files, supportsave files, and manual backup files, according to the age of the files and the configured maximum number of files to save.

You can use the **`efa system settings update`** command to determine the backup schedule and to change the maximum number of backup files to save.

- The default backup schedule is: `0:*:*:*`, meaning every day at midnight.
- The default maximum number of files is: five backup files and five supportsave files.

For more information, see the *Extreme Fabric Automation Command Reference, 2.7.0*.

You can use the **`efa system settings show`** command to view the current backup settings.

Tip

You can use the **`efa system cleanup`** REST API to delete a specified backup or supportsave file. This feature lets you delete files before they are automatically deleted.

You can use the **`efa system settings reset`** command to reset the backup system settings to default values.

Backups During Upgrades

The process of upgrading EFA also backs up the EFA system, so that you can easily recover data if the upgrade fails. For more information, see "Recover from an Upgrade Failure" in the *Extreme Fabric Automation Deployment Guide, 2.7.0*.

Logs

Logs related to backup, restore, and supportsave operations are saved to the following locations.

- Server: `/var/log/efa/system` and `/var/log/efa/monitor`

- TPVM: `/apps/efa_logs/system` and `/apps/efa_logs/monitor`

The REST APIs for backup, restore, and supportsave are part of the Monitoring service and can be accessed through port 8078. The logs for these APIs are saved to `<log_dir>/monitor/`

Back up and Restore the EFA System

You can back up and restore the EFA system, including the database and certificates.

1. To back up the system, run the following command.

```
$ efa system backup
Generating backup of EFA...
Backup Location: /apps/efa_logs/backup/EFA-2021-02-10T13-21-46.413.tar
--- Time Elapsed: 10.401999384s ---
```

2. To restore the system, take the following steps.
   a. Run the **`efa system restore`** command.

```
$ efa system restore

EFA-2021.01.12-11.19.52.tar (Version:2.3.2-GA, Generated by: User)
EFA-2021-01-12T04.59.09.tar (Version:2.4.0-7171, Generated by: User)
EFA-2021-01-12T13.50.00.tar (Version:2.4.0-121, Generated by: User)
EFA-2021-01-12T13.50.51.tar (Version:2.4.0-121, Generated by: System)
EFA-2021-01-12T16.11.47.tar (Version:2.4.0-1211, Generated by: System)
EFA-Upgrade-2.3.2-GA.tar (Version:2.3.2-GA, Generated by: System)
```

   The command output displays a list of available backup tar files.
   b. Select the backup tar file that you want to restore.

```
Choose backup option:1
Selected: EFA-2021.01.12-11.19.52.tar
Performing EFA restore using EFA-2021.01.12-11.19.52.tar
Generating backup before initiating restore
BACKUP_TAR: /apps/efa_logs/backup/EFA-2021.01.12-11.19.52.tar
Stopping all EFA services
All pods are terminated
Migrating database
Completed database migration
Checking if all PODS are in ready state...
Restore operation is successful
--- Time Elapsed: 9m3.079104969s ---
```

   c. When the restore is complete, run **`source /etc/profile`**.

      You can now log in to EFA.
   d. To enable secure connections, install the certificates on devices.

```
efa certificates device install --ips <ip-adddr> certType [ http|token]
```

      The command installs the HTTPS or OAuth2 certificate on one or more devices.
   e. To get the current state of the devices, run the **`efa inventory device update`** command after you run the restore command.
   f. Check the status of the services to ensure that they are in-sync.

Change the Host Name or IP Address

You can change the host name, the IP address, and the virtual IP address after EFA is deployed.

Review the following host name requirements:

- Host names must be unique and consist of numeric characters and lowercase alphabetic characters. Do not use uppercase alphabetic characters.
- Hyphens are the only special characters allowed. No other special characters are allowed by Kubernetes for cluster formation or by the K3s service.

Host name changes are supported in single-node and multi-node deployments.

IP address changes are supported in single-node deployments.

Virtual IP address (VIP) changes are supported in multi-node deployments.

1. To change the host name, take the following steps.
   a. In a multi-node deployment, stop the monitoring service.

      ```
      sudo systemctl stop efamonitor
      ```

   b. Change the host name of the system, such as with the following Linux command.

      ```
      hostnamectl set-hostname <new name>
      ```

   c. Add the new host name to `/etc/hosts`.
   d. Run the following command as a root user or as a user with sudo privileges.

      Important
      Do not reboot the system before running this command.

      ```
      sudo bash efa-change-hostname <old host name>

      Reading host name of the system
      Restarting mariadb service
      Restarting k3s service
      Checking k3s for the new host name
      Host is in ready state in k3s
      Setting current host as active node
      Deleting old host name references
      Waiting for EFA containers to start
      Successfully updated host name in EFA
      ```

      In a single-node deployment, EFA is not operational during this step. In a multi-node deployment, EFA remains operational if the command is running on the standby node. EFA is not operational if the command is running on the active node.

      In a TPVM deployment, you can run the command from `/apps/bin/`.

   e. In a multi-node deployment, start the monitoring service.

      ```
      sudo systemctl start efamonitor
      ```

2. To change the IP address of a single-node deployment, take the following steps.
   a. Run the following command as a root user or as a user with sudo privileges.

      ```
      sudo bash efa-change-ip

      Updating IP in EFA
      Restarting k3s service
      Updating all files with new IP
      ```

```
Deleting EFA services: gonotification-service gofabric-service gotenant-service
goauth-service gorbac-service goinventory-service goopenstack-service
govcenter-service gohyperv-service goraslog-service efa-api-docs gosystem-service
Waiting for EFA containers to start
Successfully updated IP in EFA
```

EFA is not operational during this step.

In a TPVM deployment, you can run the command from `/apps/bin/`.

b. After the IP address is updated, run `source /etc/profile` or open a new EFA session to log in.

3. To change the VIP of a multi-node deployment, take the following steps.

a. Stop the monitoring service.

```
sudo systemctl stop efamonitor
```

b. Run the following command as a root user or as a user with sudo privileges.

```
sudo bash efa-change-vip <new vip>

Updating all files with new VIP.
Restarting services on nodes
Waiting for EFA containers to start
Updating services with new VIP
Waiting for EFA containers to start
Successfully updated VIP for the installation.
```

EFA is not operational while this script runs.

In a TPVM deployment, you can run the command from `/apps/bin/`.

c. Start the monitoring service.

```
sudo systemctl start efamonitor
```

d. Run `source /etc/profile` or open a new EFA session to log in.

Note

After the Virtual IP address (VIP) update is successful, it takes a few minutes to update new VIP in all the registered devices in EFA.

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.

Note

The output of **`efa show-running-config`** command is also captured as part of the supportsave zip file.

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.

Logging and Log Files

Log directories vary by deployment type and by application.

EFA logs are saved to the following locations:

- Non-TPVM 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 deployments: `/apps/efa_logs`
- Kubernetes log files: `/var/log/pods`
- Keepalived service log files: `/supportsave/keepalived`. The directory contains Keepalived service logs and the `journalctl` log of Keepalived for the past day. This information is useful for helping debug failovers, double faults, and gateway connectivity.

In multi-node, high availability deployments, logs are replicated on all nodes in the cluster.

The **`efa system supportsave`** script gathers all logs, database dumps, pod logs, deployment details, and system support-save 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 describes how data is rendered consistent among EFA services.

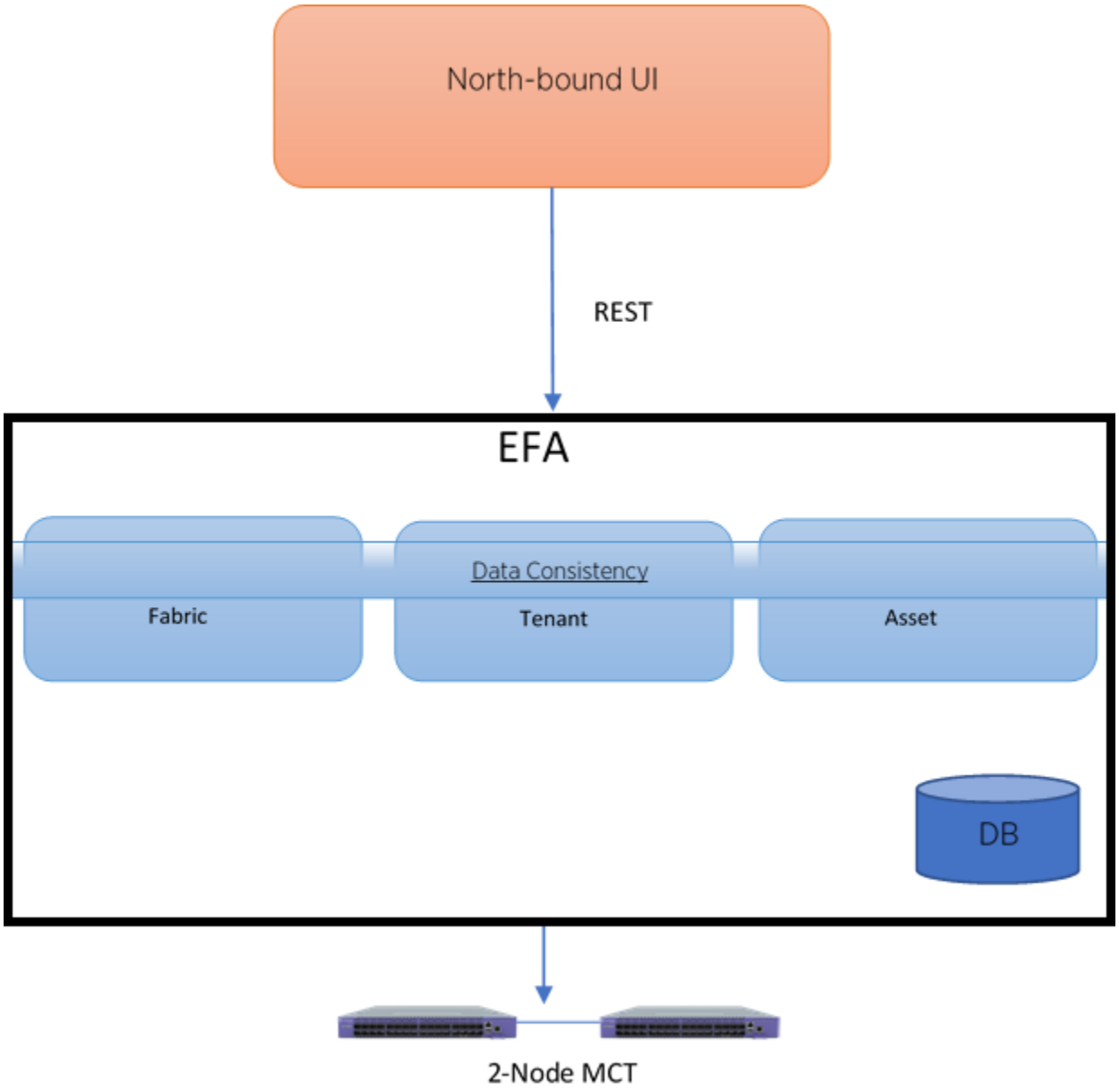


Figure 3: Data consistency overview

North-bound applications invoke 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.

Periodic Device Discovery

Asset, Tenant, Fabric, and Policy Services use periodic discovery to detect out-of-sync configurations on the devices. These 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 full device discovery using the **efa inventory device update** command.

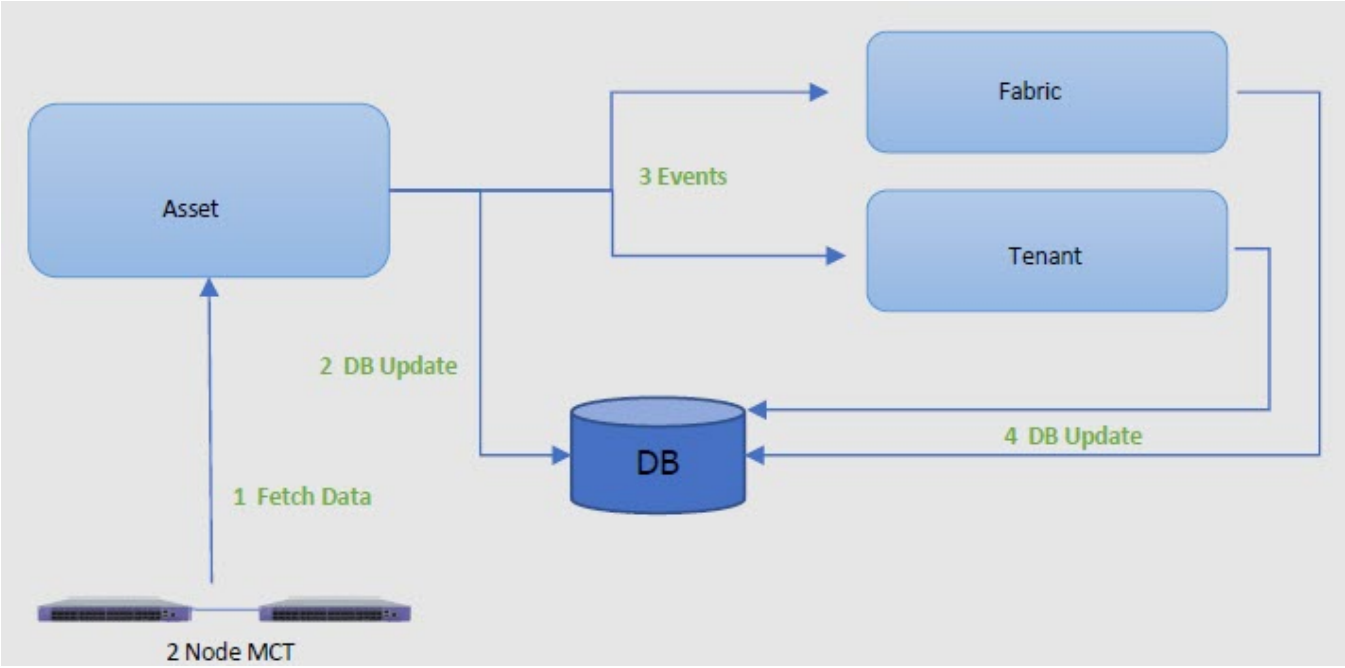


Figure 4: Device discovery and database updates

The Asset service periodically polls the devices in the fabric and keeps the database and other services updated of any changes in the underlying fabric. The default polling interval is one 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 fabric. You can use the **efa inventory device discovery-time update** command to configure the discovery interval.

EFA determines out-of-band configuration changes on the devices. If there are no out-of-band configuration changes, the device updates are optimized.

Note
Periodic device discovery is compatible with SLXOS 20.4.1 and later.

Persistent Configuration

Extreme devices support three types of configuration files:

- Default - Default configuration files are part of the firmware package for the device and are automatically applied to the startup configuration.

- Startup - Startup configuration files are persistent and are applied after system reboot.
- Running - Configuration currently effective on the device is the running configuration.

For more information on configuration files, see Extreme SLX-OS Management Configuration Guide.

In SLX-OS 20.1.1, the configuration management process maintains two databases, Running and Startup.

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

- The **show running-config** command fetches the configuration from the database.
- The **copy running-config startup-config** command creates or updates the persistent configuration.
- After a upgrade or downgrade, replaying the startup file resumes the SLX database cleanup operations.

Maintenance Mode

In SLX-OS 20.1.1, maintenance mode can be enabled by configuring **enable** under system-maintenance configuration mode. If the configuration is persistent, the switch needs to be in maintenance mode before rebooting for it to come back in maintenance mode.

In SLX-OS 20.2.1 and later, maintenance mode can be enabled by configuring **enable-on-reboot** under system-maintenance configuration mode. After the reboot, the device 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.

Non-reachable Devices

EFA tracks devices by running heart-beats to the SLX devices.

When a non-reachable device becomes reachable, EFA identifies any drift and performs reconciliation, if necessary.

Drift and Reconcile

EFA supports drift and reconcile (DRC) of a configuration at device level. A single device configuration is compared with EFA. If there is a drift in the configuration, it is conciled. EFA provides APIs to initiate drift and reconcile requests. Use the EFA command **efa inventory drift-reconcile execute** to run a manual DRC.

Drift and reconcile is also activated during the following operations:

- Switch replacement and RMA
- After the reboot of a device in maintenance mode
- Device firmware-download with “drc” flag

Drift and reconcile operations are run in parallel across all devices in the fabric. It ensures that the multiple DRC operations that take place during fabric-wide firmware-download (FWDL) or reboot of multiple devices together, run in parallel, and hence, reduce the overall maintenance window.

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.

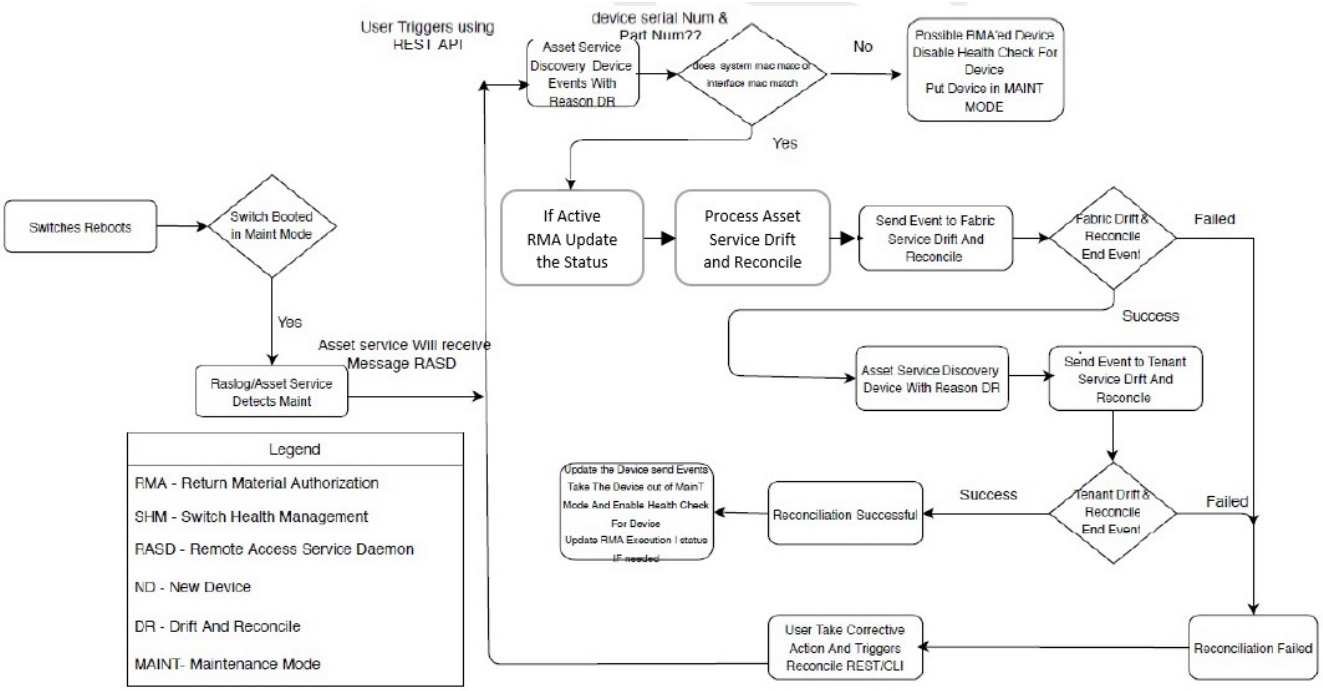


Figure 5: Drift and reconcile workflow

Note

When any attribute under "router bgp" is drifted, EFA also reconciles the cluster configuration to ensure that the BGP neighbors of MCT are reconciled, and this shows up as cluster reconciled success in addition to routerbgp.

Network Elements

Starting in EFA v2.5.0, in addition to fabric and tenant service configurations, the following asset service configurations are persisted and included in Drift and Reconcile (DRC).

The support is on two levels:

- Interface level configuration: Breakout mode, MTU, admin state, speed, FEC configuration, port dampening (link-error disable), and RME
- Global or system level configuration: NTP, SNMP v2 and v3, prefix list, and route map

Interface-level Configuration

The following table captures the various attributes of interface for which DRC and idempotency is supported.

- A drift is identified if any of the fields below is modified through the SLX, CLI command, or other management tool.
- A reconcile operation pushes the intended configuration to SLX, so bringing the SLX configuration in sync with EFA.

Note

Clean up explicitly any conflicting configuration which could cause reconciliation of device to fail. For example, if EFA configures a port as breakout and if that configuration is drifted by adding Layer 3 configuration to a parent interface, the reconciliation fails. It is recommended to explicitly remove the conflicting configuration from the device through the SLX CLI and retry the DRC process.

Table 5: Interface attributes supporting DRC and Idempotency

Field	Identify Drift	Reconcile configuration	Idempotency
Admin-state	Yes	Yes	Yes
Breakout mode	Yes	Yes	No
Speed	Yes	Yes	Yes
Layer-2 MTU	Yes	Yes	Yes
IPv4 MTU	Yes	Yes	Yes
IPv6 MTU	Yes	Yes	Yes
FEC mode	Yes	Yes	Yes
Link error disable	Yes	Yes	Yes
Toggle-threshold	Yes	Yes	Yes
Sampling time	Yes	Yes	Yes
Wait time	Yes	Yes	Yes
RME enable	Yes	Yes	Yes

The following CLI commands are available:

- `efa inventory device interface redundant-management`
- `efa inventory device interface set-fec`
- `efa inventory device interface set-link-error-disable`
- `efa inventory device interface unset-fec`
- `efa inventory device interface unset-link-error-disable`

For more information, see *Extreme Fabric Automation Command Reference, 2.7.0*.

Global or System-level Configuration

- A drift is identified if any of the fields below is modified through the SLX, CLI command, or other management tools.
- A reconcile operation pushes the intended configuration to SLX, so bringing the SLX configuration in sync with EFA.

The following CLI commands are available:

- `efa inventory device setting update --prefix-independent-convergence`
- `efa inventory device setting update --maximum-load-sharing-paths`
- `efa inventory device settings update --mct-bring-up-delay`
- `efa inventory device settings update --maint-mode-convergence-time`

For more information, see *Extreme Fabric Automation Command Reference, 2.7.0*.

Device Reload

The device reload command allows the user to reload a device. Users can provide IPs and fabric name separated by commas. All devices in a fabric will be reloaded for any given fabric name in the command.

The following CLI command is available for device reload:

- `efa inventory device reload`

Note
Drift and reconcile and idempotency configuration support is not applicable for device reload attribute.

For more information, see *Extreme Fabric Automation Command Reference, 2.7.0*.

Clear IP Route

The clear IP route command allows the user to clear a device's IPv4 and IPv6 routes. Users have the option to either clear an IPv4 or an IPv6 route.

The following CLI command is available for clearing IP route:

- `efa inventory device clear route-all`

Note
Drift and reconcile and idempotency configuration support is not applicable for clear IP route attribute.

For more information, see *Extreme Fabric Automation Command Reference, 2.7.0*.

SNMP Configuration

The following tables capture the various attributes of SNMP and NTP for which DRC and idempotency are supported.

Note
Regarding idempotency for creating an entry which already exists in EFA, an error message is returned stating that the user already exists.

Table 6: SNMP attributes supporting DRC and Idempotency

Field	Identify Drift	Reconcile Configuration	Idempotency	Notes
Community deleted	Yes	Yes	No	A valid error message is shown when a non-existent community is deleted
Group name associated with community is modified	Yes	Yes	Not Applicable	
Group deleted	Yes	Yes	Not Applicable	
Modify group version	No	No	Not Applicable	SLX does not support editing the SNMP group version
Modify read review or write view or notify view associated with group	Yes	Yes	Not Applicable	
Modify group name associated with SNMP user	Yes	Yes	Not Applicable	
Modify authentication protocol associated with SNMP user	Yes	Yes	Not Applicable	
Modify authentication password associated with SNMP user	Yes	Yes	Not Applicable	
Modify privacy protocol associated with SNMP user	Yes	Yes	Not Applicable	
Modify privacy password associated with SNMP user	Yes	Yes	Not Applicable	
Delete SNMP user	Yes	Yes	No	A valid error message is shown when a non-existent user is deleted
Modify encrypted keyword associated with SNMP user	Yes	Yes	Not Applicable	

Table 6: SNMP attributes supporting DRC and Idempotency (continued)

Field	Identify Drift	Reconcile Configuration	Idempotency	Notes
Modify authentication type associated with group, that is, auth, noauth, notify	Yes	Yes	Not Applicable	
Delete SNMP host entry	Yes	Yes	No	A valid error message is shown when a non-existent host is deleted
Update SNMP host severity level	Yes	Yes	Not Applicable	Any drift observed from EFA configured default severity level is reconciled
Update SNMP host source interface	Yes	Yes	Not Applicable	Any drift observed from EFA configured default source interface is reconciled
Update SNMP host UDP port	Yes	Yes	Not Applicable	Any drift observed from EFA configured default UDP port is reconciled
Update SNMP host VRF	Yes	Yes	Not Applicable	Any drift observed from EFA configured default VRF is reconciled
Update SNMP host engine id	Yes	Yes	Not Applicable	
Update of SNMP host notification type (traps, informs)	Yes	Yes	Not Applicable	
Update of SNMP view MIB OID access (included, excluded)	Yes	Yes	Not Applicable	

The following CLI commands are available for operations on SNMP interfaces:

- `efa inventory device snmp community create`
- `efa inventory device snmp community delete`
- `efa inventory device snmp community list`
- `efa inventory device snmp user create`
- `efa inventory device snmp user delete`
- `efa inventory device snmp user list`
- `efa inventory device snmp host create`
- `efa inventory device snmp host delete`
- `efa inventory device snmp host list`

For more information, see the *Extreme Fabric Automation Command Reference, 2.7.0*.

NTP Configuration

The NTP commands let you configure NTP server configuration on the SLX device. The configuration you set is persisted in the EFA database. DRC is also supported.

The following table captures the various attributes of the NTP configuration interface for which DRC and idempotency is supported. A drift is identified if any of the following fields are modified by you through SLX CLI or other management tools. Reconcile operation pushes the intended configuration to SLX which makes the SLX configuration synchronize with EFA.

On idempotency for creating an entry which already exists in EFA, an error message is returned stating that user already exists.

Table 7: NTP attributes supporting DRC and Idempotency

Field	Identify Drift	Reconcile configuration	Idempotency	Notes
NTP auth key ID associated with NTP serer is modified	Yes	Yes	Not Applicable	
NTP auth key name associated with NTP serer is modified	Yes	Yes	Not Applicable	
NTP server deleted	Yes	Yes	No	A valid error message is shown when a non existent NTP server is deleted.
Encryption type is modified	Yes	Yes	Not Applicable	
Trusted key is modified	Yes	Yes	Not Applicable	
Encryption level is modified	Yes	Yes	Not Applicable	
NTP server disable modified	Yes	Yes	Not Applicable	

The following CLI commands are available for operations on NTP interfaces:

- `efa inventory device ntp server create`
- `efa inventory device ntp server delete`
- `efa inventory device ntp server list`
- `efa inventory device ntp disable-server`

For more information, see the *Extreme Fabric Automation Command Reference, 2.7.0*.

IP prefix list configuration

Note
Regarding idempotency for creating an entry which already exists in EFA, an error message is returned stating that the user already exists.

Table 8: IP prefix list attributes supporting DRC and Idempotency

Field	Identify Drift	Reconcile configuration	Idempotency	Notes
IPv4 prefix list rule is deleted.	Yes	Yes	No	Deleted rule will be reconciled.
IPv4 prefix list is deleted.	Yes	Yes	No	Deleted prefix list along with all rules associated with it will be reconciled.
IPv4 prefix list rule created OOB. Different rules exist with same prefix list name in EFA.	No	No	Not applicable	Delete the OOB rule or keep it and don't act as part of DRC.
IPv4 prefix list rule created OOB. Different rules exist with same prefix list name and sequence number in EFA.	Yes	Yes	Not applicable	Prefix list rule will be reconciled to be in sync with EFA.
Create an IPv4 prefix OOB with a prefix list name not matching any of the EFA created entries.	No	No	Not applicable	These are treated as out of band entries and EFA will not perform DRC.

The following CLI commands are available for operations on IP prefix lists:

- **`efa policy prefix-list create`**
- **`efa policy prefix-list list`**
- **`efa policy prefix-list delete`**
- **`efa policy prefix-list update`**

For more information, see the *Extreme Fabric Automation Command Reference, 2.7.0*.

Route map configuration

Note
Regarding idempotency for creating an entry which already exists in EFA, an error message is returned stating that the user already exists.

Table 9: Route map attributes supporting DRC and Idempotency

Field	Identify Drift	Reconcile configuration	Idempotency	Notes
Route map deleted.	Yes	Yes	No	Recreate the route map, along with the match criteria during reconcile.
Route map rule action updated.	Yes	Yes	No	Reconcile the route map action (permit/ deny) for that rule.
Update IPv4 prefix list name in match criteria.	Yes	Yes	No	Reconcile the IPv4 prefix list name.
IPv4 prefix list match criteria deleted.	Yes	Yes	NA	Reconcile the match criteria for IPv4 prefix list.
A different match criteria NOT supported by EFA is added through OOB.	No	No	NA	
A set criteria NOT supported by EFA is added through OOB.	No	No	NA	
Route map is created through OOB and this is not present or created by EFA.	No	No	NA	

The following CLI commands are available for operations on route maps:

- **`efa policy route-map create`**
- **`efa policy route-map update`**
- **`efa policy route-map delete`**
- **`efa policy route-map list`**
- **`efa policy route-map-match create`**
- **`efa policy route-map-match list`**
- **`efa policy route-map-match delete`**

For more information, see the *Extreme Fabric Automation Command Reference, 2.7.0*.

Device Settings

The following table captures the various attributes of device settings for which DRC and idempotency are supported.

Table 10: Device settings attributes supporting DRC and Idempotency

Field	Identify Drift	Reconcile Configuration	Idempotency
BGP prefix independent convergence (PIC)	Yes	Yes	Yes
ECMP routed load-sharing max path	Yes	Yes	Yes

Note
Drift and reconcile and idempotency configuration support is not applicable for device update and viewing device settings.

BGP Prefix Independent Convergence (PIC)

Specify **`Yes`** to enable BGP PIC and **`No`** to de-configure it.

The following CLI command is available to enable BGP PIC:

- `efa inventory device setting update --prefix-independent-convergence`

Note
After configuring this command, clear the routes on the device.

For more information, see *Extreme Fabric Automation Command Reference, 2.7.0*.

ECMP Max Path

Use the command **`string`** to view route load-sharing maximum paths. Valid values include **`8`**, **`16`**, **`32`**, and **`64`**. The default value is **`64`**.

Note
The device must be reloaded for this command to take effect.

The following CLI command is available to configure ECMP route load-sharing max path:

- `efa inventory device setting update --maximum-load-sharing-paths`

For more information, see *Extreme Fabric Automation Command Reference, 2.7.0*.

Device Settings Update

Configure the maintenance mode and display the available device settings.

Use the **`device-ips`** parameter, separated by comma to view a range of device IP addresses.

Use the **`fabric-name`** parameter to specify the name of the fabric.

The **`show`** command displays the device settings.

The following CLI command is used to display the device settings:

- `efa inventory device setting show [ --ip device-ips ]`

For more information, see *Extreme Fabric Automation Command Reference, 2.7.0*.

Time Zone Configuration

By default, SLX devices come up with the GMT timezone. Using the `efa inventory device timezone` command, you can set the timezone per device or per fabric.

The following CLI commands are available for timezone settings:

- **`efa inventory device timezone set`**
- **`efa inventory device timezone unset`**
- **`efa inventory device timezone list`**

Table 11: Time zone attributes supporting DRC and Idempotency

Field	Identify Drift	Reconcile configuration	Idempotency
Time zone is set.	Yes	Yes	Yes
Time zone is unset.	Not applicable	Not applicable	Yes

Note
Identify drift, drift and reconcile, and idempotency support is not applicable for time zone display.

For more information, see the *Extreme Fabric Automation Command Reference, 2.7.0*.

Idempotent Operations

The idempotent operations produce the same result for multiple identical requests or operations.

Reissuing an EFA command should leave the system in the same state as the last time the command was run. Such idempotent operations help ensure data consistency during high-availability failovers.

In this example, running the **`efa fabric create`** command twice, with the same parameters, produces the same result each time.

```
$ efa fabric create --name fabric1  --type non-clos --description non-clos-fabric
Create Fabric nonclos [Success]

(efa:extreme)extreme@tpvm:~$ efa fabric create --name fabric1 --type non-clos --
description non-clos-fabric
Create Fabric nonclos [Success]
```

Rollback Scenarios for Data Consistency

Rollback of failed configuration changes ensures data consistency.

Failure on Some Devices during Configuration

When a REST operation succeeds on one device but fails on another, configuration changes are rolled back for both devices. In the following example, the operation fails on one MCT node but succeeds on the other. The whole operation fails and an error message is returned as part of the REST response.

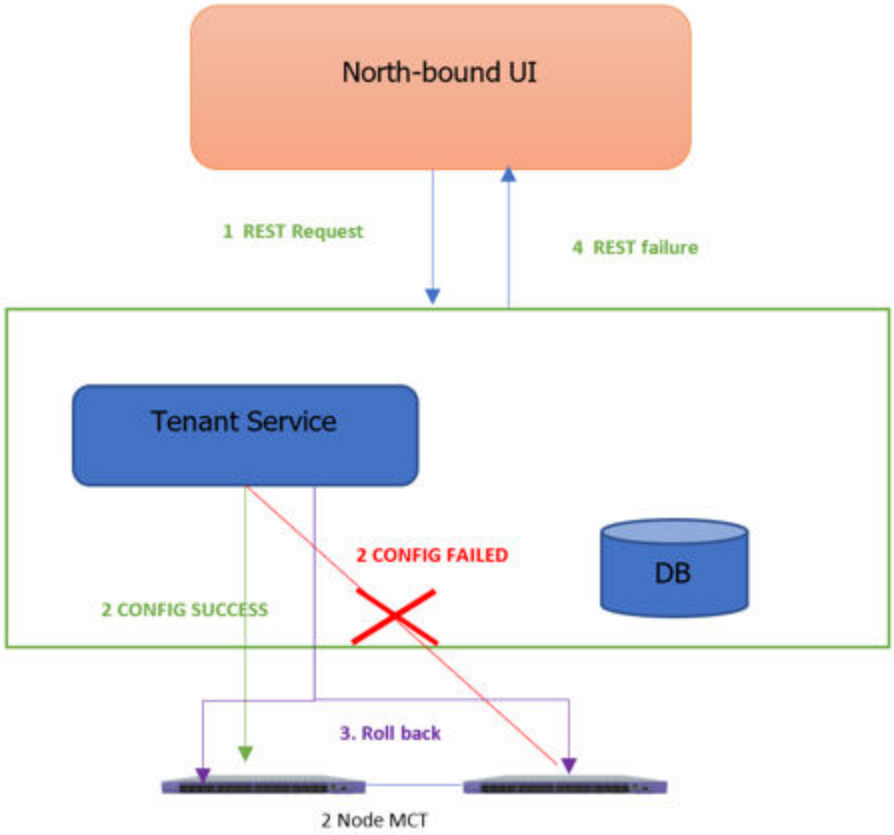


Figure 6: Rollback for failure of one node

Note

This process for partial failures is the default. You can change the process to enable partial successes even when one node fails. For more information, see Administered Partial Success on page 191.

Failure on All Devices during Configuration

When a REST operation fails on all devices in the request, configuration changes are rolled back for all devices. In this example, the operation fails on both MCT nodes and an error message is returned as part of the REST response.

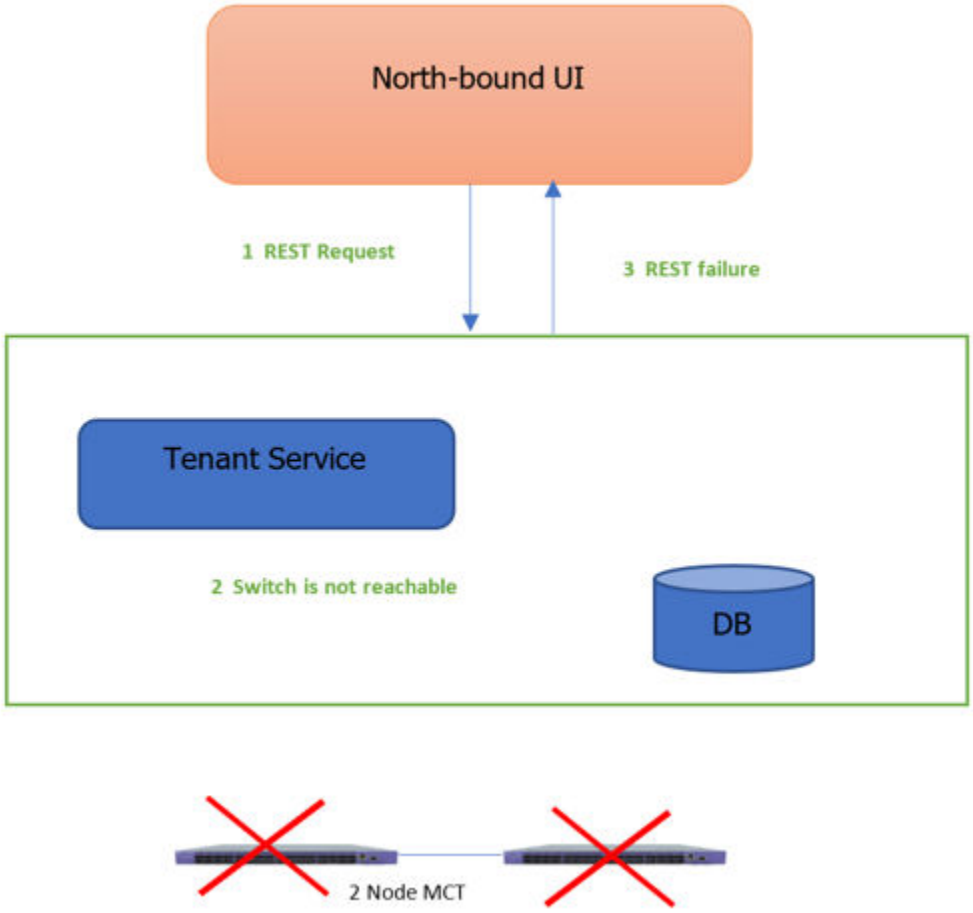


Figure 7: Rollback for failure of both nodes

Failure during De-configuration

Rollback does not occur when a REST operation fails during a de-configuration request. The status of configuration items that were not rolled back changes to "delete-pending." You must manually verify and address the status of such items.

EFA High Availability Failover Scenarios

EFA high availability provides for uninterrupted service in several different scenarios.

For information about deploying EFA for high availability, see the *Extreme Fabric Automation Deployment Guide, 2.7.0*.

SLX Device Failure

When an SLX device fails, the SLX-OS and the EFA services running on TPVM go down for the failed node. The time it takes for failover to the standby node varies depending on whether the K3s agent node is actively running the EFA services. The following image depicts a scenario in which one SLX device fails.

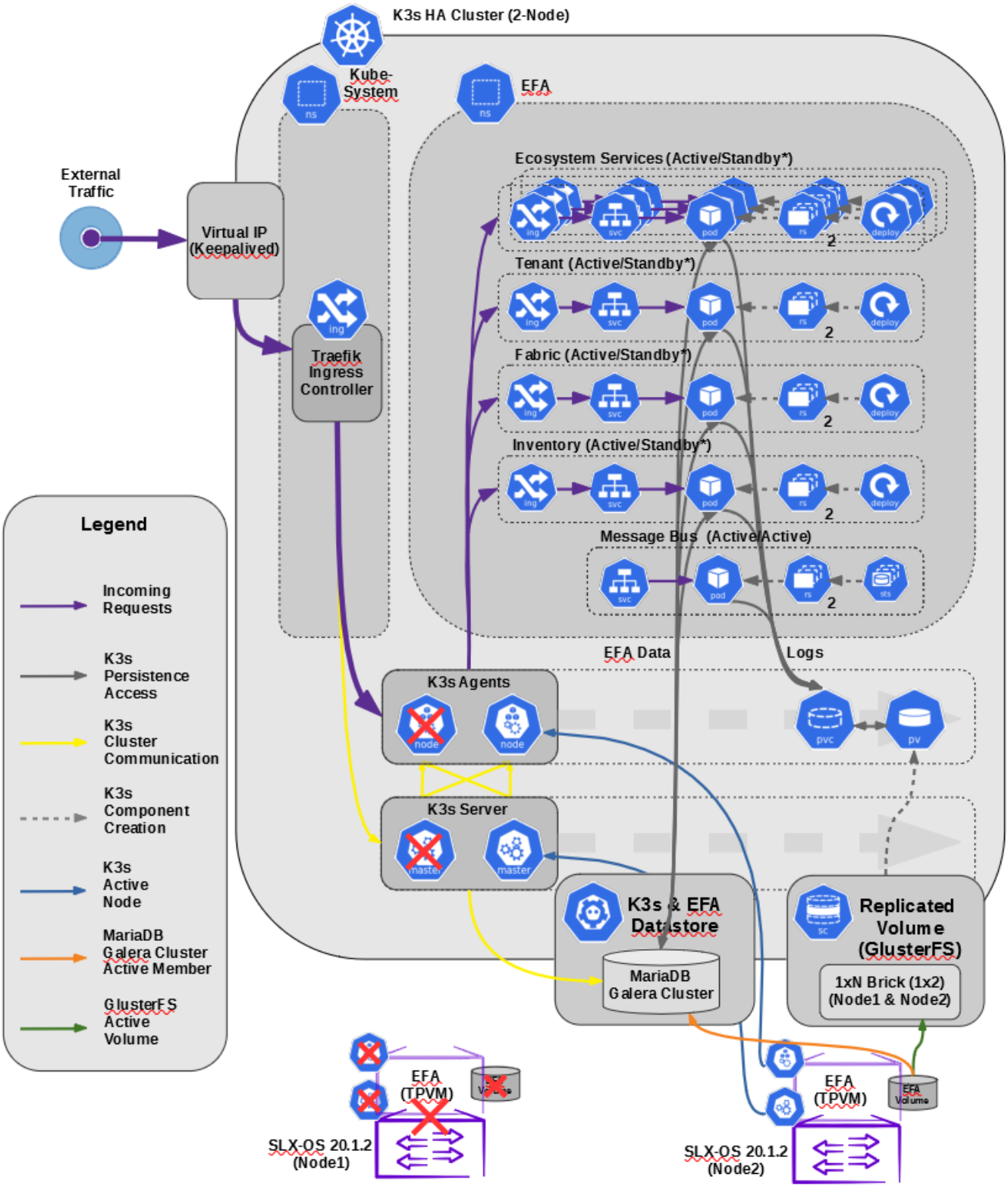


Figure 8: SLX device failure in a two-node cluster

SLX Device Failure on the Active K3s Agent Node

When the K3s agent node is actively running EFA services on a node that fails, K3s initiates failover and starts the EFA services on the standby node. Failover is complete when EFA services are running on the newly active K3s agent node (node 2).

Because the GlusterFS replicated volume remains available during failover, the K3s cluster data store and the EFA data store remain operational.

When the failed node is again operational, it becomes the standby node. The K3s agent node continues to run EFA services from node 2. When both nodes are up and K3s is running, all services fetch the latest data from devices to ensure that EFA has the latest configurations.

SLX Device Failure on the Standby K3s Agent Node

When the K3s agent node is the standby and is not running EFA services, no failover actions occur if this node fails. EFA services continue to run on the active node without interruption.

TPVM Failure

The TPVM failure scenario is similar to that of the SLX device failure scenario. The only difference is that SLX-OS continues to operate.

Two-node Failure

In the unlikely event that both nodes in the cluster fail at the same time (for reasons such as a power failure or the simultaneous reboot of SLX devices), EFA has built-in recovery functionality. If the cluster is not automatically recovered within 10 minutes of power being restored or within 10 minutes of the TPVM being rebooted, then you can manually recover the cluster.

Multiple Management IP Networks

EFA is supported on multiple management IP networks and routes, providing access to EFA from servers that are on different subnets.

Overview

The Multiple Management IP (MMIP) Networks feature offers the following support:

- Supports single node and multi-node deployments
- Supports TPVM deployments, server-based deployments, and VM-based deployments
- Supports the configuration of additional management IP networks and routes during EFA installation
- Supports adding and viewing management networks and routes after EFA installation
- Supports deleting management networks and routes after EFA installation
- Supports the migration of the multiple network configuration during the following EFA upgrade scenarios: single node to multi-node and multi-node to multi-node

- Supports up to 6 networks
- Supports the RMA, backup, restore, and upgrade functions

Note
If you do not need multiple management networks, simply reply "no" when prompted during EFA installation or upgrade. For instructions, see the installation and upgrade topics in the *Extreme Fabric Automation Deployment Guide, 2.7.0*.

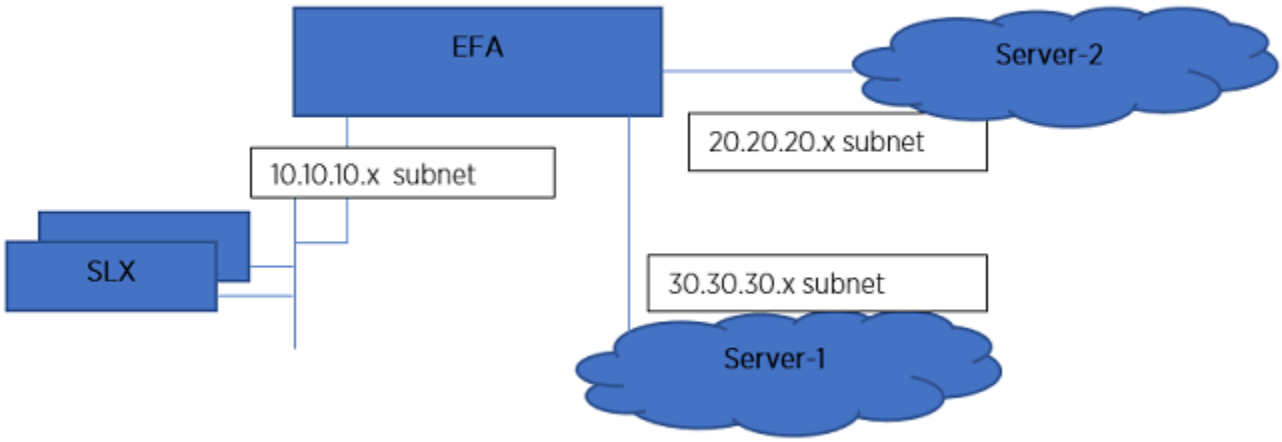


Figure 9: MMIP architecture

Assumptions

- In a multi-node deployment, the sub-interface with the VLAN is created under the same NIC as the VIP destination. In a single-node deployment, the sub-interface is created under the NIC that you specified as the host IP installation (if there are multiple NICs). Creating sub-interfaces on different NICs of the server is not supported.
- EFA does not validate connectivity to the newer IP subnets. You are responsible for ensuring reachability.
- Changing IP subnets or IP routes is not supported. To make changes to a management network or network route, you must delete the network or route and then create a new one.
- You can expect about 20 to 30 seconds of downtime when adding or deleting management networks.
- In a multi-node deployment, both nodes have to be up and available during add and delete operations (because sub-interface creation and keepalived changes are unique to the node). Because these are infrequent operations, you should verify that both the nodes are up and in READY state before beginning add or delete operations.

Add and Delete Management Routes

In a multi-node deployment, you can add, delete, and show management routes for Multiple Management IP (MMIP) networks.

Typically, the create and delete operations do not cause a high-availability failover. The route is instantiated on the active node of the cluster. If failover does occur from node 1 to node 2, then keepalived ensures that the route transitions from node 1 to node 2.

1. To add a management route, run the following command.

   ```
   $ efa mgmt route create --src <mmip-vip> --to <dest-cidr> --via <next-hop-ip>
   ```

   If a route with the same destination exists, the operation fails. This operation updates the keepalived configuration file on both nodes of the high-availability cluster.
2. To delete a management route, run the following command.

   ```
   $ efa mgmt route delete --src <mmip-vip> --to <dest-cidr> --via <next-hop-ip>
   ```

   If a route matching the three parameters does not existing, the operation fails. If a matching route is found, the keepalived configuration file is updated and reloaded on both nodes of the high-availability cluster.
3. To generate a list of all management routes, run the following command.

   ```
   $ efa mgmt route show
   +-------------+------------------+-------------+
   |  Route-Src  |     Route-To     |  Route-Via  |
   +-------------+------------------+-------------+
   | 10.21.30.40 | 192.168.100.0/24 | 10.21.30.41 |
   +-------------+------------------+-------------+
   ```

Configuration Supporting Multiple Management IP Networks

This work flow highlights the changes that occur in your system when you configure Multiple Management IP (MMIP) networks.

Day 0 and Installation Configuration

- In a multi-node deployment, the VIP (virtual IP address) that you enter during installation is the same as for a non-MMIP deployment. This VIP is distinguished from those added during MMIP network operations and cannot be deleted.
- During installation, you are prompted to create additional MMIP networks and routes.
- Keepalived, ingress, and interface changes are performed on both nodes of a multi-node deployment.
- Configuration is persisted for RMA purposes, so that the Supportsave function has data for debugging issues.

For step-by-step instructions for configuring MMIP during installation or upgrade, see the *Extreme Fabric Automation Deployment Guide, 2.7.0*.

Day 1 to Day *n* Configuration

- You can use the EFA CLI or REST APIs to add and delete management routes and IP address and VLAN combinations.
- Keepalived, ingress, and interface changes are performed on both nodes of a multi-node deployment.

- Configuration is persisted for RMA purposes, so that the Supportsave function has data for debugging issues.
- The backup and restore process also restores the previous configuration of the sub-interfaces.

Add and Delete Management Subinterfaces

Use the EFA CLI to add and delete management subinterfaces.

1. Run the following command to add a management subinterface:

   ```
   efa mgmt subinterface create --name <name> --ip-address <ip-subnet>
   --vlan-id <VLAN>
   ```

   - If a management network with the same name exists, the operation fails.
   - The changes made by this operation span three different components:
     - Subinterface creation under the physical NIC
     - Keepalived configuration changes (for high-availability deployments)
     - Ingress controller changes
   - If any operation to the component fails, it is marked as a failed operation and the configurations return to the previous state.
2. Run the following commands to delete a management subinterface:

   ```
   efa mgmt subinterface delete --name <name>
   ```

   If a management network with the name exists, it is deleted. Otherwise, the correct response is provided in the command output.

```
$ efa mgmt subinterface?
Management subinterface commands

Usage:
  efa mgmt subinterface [command]

Available Commands:
  create        Create sub-interface (sub-interface)
  delete        Delete sub-interface (sub-interface)
  show          List of sub-interfaces (sub-interfaces)

Flags:
  -h, --help          help for sub-interface

Use "efa mgmt subinterface [command] --help" for more information about a command.
$ efa mgmt subinterface create -h
Create management subinterface (sub-interface)
Usage:
  efa mgmt subinterface create [flags]
Flags:
      --name string         Name of the sub-interface
      --vlan-id int         VLAN Id of sub-interface
      --ip-address string   IP Address of sub-interface including subnet mask.
                            Example: 10.24.80.150/24
  -h, --help                help for create
$ efa subinterface delete -h
Delete management subinterface (sub-interface)
Usage:
  efa mgmt subinterface delete [flags]
Flags:
```

```
      --name string   Name of the sub-interface
  -h, --help          help for delete

$ efa mgmt subinterface show -h
List of management sub-interfaces (sub-interfaces)

Usage:
  efa mgmt subinterface show [flags]
 Flags:
       --name string   Name of the sub-interface
   -h, --help          help for show

$ efa mgmt subinterface create --name server1 --vlan-id 20 --ip-address
          20.20.20.2/24
Subinterface server1 created successfully

$ efa mgmt subinterface delete --name server1
Subinterface server1 deleted successfully

$ efa mgmt subinterface show
+---------------+------------------+------+---------------+
| Sub-Interface | Parent Interface | Vlan |   IP Subnet   |
+---------------+------------------+------+---------------+
| server1       | eth0             | 20   | 20.20.20.2/24 |
+---------------+------------------+------+---------------+
| server2       | eth0             | 50   | 50.50.50.2/24 |
+---------------+------------------+------+---------------+
Management Subinterfaces Details

$ efa mgmt subinterface show --name server1
+---------------+------------------+------+---------------+
| Sub-Interface | Parent Interface | Vlan |   IP Subnet   |
+---------------+------------------+------+---------------+
| server1       | eth0             | 20   | 20.20.20.2/24 |
+---------------+------------------+------+---------------+
Management Subinterface Details
```

Configure Static IP Addresses for Management Subinterfaces

Use the EFA CLI to add, delete, and show the static IP addresses for management subinterfaces.

1. To add static IP addresses to a specified subinterface, run the following command.

   ```
   efa mgmt subinterface staticip add
   --subinterface <int-id> --ip1 <ip-addr/mask> --ip2 <ip-addr/mask>
   ```

2. To remove static IP addresses from a specified subinterface, run the following command.

   ```
   efa mgmt subinterface staticip remove
   --subinterface <int-id>
   ```

3. To show all subinterfaces and the IP addresses that are attached to them, run the following command.

   ```
   efa mgmt subinterface staticip show
   ```

Change the Default Gateway of a TPVM

Use this procedure to change the default gateway of a TPVM. This procedure has no impact on the functioning of EFA high-availability mode.

- The gateway IP address must be in the same subnet as one of the subinterfaces that are created in EFA.
- Perform this procedure on both nodes where EFA is deployed, to avoid a loss of EFA functionality.

- Perform this procedure with caution. EFA and SLX-OS do not validate reachability of the gateway during this operation.
- Vital services, such as DNS, NTP, and LDAP, must be reachable from the new gateway.

1. Add static IP addresses to the subinterface.

   You can assign a maximum of one pair of static IP addresses. Only one subinterface at a time can have static IP addresses.

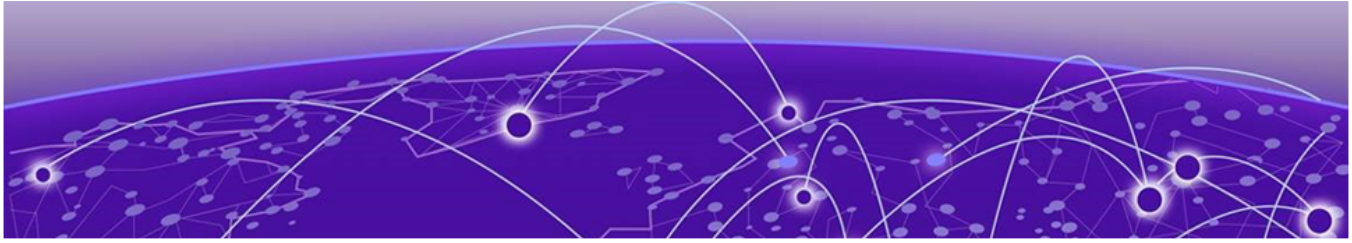
   ```
   efa mgmt subinterface staticip --name <sub-int-name> --ip1 <ip-addr>
   --ip2 <ip-addr>
   ```

2. Change the gateway of the standby TPVM.

   ```
   efa inventory device execute-cli --ip <standby-slx-ipaddr>
   --command "tpvm TPVM,interface management ip <standby-tpvm-ipaddr>
   gw <new-gateway-ipaddr>" --config
   ```

3. Change the gateway of the active TPVM.

   ```
   efa inventory device execute-cli --ip <active-slx-ipaddr>
   --command "tpvm TPVM,interface management ip <active-tpvm-ipaddr>
   gw <new-gateway-ipaddr>" --config
   ```



Fabric Infrastructure Provisioning

Fabric Service Overview on page 59
IP Fabric and Clos Orchestration Overview on page 60
SLX Device Prerequisites for Fabric Service on page 60
Clos Overview on page 61
Configure a 3-Stage Clos Fabric on page 62
Configure a 5-Stage Clos Fabric on page 64
Overview of Day-0 Operations for a Small Data Center Fabric on page 66
Supported Small Data Center Topologies on page 66
Configure a Small Data Center Fabric on page 68
IP Multicast Fabric Provisioning on page 69
View Fabric Details on page 76
Edit Fabric Settings on page 77
Fabric Event Handling on page 80
Northbound IPv6 support on page 81

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 (small data center support)
- Support for 3- and 5-stage Clos fabrics
- Support for MCT configuration
- Support for Eco-System Integration; Openstack, VMWare vCenter, Microsoft Hyper-V, and 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.

  For more information about the commands, see the *Extreme Fabric Automation Command Reference, 2.7.0*

  Reference, 2.5.0
- 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 you configure your fabric.

- Management IP addresses must be configured on all devices.
- SLX devices must have the appropriate firmware version. For more information, see the list of supported platforms in the *Extreme Fabric Automation Deployment Guide, 2.7.0*.
- SLX 9850: Fabric links must be enabled manually, through `no shut`.
- SLX 9540: The appropriate TCAM profile must be set and the device 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)#
```

- Refer to the release-specific *Extreme SLX-OS Management Configuration Guide* for configuration steps for each platform.

Clos Overview

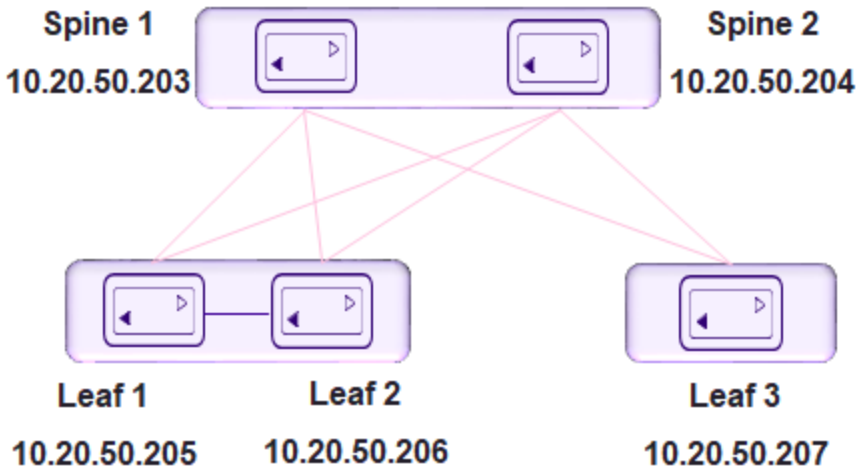
Extreme Fabric Automation (EFA) manages the life cycle of IP Fabric Clos networks.

EFA offers unique flexibility in supporting 3- and 5-stage Fabric Clos topologies based on a BGP underlay with a BGP or EVPN overlay.

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

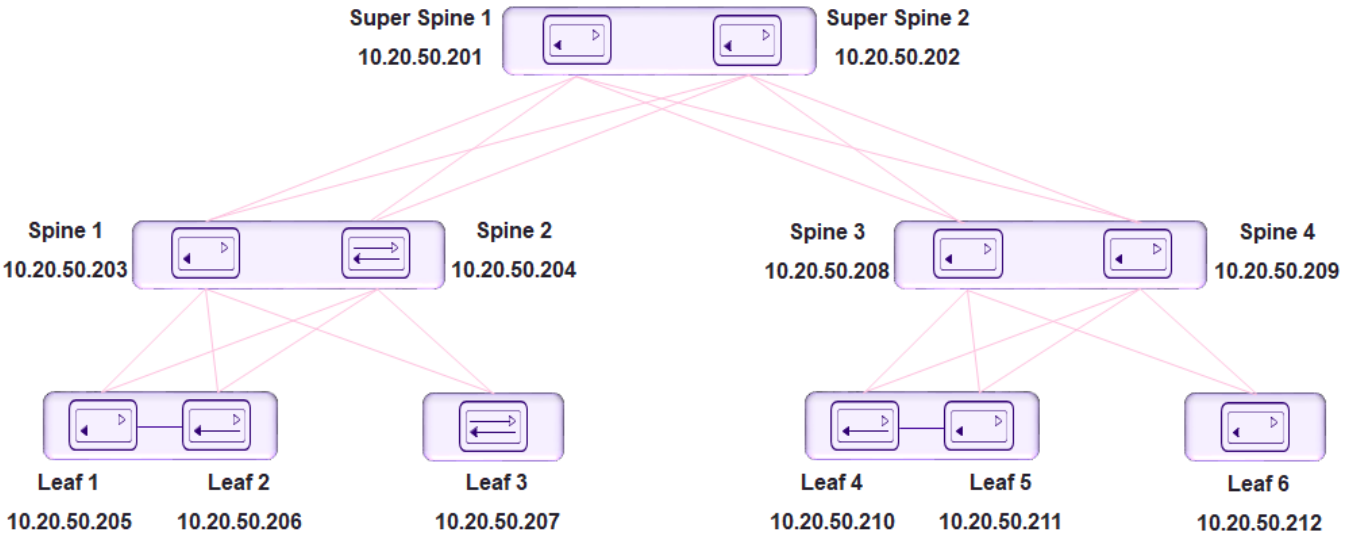
3-Stage Clos

3-stage Clos consists of an ingress leaf layer, a middle spine layer, and an egress leaf layer. Servers are connected to leaf devices and leaf devices are connected to all spines. No leaf devices are connected to other leaf devices, nor are spines connected to spines. Data enters at an ingress leaf, is routed through a spine to an egress leaf, and then out of the network to the next server in the path. In this topology, servers are always 3 hops (leaf, spine, leaf) away from another server.



5-Stage Clos

5-stage Clos is a 3-stage topology that is divided into clusters and on which a Super-spine layer is added. All links between leaf and spine must be connected. Spine are not be interconnected. Similarly, all the links between the spine and Super-spine must be connected.



Configure a 3-Stage Clos Fabric

The 3-stage topology has 2 layers of devices: leaf and spine. All links between leaf and spine must be connected. Spine nodes are not interconnected.

Tip
If any devices in a fabric are in "admin-down" state, use of the following commands in that same fabric will not add or delete devices in the fabric: **`efa fabric device add-bulk`** and **`efa fabric device remove`**.

1. Create the fabric.

```
efa fabric create
```

2. Add a device to the fabric.

```
efa fabric device add
```

A device must be registered with the Inventory Service before you can add it to a fabric. However, if you provide a user name and password when you run the command, then the devices are automatically registered with the Inventory Service. See the examples at the end of this procedure.

You can add multiple devices by using the **efa fabric device add-bulk** command.

Tip

To validate fabric port-link status, complete the following operations before running the **efa fabric device add-bulk** command:

a. Run the **efa inventory device register –ip <list of device-ips>** command.
b. Run the **efa inventory device interface list –ip <device-ip>** command.
    i. Verify port link status (up or down) in Admin Status and Oper Status fields.
    ii. Confirm they are as expected.
    iii. If not, manually check for physical cabling and fix any issues. Continue with the `efa fabric device add-bulk` operation.

3. Configure the fabric.

```
efa fabric configure
```

Topology validation occurs during the addition of a device and during fabric configuration. The following 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 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.
- A leaf node marked as "multi-homed" must have an MCT neighbor.
- A leaf node marked as "single-homed" is not connected to other leaf nodes.
- A border beaf node marked as "multi-homed" must have an MCT neighbor.
- A border leaf node marked as "single-homed" 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).

Tip

The validation process reports any errors as a response to the **efa fabric device add** or **efa fabric configure** operations. You can use the **efa fabric error show** command to export these errors to a CSV file.

Note
You cannot change fabric settings after you add devices to the fabric, with the following exceptions: `--md5-password-enable`, `--md5-password`, `--bgp-dynamic-peer-listen-limit`, and `--single-rack-deployment settings`.

This example creates the fabric.

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

This example adds multiple devices 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
```

This example configures the fabric.

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

Configure a 5-Stage Clos Fabric

The 5-stage topology has three layers of devices: leaf, spine, and super-spine.

You can build a 5-stage Clos from top to bottom or bottom to top. The following example builds from top to bottom.

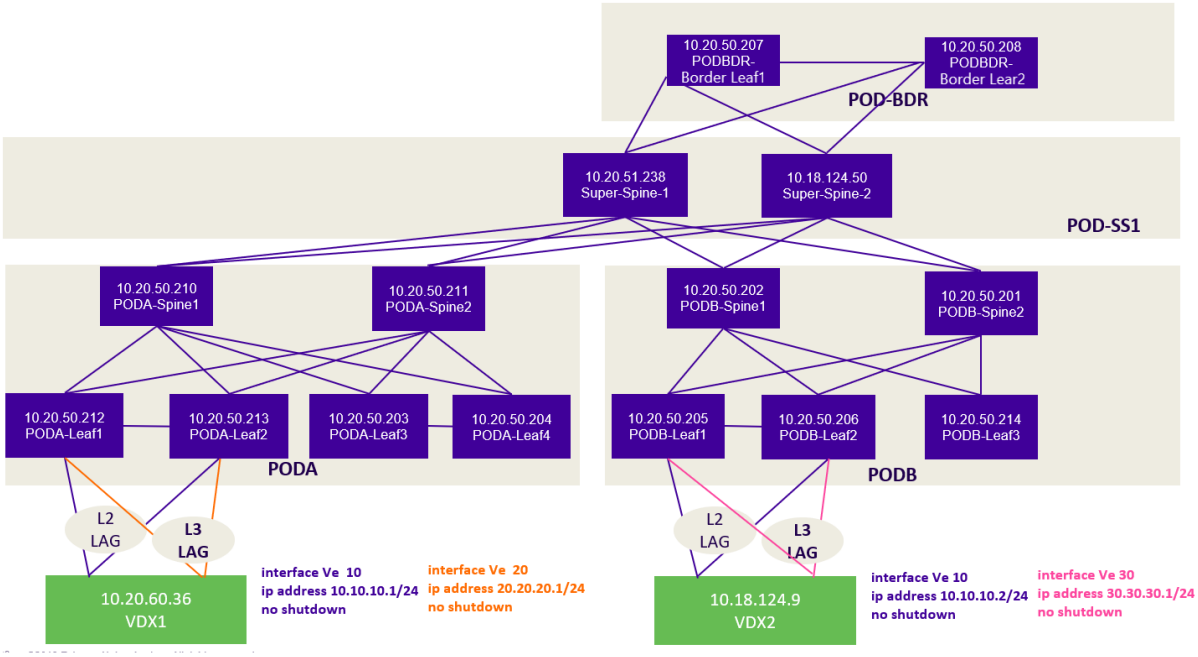


Figure 10: 5-Stage Clos fabric topology

Tip
If any devices in a fabric are in "admin-down" state, use of the following commands in that same fabric will not add or delete devices in the fabric: **`efa fabric device add-bulk`** and **`efa fabric device remove`**.

1. Create the fabric.

```
efa fabric create
```

2. Add a device to the fabric.

```
efa fabric device add
```

A device must be registered with the Inventory Service before you can add it to a fabric. However, if you provide a user name and password when you run the command, then the devices are automatically registered with the Inventory Service. See the examples at the end of this procedure.

You can add multiple devices by using the **efa fabric device add-bulk** command. If you choose to add multiple devices in bulk, ensure you perform the following operations first:

- Run the **efa inventory device register --ip <list-of-device-ips>** command.
- Run the **efa inventory device interface list --ip <device-ip>** command. In the output of the command, verify that the states of the port links are as you expected (in the Admin Status and Oper Status fields). If not, manually check the physical cabling and fix any issues. Then continue with the **efa fabric device add-bulk** operation.

3. Configure the fabric.

```
efa fabric configure
```

Topology validation occurs during the addition of a device and during fabric configuration. The following 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.
- A leaf node marked as "multi-homed" must have an MCT neighbor.
- A leaf node marked as "single-homed" is not connected to other leaf nodes.
- A border leaf node marked as "multi-homed" must have an MCT neighbor.
- A border leaf node marked as "single-homed" 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).

Tip

The validation process reports any errors as a response to the **efa fabric device add** or **efa fabric configure** operations. You can use the **efa fabric error show** command to export these errors to a CSV file.

This example creates the fabric.

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

This example adds a device to the 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
```

This example adds multiple devices to the 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
```

This example configures the fabric topology.

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

Overview of Day-0 Operations for a Small Data Center Fabric

Day-0 operations consist of forming the fabric.

This table provides examples of the commands that you use to create a small data center (non-Clos) fabric with two SLX devices. For more information about commands and supported parameters, see *Extreme Fabric Automation Command Reference, 2.7.0*.

Table 12: Day-0 operations

Operation	Command Example
Create a fabric	**efa fabric create --name CNCF type non-clos**
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

Supported Small Data Center Topologies

EFA supports small data center (non-Clos) fabrics.

EFA provides the following support for small data center fabrics on, SLX 9150, SLX 9250, and SLX 9740 devices:

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

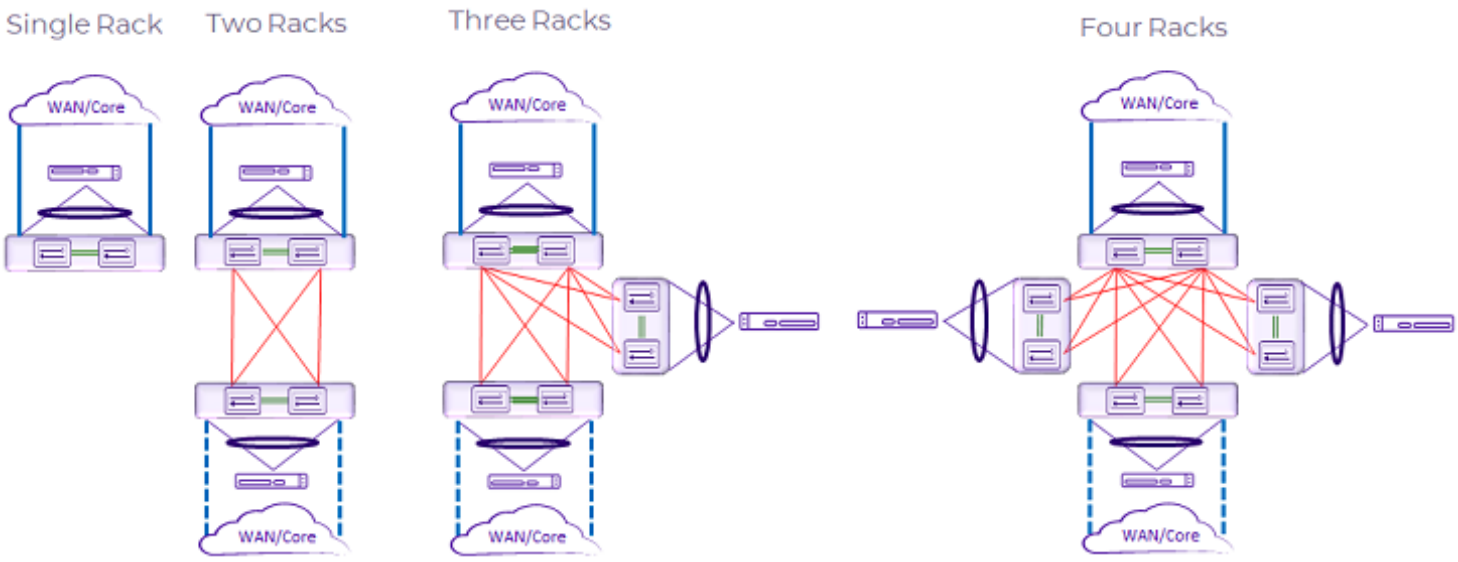


Figure 11: Supported small data center topologies

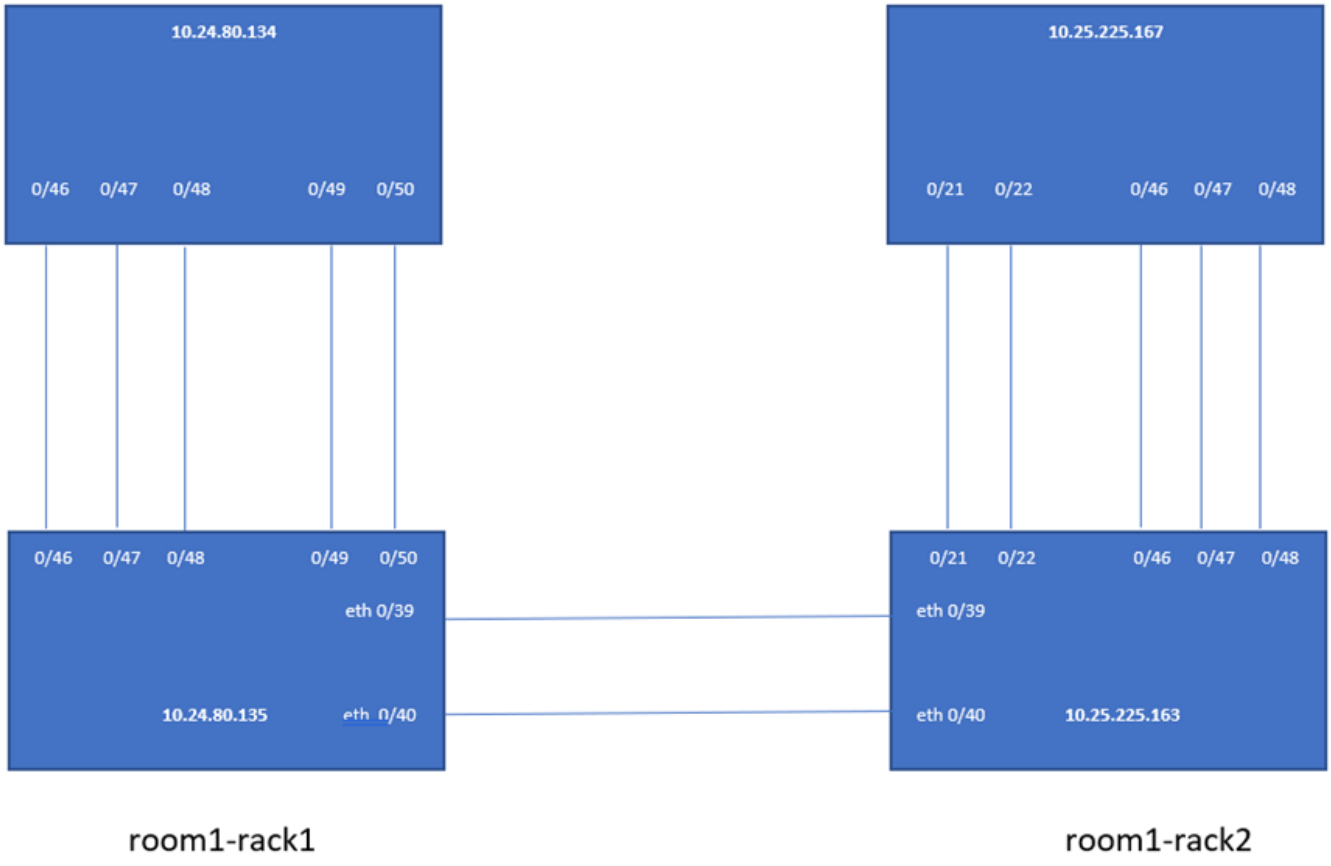


Figure 12: Multi-rack configuration example

Configure a Small Data Center Fabric

A small data center (non-Clos) fabric is used in small data centers.

Tip
If any devices in a fabric are in "admin-down" state, use of the following commands in that same fabric will not add or delete devices in the fabric: **efa fabric device add-bulk** and **efa fabric device remove**.

1. Create the fabric.

   ```
   efa fabric create
   ```

2. Add a device to the fabric.

   ```
   efa fabric device add
   ```

   A device must be registered with Inventory Service before you can add it to a fabric. However, if you provide a user name and password when you run the command, then the devices are automatically registered with the Inventory Service. See the examples at the end of this procedure.

   You can add multiple devices by using the **efa fabric device add-bulk** command. If you choose to add multiple devices in bulk, ensure you perform the following operations first:

   - Run the **efa inventory device register --ip <list-of-device-ips>** command.
   - Run the **efa inventory device interface list --ip <device-ip>** command. In the output of the command, verify that the states of the port links are as you expected (in the Admin Status and Oper Status fields). If not, manually check the physical cabling and fix any issues. Then continue with the **efa fabric device add-bulk** operation.

3. Configure the fabric.

   ```
   $ efa fabric configure
   ```

Tip
The validation process reports any errors as a response to the **efa fabric device add** or **efa fabric configure** operations. You can use the **efa fabric error show** command to export these errors to a CSV file.

This example creates the fabric.

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

This example adds a device to the fabric.

```
efa fabric device add --name extr-fabric --ip 10.x.x.x --rack room1-rack1
--username admin --password password
```

This example adds multiple devices to the fabric.

```
$ 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
```

This example configures the fabric.

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

Dynamic ICL in small data center

Dynamic Inter-Chassis Link (ICL) in small data center (non-Clos) fabric dynamically identifies the ICL links for all the racks. In the latest version, you cannot specify the ICL ports manually.

Dynamic ICL in the non-Clos (small data center) fabric contains the following configuration changes:

1. Fabric settings CLI does not contain MCT ports and L3 backup port options.
2. There is no distinction between small data center and Clos MCT ports or LD-MCT ports in the backend. They do not have static configuration options and are identified by LLDP.
3. EFA running config does not have any references to the MCT ports.
4. Dynamic ICL configuration maintains backward compatibility for upgrade and downgrade operations for EFA.

Note

For small data center fabrics in EFA 2.5.5, MCT ICL ports are defined in the fabric settings; `rack-mct-ports` or `rack-ld-mct-ports`, and are the only ports used to form an ICL. Starting in EFA 2.6.0, all the LLDP enabled ports that interconnect the MCT nodes are used for MCT ICL

Therefore, any ports, which were not previously defined in the fabric settings; `rack-mct-ports` or `rack-ld-mct-ports`, but are interconnecting the MCT nodes, will be automatically used for the MCT ICL after the upgrade from EFA 2.5.5 to EFA 2.6.0 and above.

IP Multicast Fabric Provisioning

IP Multicast Fabric Overview

When multicast traffic is sent over unicast tunnels, ingress replication is done for each remote VTEP node. IP multicast fabric enables IP fabric to distribute BUM (Broadcast, Unknown Unicast, and Multicast Overlay) traffic using multicast VxLAN tunnels established over underlay fabric links.

Multicast Vxlan tunnels use Protocol Independent Multicast - Source Specific Multicast (PIM-SSM) and Multicast Distribution Tree (MDT) to deliver traffic effectively while minimizing packet replication in the fabric.

When multicast fabric is configured, a default MDT is created using PIM-SSM protocol running on fabric links and all the EVPN domain (VLANs and BDs) traffic is routed using the default tree.

The following figures show Clos topology for VxLAN unicast and multicast tunnels.

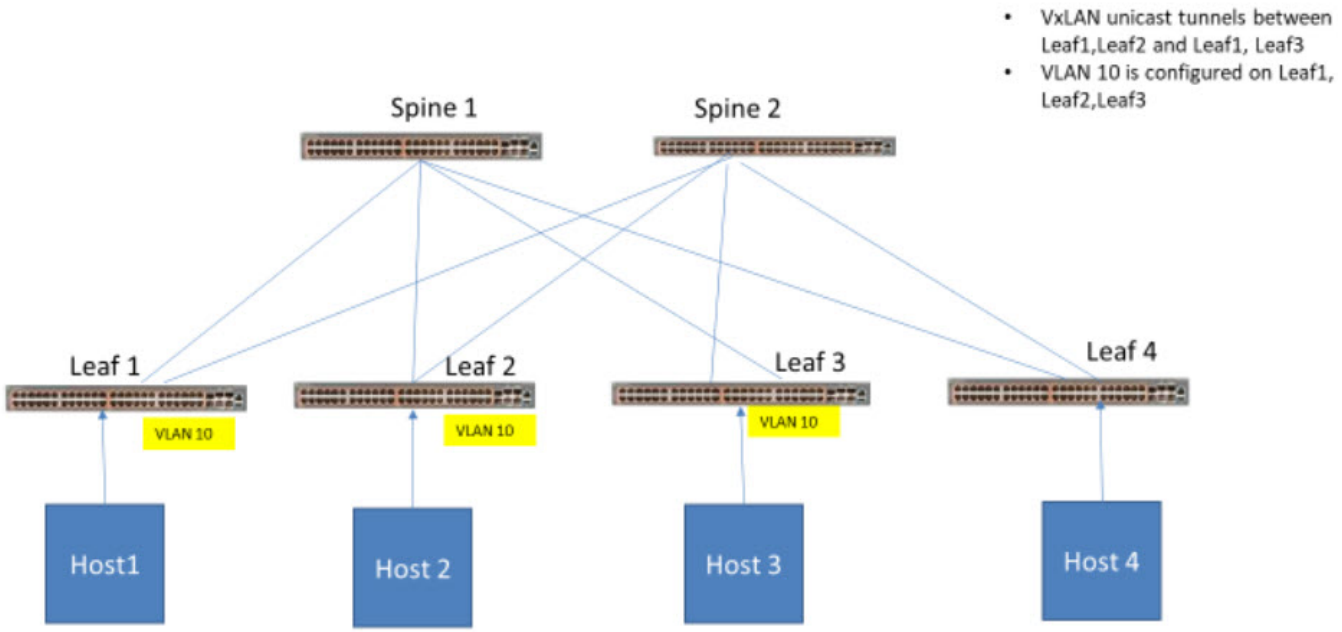


Figure 13: Clos topology with VxLAN unicast tunnels

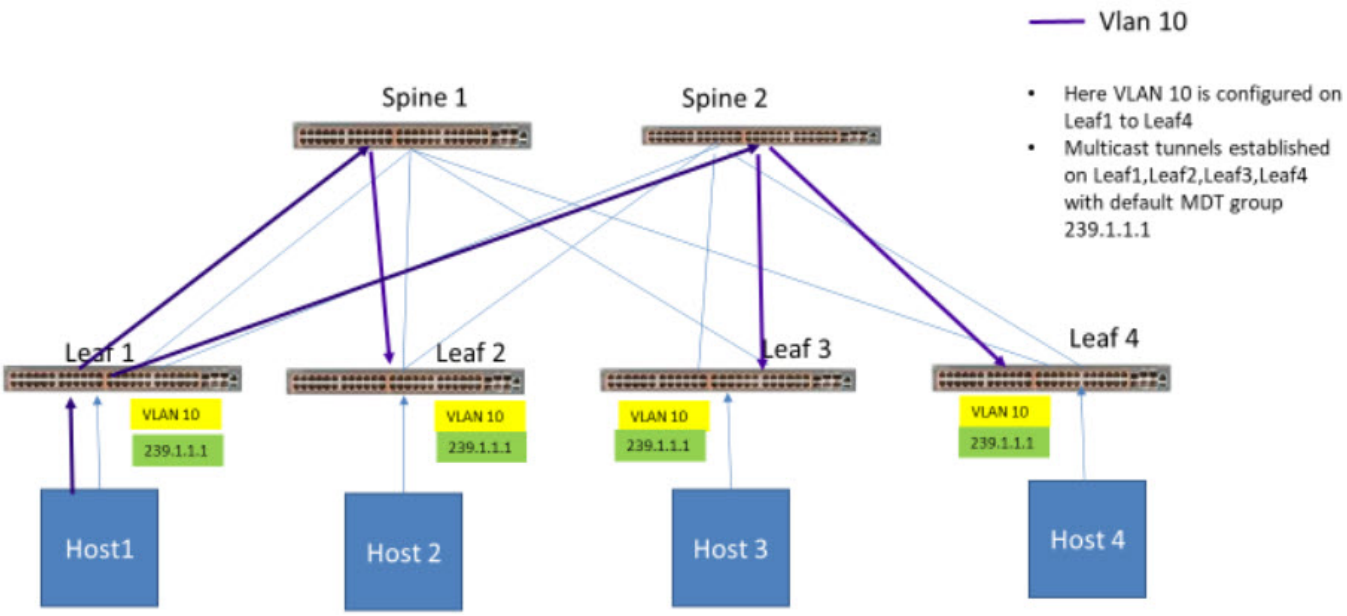


Figure 14: Clos topology with multicast tunnels

Bidirectional Forwarding Detection

Bidirectional Forwarding Detection (BFD) protocol detects faults between two forwarding engines.

When fabric is created, BFD is enabled by default along with fabric links and BGP neighbors. The following example shows BFD configuration settings.

```
# efa fabric setting show --name clos_fabric --advanced
```

```
+--------------------------------+-------------------------+
|              NAME              |          VALUE          |
+--------------------------------+-------------------------+
| Fabric Name                    | default                 |
+--------------------------------+-------------------------+
| Link IP Range                  | 10.10.10.0/23           |
+--------------------------------+-------------------------+
| Loopback IP Range              | 172.31.254.0/24         |
+--------------------------------+-------------------------+
| Loopback Port Number           | 1                       |
+--------------------------------+-------------------------+
| VTEP Loopback Port Number      | 2                       |
+--------------------------------+-------------------------+
| Spine ASN Block                | 64512-64768             |
+--------------------------------+-------------------------+
| SuperSpine ASN Block           | 64769                   |
+--------------------------------+-------------------------+
| Leaf ASN Block                 | 65000-65534             |
+--------------------------------+-------------------------+
| Border Leaf ASN Block          | 66000-66100             |
+--------------------------------+-------------------------+
| P2P IP Type                    | numbered                |
+--------------------------------+-------------------------+
| Any cast MAC                   | 0201.0101.0101          |
+--------------------------------+-------------------------+
| IPV6 Any cast MAC              | 0201.0101.0102          |
+--------------------------------+-------------------------+
| MAC Aging Timeout              | 1800                    |
+--------------------------------+-------------------------+
| MAC Aging Conversational       | 300                     |
| Timeout                        |                         |
+--------------------------------+-------------------------+
| MAC Move Limit                 | 20                      |
+--------------------------------+-------------------------+
| Duplicate MAC Timer            | 5                       |
+--------------------------------+-------------------------+
| Duplicate MAC Timer MAX Count  | 3                       |
+--------------------------------+-------------------------+
| BFD Enable                     | Yes                     |
+--------------------------------+-------------------------+
| BFD Tx                         | 300                     |
+--------------------------------+-------------------------+
| BFD Rx                         | 300                     |
+--------------------------------+-------------------------+
| BFD Multiplier                 | 3                       |
+--------------------------------+-------------------------+
| BGP MultiHop                   | 2                       |
+--------------------------------+-------------------------+
| MaxPaths                       | 8                       |
+--------------------------------+-------------------------+
| AllowAsIn                      | 0                       |
+--------------------------------+-------------------------+
| MTU                            | 9216                    |
+--------------------------------+-------------------------+
| IPMTU                          | 9100                    |
+--------------------------------+-------------------------+
| MCT Link IP Range              | 10.20.20.0/24           |
+--------------------------------+-------------------------+
| MCT PortChannel                | 64                      |
+--------------------------------+-------------------------+
| LACP Timeout                   | long                    |
+--------------------------------+-------------------------+
| Control Vlan                   | 4090                    |
+--------------------------------+-------------------------+
```

```
| Control VE                      | 4090                    |

…Skipped
```

Fabric Settings to Update BGP MD5 Password, BGP Dynamic Peer Listen Limit, and Single Rack Deployment

Use the fabric settings to update BGP MD5 password using `--md5-password` and `--md5-password-enable`. Update the BGP dynamic peer listen limit using `--bgp-dynamic-peer-listen-limit`. A setting to denote a single rack or multi-rack deployment `--single-rack-deployment` is added under fabric setting. You can update these settings after fabric is configured.

For details on BGP MD5 Authentication, see *Extreme Fabric Automation Security Guide, 2.7.0*.

The following is an example of output from the **`Rack Low Density L3 backup`** and **`efa fabric setting show --name --advanced`** command.

```
| Rack Low Density L3 backup     | 0/30                        |
| port (not applicable to        |                             |
| SLX-9250)                      |                             |
+--------------------------------+-----------------------------+
| Rack Low Density MCT Ports     | 0/19,0/22                   |
+--------------------------------+-----------------------------+
| Configure Overlay Gateway      | Yes                         |
+--------------------------------+-----------------------------+
| VNI Auto Map                   | Yes                         |
+--------------------------------+-----------------------------+
| Backup Routing Enable          | Yes                         |
+--------------------------------+-----------------------------+
| Backup Routing IPv4 Range      | 10.40.40.0/24               |
+--------------------------------+-----------------------------+
| Backup Routing IPv6 Range      | fd40:4040:4040:1::/120      |
+--------------------------------+-----------------------------+
| MD5 Password Enable            | Yes                         |
+--------------------------------+-----------------------------+
| MD5 Password                   | $9$9Yc+EqhoseRMPh9fDhk9TQ== |
+--------------------------------+-----------------------------+
| BGP Dynamic Peer Listen Limit  | 101                         |
+--------------------------------+-----------------------------+
```

```
# efa fabric setting show  --name fabric1 --advanced
+--------------------------------+-----------------------------+
|              NAME              |            VALUE            |
+--------------------------------+-----------------------------+
| Fabric Name                    | fabric1                     |
+--------------------------------+-----------------------------+
| Link IP Range                  | 10.10.10.0/23               |
+--------------------------------+-----------------------------+
| Loopback IP Range              | 172.31.254.0/24             |
+--------------------------------+-----------------------------+
| Loopback Port Number           | 1                           |
+--------------------------------+-----------------------------+
| VTEP Loopback Port Number      | 2                           |
+--------------------------------+-----------------------------+
| Rack ASN Block                 | 4200000000-4200065534       |
+--------------------------------+-----------------------------+
| Rack Border Leaf ASN Block     | 4200065535-4200065635       |
+--------------------------------+-----------------------------+
```

```
| Single Rack Deployment         | Yes                          |
+--------------------------------+------------------------------+
| P2P IP Type                    | numbered                     |
+--------------------------------+------------------------------+
| Any cast MAC                   | 0201.0101.0101               |
+--------------------------------+------------------------------+
| IPV6 Any cast MAC              | 0201.0101.0102               |
+--------------------------------+------------------------------+
| MAC Aging Timeout              | 1800                         |
+--------------------------------+------------------------------+
| MAC Aging Conversational       | 300                          |
| Timeout                        |                              |
+--------------------------------+------------------------------+
| MAC Move Limit                 | 20                           |
+--------------------------------+------------------------------+
| Duplicate MAC Timer            | 5                            |
+--------------------------------+------------------------------+
| Duplicate MAC Timer MAX Count  | 3                            |
+--------------------------------+------------------------------+
| BFD Enable                     | Yes                          |
+--------------------------------+------------------------------+
| BFD Tx                         | 400                          |
+--------------------------------+------------------------------+
| BFD Rx                         | 400                          |
+--------------------------------+------------------------------+
| BFD Multiplier                 | 5                            |
+--------------------------------+------------------------------+
| BGP MultiHop                   | 4                            |
+--------------------------------+------------------------------+
| MaxPaths                       | 8                            |
+--------------------------------+------------------------------+
| AllowAsIn                      | 0                            |
+--------------------------------+------------------------------+
| MTU                            | 9216                         |
+--------------------------------+------------------------------+
| IPMTU                          | 9100                         |
+--------------------------------+------------------------------+
| MCT Link IP Range              | 10.20.20.0/24                |
+--------------------------------+------------------------------+
| MCT PortChannel                | 64                           |
+--------------------------------+------------------------------+
| LACP Timeout                   | short                        |
+--------------------------------+------------------------------+
| Control Vlan                   | 4090                         |
+--------------------------------+------------------------------+
| Control VE                     | 4090                         |
+--------------------------------+------------------------------+
| Rack L3 Backup IP Range        | 10.30.30.0/24                |
+--------------------------------+------------------------------+
| Rack Underlay EBGP Peer Group  | underlay-ebgp-group          |
+--------------------------------+------------------------------+
| Rack Overlay EBGP Peer Group   | overlay-ebgp-group           |
+--------------------------------+------------------------------+
| Rack L3 backup port            | 0/48                         |
+--------------------------------+------------------------------+
| Rack MCT Ports                 | 0/46,0/47                    |
+--------------------------------+------------------------------+
```

Configure an IP Multicast Fabric

Tip

If any devices in a fabric are in "admin-down" state, use of the following commands in that same fabric will not add or delete devices in the fabric: **`efa fabric device add-bulk`** and **`efa fabric device remove`**.

1. Create a Clos fabric.

```
# efa fabric create --name clos_fabric --type clos --stage 3
```

Note

Optimized replication is not supported on small data center fabric.

2. Enable multicast fabric settings.

```
# efa fabric setting update --optimized-replication-enable yes --name clos_fabric
```

3. (Optional) Override the default MDT group and group range.

```
# efa fabric setting update --name clos_fabric --mdtgroup-range <A.B.C.D/L> --default-
mdtgroup <A.B.C.D>

# efa fabric setting update --name clos_fabric --mdtgroup-range 239.0.0.0/8 --default-
mdtgroup 239.1.1.1
```

4. Verify the fabric settings.

```
# efa fabric setting show --name clos_fabric --advanced

+---------------------------------+-------------------------+
| Optimized Replication Enable    | Yes                     |
+---------------------------------+-------------------------+
| MDT Group IPv4 Range            | 238.0.0.0/8             |
+---------------------------------+-------------------------+
| Default MDT Group IPv4 address | 238.1.1.1               |
+---------------------------------+-------------------------+
```

5. Add devices to the fabric.

```
# efa fabric device add-bulk --name clos_fabric --leaf Leaf1IP,Leaf2IP,Leaf3IP,Leaf4IP
--spine Spine1IP,Spine2IP --username admin --password password
```

6. Configure the fabric.

```
# efa fabric configure -name clos_fabric
```

7. Verify the fabric configuration.

```
# efa fabric show-config --name clos_fabric --advanced
```

- All underlay configuration and overlay configurations are pushed to the devices and underlay topology is operational.
- All BGP connections between leaf and spine nodes are established and neighbors are reachable.
- Basic overlay configuration with optimized replication is configured.
- All device configurations are applied to the devices in fabric. For more information, see Device Configuration on page 75.

8. Create a tenant and endpoint group to bring up the multicast tunnels with leaf nodes.

```
# efa tenant create --name tenant1 --l2-vni-range 10002-14190 --l3-vni-range
14191-14200 --vrf-count 10 --vlan-range 2-4090 --port
Leaf1IP[0/12-16],Leaf2IP[0/12-16], Leaf3IP[0/12-16],Leaf4IP[0/12-16] --description
Subscriber1

# efa tenant epg create --name epg1 --tenant tenant1 --port
Leaf1IP[0/15],Leaf2IP[0/15],Leaf3IP[0/16 --switchport-mode trunk --ctag-range 100
```

Device Configuration

When IP multicast fabric is enabled, the following device configurations are pushed to all devices in the fabric.

- `router pim` for default VRF is enabled on all nodes.
- `ip prefix-list` is configured on all nodes with the `mdt-range` specified in fabric settings or default range.
- Under `router-pim` mode, PIM-SSM is enabled for all nodes with range specified in `ip prefix-list`.
- Under `interface` mode, `PIM sparse` mode is enabled for all fabric links.
- Under overlay-gateway, `optimized replication` is enabled on all leaf nodes.
- Under `optimized replication` mode, `underlay-default-mdtgroup` is configured to default value specified in fabric settings on all leaf nodes.

Configure Drift and Reconcile on Multicast Fabric

1. Configure drift and reconcile on multicast fabric.

```
# efa fabric debug device drift --ip A.B.C.D --name dni --reconcile
```

Note

Any drift in Router PIM, IP prefix list, and overlay-gateway EFA configuration compared to the configured device is detected and reconciled.

2. Configure drift and reconcile of all services on a device.

```
# efa inventory drift-reconcile execute --ip A.B.C.D --reconcile
```

View Fabric Details

You can use several commands to view the details of topologies and configuration in your fabric.

Table 13: Fabric show commands

Command	Description
`efa fabric topology show overlay`	Shows the overlay (VxLAN tunnels) connectivity of devices in a fabric.
`efa fabric topology show physical`	Shows the physical topology (LLDP neighbors) connectivity of devices in a fabric.
`efa fabric topology show underlay`	Shows the underlay (BGP neighbors) connectivity of devices in a fabric.
`efa fabric show`	Displays fabric details.
`efa fabric show-config`	Displays fabric configuration details for the specified role (leaf, spine, super-spine, border leaf) or IP address.
`efa fabric show summary`	Displays a summary of all fabrics or of the specified fabric.

The following is an example of output from the **`efa fabric topology show overlay`** command.

```
efa fabric topology show overlay --name fabric1
+-----------+--------------------------+--------------------------+----------------
+---------------+-------------+------------+----------
| OVERLAY     |      SOURCE LEAF IP      |    DESTINATION LEAF IP    | SOURCE VTEP IP |
DESTINATION    | OVERLAY     | OVERLAY    | OVERLAY
  ECAP TYPE
VTEP IP          ADMIN STATE   OPER STATE   BFD STATE
+-----------+--------------------------+--------------------------+----------------
+---------------+-------------+------------+----------
| vxlan      | 10.25.225.11,10.25.225.46 | 10.24.85.76,10.24.85.74   | 172.31.254.86  |
172.31.254.81 | up          | up         | down
| vxlan      | 10.25.225.11,10.25.225.46 | 10.24.80.134,10.24.80.135 | 172.31.254.86  |
172.31.254.83 | up          | up         | down
| vxlan      | 10.24.85.76,10.24.85.74   | 10.25.225.11,10.25.225.46 | 172.31.254.81  |
172.31.254.86 | up          | up         | down
| vxlan      | 10.24.85.76,10.24.85.74   | 10.24.80.134,10.24.80.135 | 172.31.254.81  |
172.31.254.83 | up          | up         | down
| vxlan      | 10.24.80.134,10.24.80.135 | 10.25.225.11,10.25.225.46 | 172.31.254.83  |
172.31.254.86 | up          | up         | down
| vxlan      | 10.24.80.134,10.24.80.135 | 10.24.85.76,10.24.85.74   | 172.31.254.83  |
172.31.254.81 | up          | up         | down
+-----------+--------------------------+--------------------------+----------------
+---------------+-------------+------------+----------
```

The following is an example of output from the **`efa fabric show`** command.

```
efa fabric topology show overlay --name fabric1
+-------------+-------+-------------+-----------+------+-------------+-------------
+------------+----------+------+-------
| IP ADDRESS   | RACK  |  HOST NAME  | ASN       | ROLE | DEVICE STATE | APP STATE   |
CONFIG     | PENDING  | VTLB | LB ID
```

```
GEN REASON   CONFIGS    ID
+-------------+-------+-------------+-----------+------+--------------+-------------
+------------+----------+------+-------
| 10.24.51.131 | rack2 | Freedom-07  | 4200000001| leaf | provisioned  | cfg in-sync |
NA         | NA       | 2    | 1
| 10.25.255.58 | rack2 | Freedom-04  | 4200000001| leaf | provisioned  | cfg in-sync |
NA         | NA       | 2    | 1
| 10.24.51.135 | rack1 | Freedom-06  | 4200000000| leaf | provisioned  | cfg in-sync |
NA         | NA       | 2    | 1
| 10.24.48.131 | rack1 | Freedom-05  | 4200000002| leaf | provisioned  | cfg in-sync |
NA         | NA       | 2    | 1
```

Edit Fabric Settings

You can edit certain fabric settings after the fabric configuration. Changes in the fabric settings are displayed in the **`efa fabric show`** output. The **`efa fabric show`** output marks the changed settings as `Updated Fabric Settings` and corresponding modification codes are displayed.

```
efa fabric show output:
...
Updated Fabric Settings: BGP-LL
FABRIC SETTING:
BGPLL - BGP Dynamic Peer Listen Limit, BGP-MD5 - BGP MD5 Password , BFD-RX - Bfd Rx
Timer, BFD-TX - Bfd Tx Timer, BFD-MULTIPLIER - Bfd Timer,BFD-ENABLE - Enable Bfd, BGP-
MULTIHOP - BGP ebgp multihop, P2PLR - Point-to-Point Link Range, MCTLR - MCT Link Range,
LOIP - Loopback IP Range..
```

- Changes does not reflect in `app-state`, `drc-drift`, and `drc-reconcile show` outputs.
- Changes are pushed into SLX only after you run the **`efa fabric configure`** command.
- After successful execution of the **`efa fabric configure`** command, the **`efa fabric show`** command clears the fabric status and fabric settings updated codes.

Edit Fabric IP Range Settings

- You can edit the following active fabric settings even after the devices are added in fabric:
  - `--loopback-ip-range`
  - `--p2p-link-range`
  - `--mctlink-ip-range`
- You can generate increased IP or IP-Pair and update them in the available IP-Pool. There are no updates to the used-IP pair. For example,
  - Old IP range: 10.10.10.1/24
  - New IP range: 10.10.10.1/23
  - Increased IP range: 10.10.10.1/23 - 10.10.10.1/24
- The increased IP addresses are available for use when new devices are added into the fabric.
- You can edit `loopback-ip-range`, `p2p-link-range`, and `mctlink-ip-range` settings. Ensure that the prefix mask length is lower than the configured prefix mask length value within the same network.

```
'efa fabric show' output sample:
...
Fabric Name: fs, Fabric Description: , Fabric Type: non-clos, Fabric Status: settings-
updated
Updated Fabric Settings: MCTLR,P2PLR,LOIP
```

```
FABRIC SETTING:
BGPLL - BGP Dynamic Peer Listen Limit, BGP-MD5 - BGP MD5 Password , BFD-RX - Bfd Rx
Timer, BFD-TX - Bfd Tx Timer, BFD-MULTIPLIER - Bfd Timer,BFD-ENABLE - Enable Bfd, BGP-
MULTIHOP - BGP ebgp multihop, P2PLR - Point-to-Point Link Range, MCTLR - MCT Link
Range, LOIP - Loopback IP Range
```

Edit Fabric BFD Settings

- You can edit the following active fabric settings even after the devices are added in fabric:
  - `--bfd-enable <yes|no>`
  - `--bfd-tx <val>`
  - `---bfd-rx <val>`
  - `--bfd-multiplier<val>`

```
'efa fabric show' output sample:
...
Fabric Name: fs, Fabric Description: , Fabric Stage: 3, Fabric Type: clos, Fabric
Status: settings-updated

Updated Fabric Settings: BFD-ENABLE,BFD-TX,BFD-RX,BFD-MULTIPLIER

FABRIC SETTING:
BGPLL - BGP Dynamic Peer Listen Limit, BGP-MD5 - BGP MD5 Password , BFD-RX - Bfd Rx
Timer, BFD-TX - Bfd Tx Timer, BFD-MULTIPLIER - Bfd Timer,BFD-ENABLE - Enable Bfd, BGP-
MULTIHOP - BGP ebgp multihop, P2PLR - Point-to-Point Link Range, MCTLR - MCT Link
Range, LOIP - Loopback IP Range
```

- The **`efa fabric configure`** command generates BFD recipe for `router BGP`, `interface links`, `MCT port-channel interfaces`, and pushes the configuration to the SLX devices.

Fabric Settings in Active Fabric: Small Data Center Fabric

Cannot be Updated	Can be Updated	Invalid Settings for Small Data Center Fabric
--rack-l3-backup-ip-range --loopback-port-number --vtep-loopback-port-number --rack-asn-block --rack-border-leaf-asn-block --anycast-mac-address --ipv6-anycast-mac-address --mac-aging-timeout --mac-aging-conversation-timeout --mac-move-limit --duplicate-mac-timer --duplicate-mac-timer-max-count --bgp-multihop --max-paths --allow-as-in --mtu --ip-mtu --rack-underlay-ebgp-peer-group --rack-overlay-ebgp-peer-group --lacp-timeout --mct-port-channel --control-vlan --control-ve --vni-auto-map --backup-routing-enable --backup-routing-ipv4-range --backup-routing-ipv6-range	--p2p-link-range --loopback-ip-range --bfd-enable --bfd-tx --bfd-rx --bfd-multiplier --mctlink-ip-range --single-rack-deployment --md5-password-enable --md5-password --bgp-dynamic-peer-listen-limit	--spine-asn-block --super-spine-asn-block --leaf-asn-block --border-leaf-asn-block --configure-overlay-gateway --leaf-peer-group --spine-peer-group --super-spine-peer-group --optimized-replication-enable --mdtgroup-range --default-mdtgroup

Fabric Settings in Active Fabric: Clos Fabric

Cannot be Updated	Can be Updated	Invalid Settings for Clos Fabric
--loopback-port-number --vtep-loopback-port-number --spine-asn-block --super-spine-asn-block --leaf-asn-block --border-leaf-asn-block --anycast-mac-address --ipv6-anycast-mac-address --mac-aging-timeout --mac-aging-conversation-timeout --mac-move-limit --duplicate-mac-timer --duplicate-mac-timer-max-count --configure-overlay-gateway --bgp-multihop --max-paths --allow-as-in --mtu --ip-mtu --leaf-peer-group --spine-peer-group --super-spine-peer-group --lacp-timeout --mct-port-channel --control-vlan --control-ve --vni-auto-map --optimized-replication-enable --backup-routing-enable --backup-routing-ipv4-range --backup-routing-ipv6-range --mdtgroup-range --default-mdtgroup	--p2p-link-range --loopback-ip-range --bfd-enable --bfd-tx --bfd-rx --bfd-multiplier --mctlink-ip-range --md5-password-enable --md5-password --bgp-dynamic-peer-listen-limit	--rack-l3-backup-ip-range --rack-asn-block --rack-border-leaf-asn-block --single-rack-deployment --rack-underlay-ebgp-peer-group --rack-overlay-ebgp-peer-group

Fabric Event Handling

Event handling reason code is generated only in case of a drift in configuration at the device after EFA fabric is configured. The following listed events generated by RASlog event or device update is handled in fabric services:

The **efa fabric show** command displays all the events handled at fabric services.

```
CONFIG GEN REASON:
LA/LD - Link Add/Delete, IA/ID/IU - Interface Add/Delete/Update, PLC/PLD/PLU -
```

```
IPPrefixList Create/Delete/Update
MD/MU - MCT Delete/Update, OD/OU - Overlay Gateway Delete/Update, EU/ED - Evpn Delete/
Update, PC/PD/PU - RouterPim Create/Delete/Update
DD - Dependent Device Update, DA/DR - Device Add/ReAdd, ASN - Asn Update, SYS - System
Properties Update
MD5 - BGP MD5 Password, BGPU - Router BGP Update, BGPLL - BGP Listen Limit, POU - Port
Channel Update, NA - Not Applicable
```

Drift (SLX vs EFA)	App State	CONFIG GEN REASON (efa fabric show)
Evpn	Cfg-refresh	EU/ED - Evpn Delete or Update
Cluster	Cfg-refresh	MD/MU - MCT Delete or Update
Router bgp	Cfg-refresh	BGPU - Router BGP Update
Overlay gateway	Cfg-refresh	OD/OU - Overlay Gateway Delete/Update,
Interface (Phy/ Lo/Po)	Cfg-refresh	IA/ID/IU - Interface Add, Delete or Update
LLDP	Cfg-refresh	LA/LD - Link Add or Delete

Northbound IPv6 support

EFA 2.7.0 supports IPv6 via an additional virtual IP that is assigned during EFA installation or upgrade. Clients can make REST API calls to EFA via either an IPv4 or an IPv6 address. An IPv6 address is optional, but an IPv4 address is mandatory.

You can configure the API gateway with IPv4 and IPv6 addresses.

Note

- Dual mode or IPV4 only mode is supported. TPVM does not support assigning an IPV6 address via the SLX CLI.
- Upgrades from earlier releases or from single-node to multi-node and fresh install requires you to configure an IPv6 address (optional).
- Single-node installation supports IP address change for both IPv4 and IPv6 addresses.
- API gateway supports IPv6 virtual IP address for multi-node and single-node installation.

Verify the Upgrade

1. On the SLX device, run the **show efa status** command to see details of the installation and the state of services.
2. From the EFA command line, run the **efactl status** command to see the status of nodes, pods, and services.
3. Run the **`efa status`** command for concise status information.

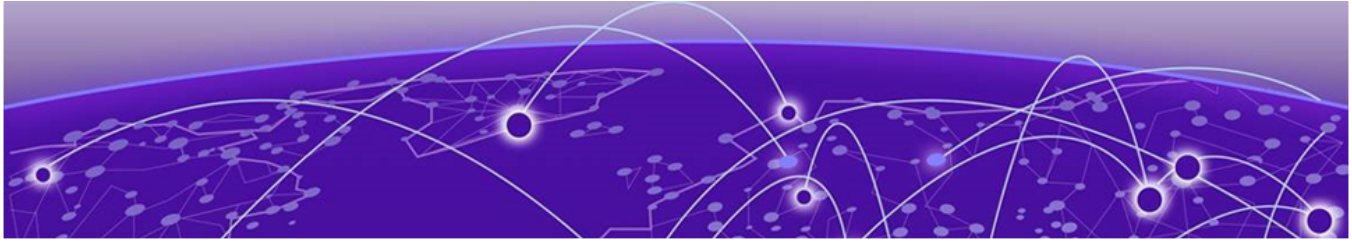
Note

For information about upgrading EFA, see the *Extreme Fabric Automation Deployment Guide, 2.7.0*.

Verify the IPv6 Address

During a fresh installation or upgrade, ensure that you have specified the valid IPv6 address.

1. When prompted, select Yes to assign an IPv6 address as a virtual IP for cluster.
2. Specify the virtual IPv6 address. The system validates the IPv6 address to ensure it is not used already. If the entered IPv6 address is already in use, change the IPv6 address.



Tenant Services Provisioning

Tenant Services Provisioning Overview on page 83
Provisioning a Tenant on page 86
Clos Fabric with Non-auto VNI Maps on page 88
Clos Fabric with Auto VNI Map on page 89
Layer 3 Network Services on page 93
Administered Partial Success on page 191
In-flight Transaction Recovery on page 206
Traffic Mirroring on page 208
Scalability on page 229

Tenant Services Provisioning Overview

Tenant Services exposes the CLI and REST API for automating the Tenant network configuration on the Clos and small data center (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.

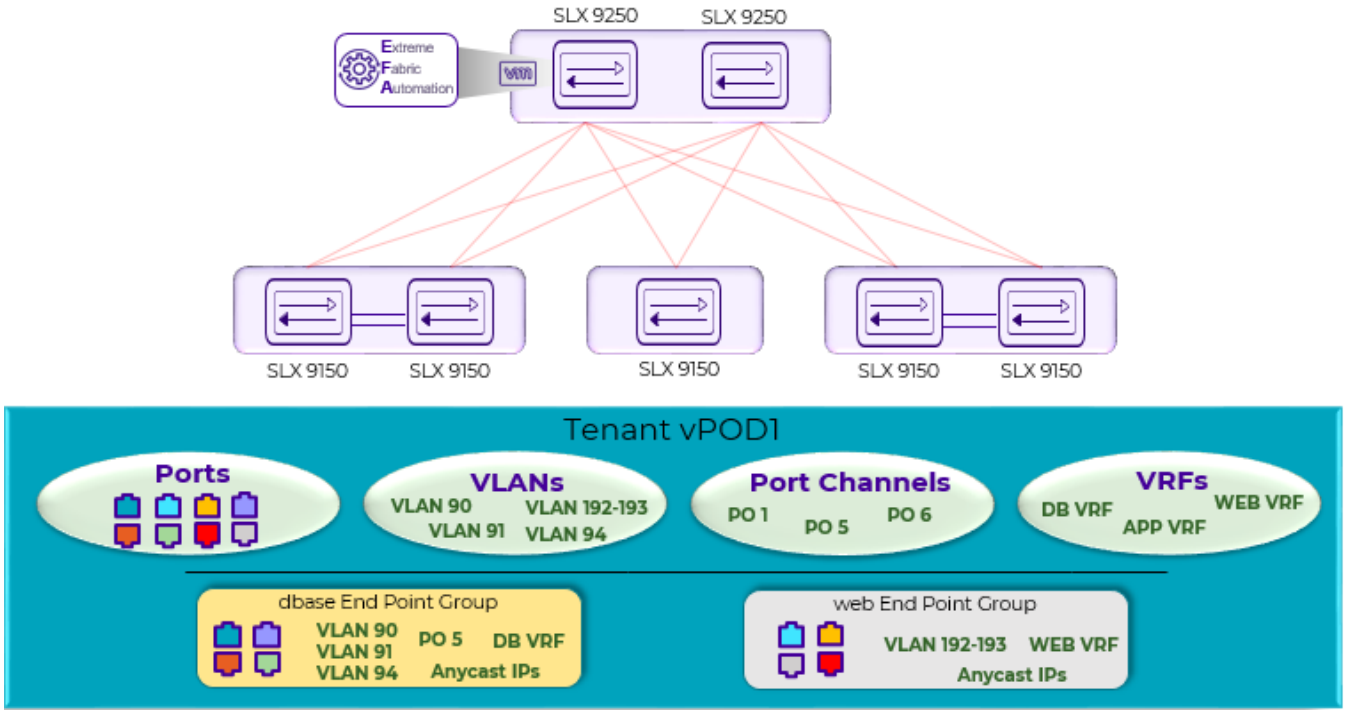


Figure 15: Tenant Services Overview

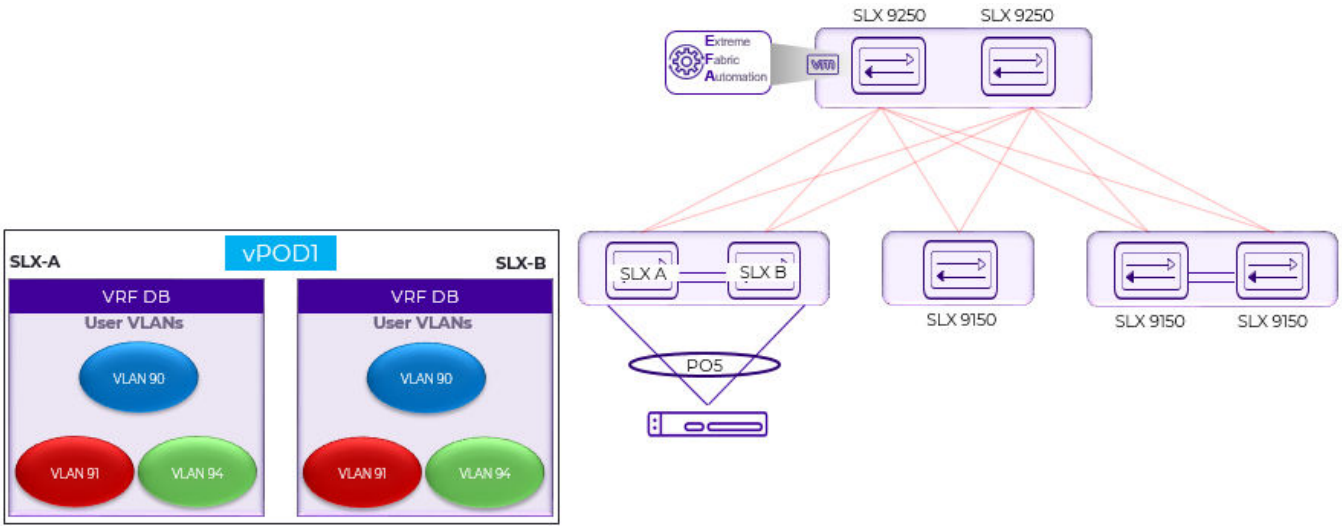


Figure 16: Tenant Name vPOD1, VRF Name DB

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. You 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 Endpoint Group, are not deleted whereas references for ports or port-channels of deleted devices are removed.

Provisioning a Tenant

This high-level process describes the tasks that you complete to provision a tenant in your EFA fabric.

A tenant is a group of users that own or have access to shared resources.

Create a Tenant

This step in the process provides a name for the tenant and identifies the resources that are reserved for the tenant, including the Layer 2 and Layer 3 Virtual Network Identifiers (VNI), VLAN, VRF, and bridge domain. You can later apply these resources to an endpoint group.

Use the **`efa tenant create`** command to create your tenant. For syntax and command examples, see the *Extreme Fabric Automation Command Reference, 2.7.0*.

For related information, see the following topics.

- Clos Fabric with Auto VNI Map on page 89
- Configure MaxPaths on Tenant VRF on page 110
- Sharing Resources Across Tenants

Create a Port Channel

This step in the process creates the port channel for the tenant. A port channel, also known as a Link Aggregation Group (LAG) is a communication link between devices. You can specify speed, LACP negotiation, port, port channel number, LACP timeout, and the number of links that are required to be up.

Use the **`efa tenant po create`** command to create the port channel. For syntax and command examples, see the *Extreme Fabric Automation Command Reference, 2.7.0*.

For related information, see the following topics.

- Configure CEP Cluster Tracking for Endpoint Groups on page 126
- Exclusion of VLANs and Bridge from Cluster Instance on page 164
- Sharing Resources Across Tenants
- Configure Minimum Link Count on Port Channel on page 94

Create the Tenant VRF

This step in the process sets up virtual routing and forwarding (VRF) for the tenant. You can specify the VRF name and the associated tenant, the target VPN community, the Route Target and Route Distinguisher, the local ASN, IPv4 and IPv6 static BFD routes, IPv4 and IPv6 static next hop routes, the number of load sharing paths, the redistribute type, whether resilient hashing is on SLX devices, and the routing type.

Use the **`efa tenant vrf create`** command to configure VRF. For syntax and command examples, see the *Extreme Fabric Automation Command Reference, 2.7.0*.

For related information, see the following topics.

- IPv6 Support on page 156
- Configure Static VRF Route on page 100
- Configure BFD on Static VRF Route on page 101
- Configure Local ASN on Tenant VRF on page 101
- Configure Resilient Hashing on Tenant VRF on page 108
- Configure MaxPaths on Tenant VRF on page 110
- Configure Redistribute Attribute on Tenant VRF on page 111
- Sharing Resources Across Tenants

Create the Tenant Endpoint Group

This step in the process creates the endpoint group for the tenant. An endpoint group is a logical group of endpoints, which are devices that are connected to the network. You can specify such parameters as group name, the IP address, the port channels, the switchport mode, the BGP service type, native VLAN, CTAG range, the associated VRF, the Layer 2 and Layer 3 VNI, the bridge domain, and neighbor discovery preferences.

Use the **`efa tenant epg create`** command to create the endpoint group. For syntax and command examples, see the *Extreme Fabric Automation Command Reference, 2.7.0*.

For related information, see the following topics.

- Clos Fabric with Auto VNI Map on page 89
- IPv6 Support on page 156
- Configure Static VRF Route on page 100
- Configure BFD on Static VRF Route on page 101
- Configure CEP Cluster Tracking for Endpoint Groups on page 126
- Exclusion of VLANs and Bridge from Cluster Instance on page 164
- Configure MaxPaths on Tenant VRF on page 110
- Configure Redistribute Attribute on Tenant VRF on page 111

Create the BGP Peer Group

This step in the process creates the Border Gateway Protocol (BGP) peer group, which is a set of BGP neighbors that share outbound policies. You can specify the group name, the group ASN, the BFD (bidirectional forwarding detection) properties of the group, source IP and next hop information, and the name of the associated tenant.

Use the **`efa tenant service bgp peer-group create`** command to create the peer group. For syntax and command examples, see the *Extreme Fabric Automation Command Reference, 2.7.0*.

For related information, see BGP Peer Group on page 133.

Create the BGP Peer

This step in the process creates the BGP peer for the tenant. BGP peers are devices that exchange BGP routing information. You can specify the IPv4 and IPv6 dynamic unicast neighbors, the IPv4 and IPv6 unicast neighbors, and the IPv4 and IPv6 BFD unicast neighbors. Additional parameters include next-hop-self information, source IP address, the name of the peer, and the name of the associated tenant.

Use the **`efa tenant service bgp peer create`** command to create the peer. For syntax and command examples, see the *Extreme Fabric Automation Command Reference, 2.7.0*.

For related information, see BGP Static Peer on page 135 and BGP Dynamic Peer on page 137.

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.

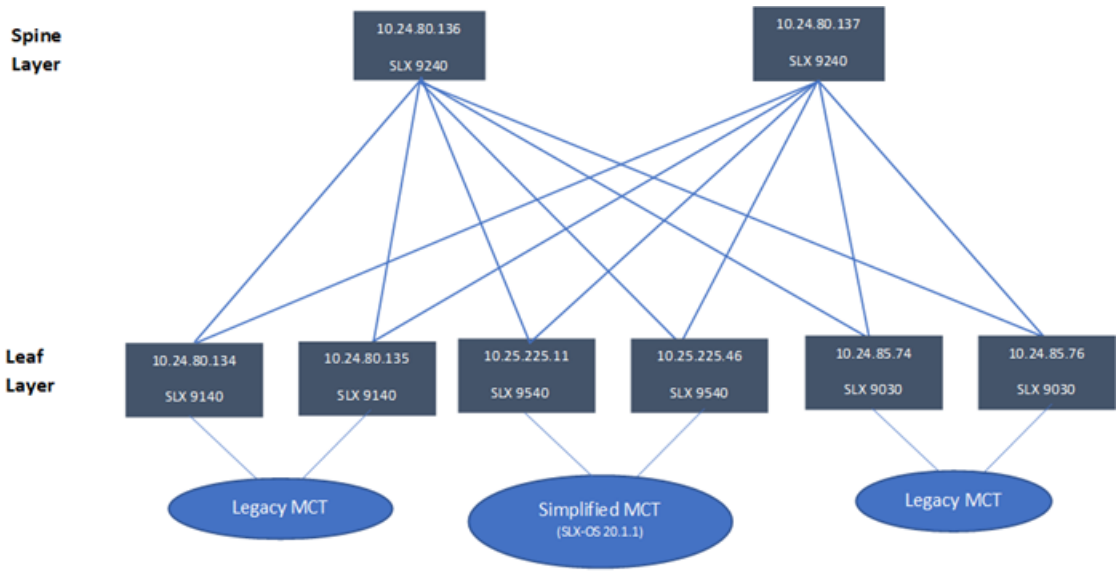


Figure 17: 3-stage Clos topology

The following commands configure 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
  - You cannot reserve l2-vni-range or l3-vni-range for a given tenant.
  - You cannot provide a specific l2-vni or l3-vni in an endpoint group.
- 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 are 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],10.2
5.225.11[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],10.2
5.225.11[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.

Configure Description on Port Channel

Configure “description” per EFA port-channel which gets configured on SLX port-channel. The default value of po “description” is **“EFA Port-channel <efa-po-name>”**. You can configure PO “description” during `po create` and `po update` operations.

Note

When you upgrade from EFA 2.3.2 to EFA 2.4.0 or EFA 2.5.0, all the POs contains the default “description” and is displayed as part of **`efa tenant po show --detail`**.

1. Configure Description during Port-Channel Create.

```
efa tenant po create --name <po-name> --tenant <tenant-name> --description <po-
description>
               --speed <100Mbps|1Gbps|10Gbps|25Gbps|40Gbps|100Gbps> --negotiation
<active|passive|static>
    --port <list-of-po-members> --min-link-count <min-link-count> --number <po-number>
--lacp-timeout <short|long>
```

2. Configure Description during Port-Channel Update.

```
efa tenant po update --name <po-name> --tenant <tenant-name>
             --operation <port-add|port-delete|lacp-timeout|description|min-link-
count>
             --port <list-of-po-members> --lacp-timeout string <short|long> --min-
link-count <min-link-count>
             --description <po-description>
```

Example

```
efa tenant po create --name ten1po1 --tenant ten1 --port
10.20.246.15[0/1],10.20.246.16[0/1] --speed 10Gbps --negotiation active --description
tenant1po1


efa tenant po create --name ten1po2 --tenant ten1 --port
10.20.246.15[0/2],10.20.246.16[0/2] --speed 10Gbps --negotiation active

efa tenant po update --name ten1po1 --tenant ten1 --operation description --
description tenat1po1changed
```

efa tenant po show --name ten1po1 --tenant ten1 --detail

```
Name             : ten1po1
Tenant           : ten1
ID               : 1
Description      : tenat1po1changed
Speed            : 10Gbps
Negotiation      : active
Min Link Count   : 1
Lacp Timeout     : long
Ports            : 10.20.246.15[0/1]
                 : 10.20.246.16[0/1]
State            : po-created
Dev State        : provisioned
App State        : cfg-in-sync
```

efa tenant po show --name ten1po2 --tenant ten1 --detail

```
Name             : ten1po2
Tenant           : ten1
ID               : 2
Description      : EFA Port-channel
ten1po2
Speed            : 10Gbps
Negotiation      : active
Min Link Count   : 1
Lacp Timeout     : long
Ports            : 10.20.246.15[0/2]
                 : 10.20.246.16[0/2]
State            : po-created
Dev State        : provisioned
App State        : cfg-in-sync
```

Configure Minimum Link Count on Port Channel

Minimum number of links per port-channel indicates the least number of links that need to be operationally UP. This indicates the number of port-channel as operationally UP. You can provide an optional "min-link-count" per EFA port-channel during PO create and PO update operations, which gets configured on the SLX port-channel.

Default value of min-link-count for a port-channel is 1, which is same as the SLX default value. Update the min-link-count attribute value using `min-link-count` operation. EFA validates the po-member-port-count >= po-min-link-count per device.

Note

During upgrade from EFA 2.3.0 to EFA 2.4.0 or EFA 2.5.0, the min-link-count for the port-channels is set to the default value 1.

- Empty PO: EFA 2.4.0 and EFA 2.5.0 does not support empty PO. Therefore, during upgrade from EFA 2.3.2 to EFA 2.4.0 or EFA 2.5.0, all the empty POs are marked with “delete-pending” state.
- Non empty PO: During upgrade from EFA 2.3.2 to EFA 2.4.0 or EFA 2.5.0, all the non-empty PO gets configured with the default value (1) of “min-link-count”, and the default value is displayed as part of “efa tenant po show” output.
- Single Homed to Dual Homed PO conversion is not allowed in EFA 2.4.0 and EFA 2.5.0.

1. Configure Minimum Link Count during Port-Channel Create.

```
efa tenant po create --name <po-name> --tenant <tenant-name> --description <po-
description>
        --speed <100Mbps|1Gbps|10Gbps|25Gbps|40Gbps|100Gbps> --negotiation <active|
passive|static>
     --port <list-of-po-members> --min-link-count <min-link-count>
        --number <po-number> --lacp-timeout <short|long>
```

2. Configure Minimum Link Count during Port-Channel Update.

```
efa tenant po update --name <po-name> --tenant <tenant-name>
         --operation <port-add|port-delete|lacp-timeout|description|min-link-count>
         --port <list-of-po-members> --lacp-timeout string <short|long> --min-link-
count <min-link-count>
         --description <po-description>
```

Example

```
efa tenant po create --name ten1po1 --tenant ten1 --port
10.20.246.15[0/1-2],10.20.246.16[0/1-2]
--speed 10Gbps --negotiation active --description tenant1po1 --min-link-count 2

efa tenant po create --name ten1po2 --tenant ten1 --port
10.20.246.15[0/3],10.20.246.16[0/3]
--speed 10Gbps --negotiation active

efa tenant po update --name ten1po1 --tenant ten1 --operation port-delete --port
10.20.246.15[0/1],10.20.246.16[0/1] --min-link-count 1

efa tenant po update --name ten1po1 --tenant ten1 --operation port-add --port
10.20.246.15[0/1],10.20.246.16[0/1] --min-link-count 2

efa tenant po update --name ten1po1 --tenant ten1 --operation min-link-count --min-
link-count 1

efa tenant po update --name ten1po1 --tenant ten1 --operation min-link-count --min-
link-count 2

efa tenant po show
+---------+--------+----+--------+-------------+----------+---------
+---------------------+------------+-------------+-------------+
|  Name   | Tenant | ID | Speed  | Negotiation | Min Link |  Lacp   |
Ports        |   State    |  Dev State  |  App State  |
|         |        |    |        |             |  Count   | Timeout
```

```
|                     |            |             |             |
+---------+--------+----+--------+-------------+----------+---------
+---------------------+------------+-------------+-------------+
| ten1po1 |  ten1  | 1  | 10Gbps |   active    |    2     |  long   |
10.20.246.15[0/1-2] | po-created | provisioned | cfg-in-sync |
|         |        |    |        |             |          |         |
10.20.246.16[0/1-2] |            |             |             |
+---------+--------+----+--------+-------------+----------+---------
+---------------------+------------+-------------+-------------+
| ten1po2 |  ten1  | 2  | 10Gbps |   active    |    1     |  long   |
10.20.246.15[0/3]  | po-created | provisioned | cfg-in-sync |
|         |        |    |        |             |          |         |
10.20.246.16[0/3]  |            |             |             |
+---------+--------+----+--------+-------------+----------+---------
+---------------------+------------+-------------+-------------+
```

Configure Backup Routing on Tenant VRF

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

IPv4 and IPv6 range are the input at the fabric level. A pair of IPv4 and IPv6 address is allocated to each MCT pair across all the tenant VRFs, and the BGP session is established between the same IP Pair.

1. Allocate a pair of IPv4 and IPv6 address to each MCT pair across all the tenant VRFs.
   a. Allocate a Bridge domain per VRF for backup routing.
   b. Allocate a corresponding router-interface VE per BD or VRF.
   c. Assign the IPv4 and IPv6 address allocated to each device on each of the VE interface.

      Note
      Same IPv4 and IPv6 address is allocated on each of the VE interface which belongs to different VRF.

   d. Establish IBGP IPv4 neighborship with the MCT peer on a set of IP address per VRF.
   e. Establish IBGP IPv6 neighborship with the MCT peer on a set of IPv6 address per VRF.
   f. Configure "next-hop-self" on both the IPv4 and IPv6 neighbor.
   g. 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 when backup routing is enabled:

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

2. Configure devices on tenant VRFs.

Table 15: 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 16: 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 16: 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 ! !

Route-Distinguisher (RD) allocation independent of Route-Target (RT)

Provisioning in EFA pre-2.5.0

EFA auto allocates VRF RT in the format `xx:yy`.

EFA auto allocates VRF RD in the format `<router-id>:<lowest-yy-of-rt-in-a-given-vrf>`.

If users have not provided explicit RT values, then EFA uniquely allocates the RT values for each VRF. Since RD allocation is based on the RT value, EFA uniquely allocates the RD value for each VRF.

But if users provide RT values (in the format xx:yy and yy is common across multiple VRFs), then the RD which is derived based on the yy (<router-id>:<lowest-yy-of-rt>), cannot be unique across the VRFs and hence results in the failure indicating multiple VRFs cannot have common RD.

```
efa tenant vrf create --tenant "ten1" --name "ten1vrf1" --routing-type "distributed" --rt-
type export --rt 65010:1 --rt-type import --rt 65010:2 --max-path 8 --redistribute
connected

efa tenant vrf create --tenant "ten1" --name "ten1vrf2" --routing-type "distributed" --rt-
type export --rt 65010:2 --rt-type import --rt 65010:1 --max-path 8 --redistribute
connected

efa tenant epg create --tenant "ten1" --name "ten1epg1" --type extension --switchport-
mode trunk --single-homed-bfd-session-type auto --po ten1po1 --vrf ten1vrf1 --ctag-range
25 --l3-vni 32821 --anycast-ip 25:10.0.21.1/24 --ctag-description "25:Tenant L3 Extended
VLAN" --l2-vni 25:32770 --suppress-arp 25:true --suppress-nd 25:false

efa tenant epg create --tenant "ten1" --name "ten1epg2" --type extension --switchport-
mode trunk --single-homed-bfd-session-type auto --po ten1po1 --vrf ten1vrf2 --ctag-range
26 --l3-vni 32823 --anycast-ip 26:11.0.21.1/24 --ctag-description "26:Tenant L3 Extended
VLAN" --l2-vni 26:32771 --suppress-arp 26:true --suppress-nd 26:false

Device: 10.20.246.15
Network Policy Error: Vrf ten1vrf2 creation failed due to netconf rpc [error] '% Error:
This RD is already configured for another VRF',
Device: 10.20.246.16
Network Policy Error: Vrf ten1vrf2  creation failed due to netconf rpc [error] '% Error:
This RD is already configured for another VRF',
```

Provisioning in EFA 2.5.0 or above

EFA auto allocates VRF RT in the format xx:**yy**

EFA auto allocates VRF RD in the format <router-id>:<unique-number-per-vrf-independent-of-rt>

Since RD value has no relation to yy value of RT anymore, and since EFA auto allocates the unique-number-per-vrf which is appended to the router-id, the uniqueness of RD is always guaranteed.

```
efa tenant vrf create --tenant "ten1" --name "ten1vrf1" --routing-type "distributed" --rt-type export --rt 65010:1 --rt-type import --rt 65010:2 --max-path 8 --redistribute connected

efa tenant vrf create --tenant "ten1" --name "ten1vrf2" --routing-type "distributed" --rt-type export --rt 65010:2 --rt-type import --rt 65010:1 --max-path 8 --redistribute connected

efa tenant epg create --tenant "ten1" --name "ten1epg1" --type extension --switchport-mode trunk --single-homed-bfd-session-type auto --po ten1po1 --vrf ten1vrf1 --ctag-range 25 --l3-vni 32821 --anycast-ip 25:10.0.21.1/24 --ctag-description "25:Tenant L3 Extended VLAN" --l2-vni 25:32770 --suppress-arp 25:true --suppress-nd 25:false

efa tenant epg create --tenant "ten1" --name "ten1epg2" --type extension --switchport-mode trunk --single-homed-bfd-session-type auto --po ten1po1 --vrf ten1vrf2 --ctag-range 26 --l3-vni 32823 --anycast-ip 26:11.0.21.1/24 --ctag-description "26:Tenant L3 Extended VLAN" --l2-vni 26:32771 --suppress-arp 26:true --suppress-nd 26:false
```

Switch Config in EFA 2.5.0 or above

```
Rack1-Device1# show running-config
vrf
vrf ten1vrf1
rd 172.31.254.40:1
evpn irb ve 8192
address-family ipv4 unicast
  route-target export 65010:1 evpn
  route-target import 65010:2 evpn
!
address-family ipv6 unicast
  route-target export 65010:1 evpn
  route-target import 65010:2 evpn
!
!
vrf ten1vrf2
rd 172.31.254.40:2
evpn irb ve 8190
address-family ipv4 unicast
  route-target export 65010:2 evpn
  route-target import 65010:1 evpn
!
address-family ipv6 unicast
  route-target export 65010:2 evpn
  route-target import 65010:1 evpn
!
```

```
Rack1-Device2# show running-config
vrf
vrf ten1vrf1
rd 172.31.254.209:1
evpn irb ve 8192
address-family ipv4 unicast
  route-target export 65010:1 evpn
  route-target import 65010:2 evpn
!
address-family ipv6 unicast
  route-target export 65010:1 evpn
  route-target import 65010:2 evpn
!
!
vrf ten1vrf2
rd 172.31.254.209:2
evpn irb ve 8190
address-family ipv4 unicast
  route-target export 65010:2 evpn
  route-target import 65010:1 evpn
!
address-family ipv6 unicast
  route-target export 65010:2 evpn
  route-target import 65010:1 evpn
!
```

Configure Static VRF Route

The Static Route Configuration at the tenant VRF level enables you to provide static routes per tenant VRF.

1. To configure static VRF route, run the following commands:

```
# efa tenant vrf create [ --name vrf name | --tenant tenant name | --rttype <both |
import | export > | --rt | --ipv4-static-route-next-hop < device ip,ipv4 static route
network,nexthop ip,nexthop distance > | --ipv6-static-route-next-hop < device ip,ipv4
static route network,nexthop ip,nexthop distance > | --local-asn | --ipv4-static-route-
bfd < device-ip,dest-ipv4-addr,source-ipv4-addr[interval,minrx,multiplier] > | --ipv6-
static-route-bfd < device-ip,dest-ipv4-addr,source-ipv4-addr[interval,min-
rx,multiplier] > | --max-path uint <1-64> | --redistribute < static | connected >| --
rh-max-path uint | --rh-ecmp-enable | --help ]
```

2. To update static VRF route, run the following commands:

```
# efa tenant vrf update [ --name <vrf-name> --tenant <tenant-name> --operation <
staticroute-add | static-route-delete> --ipv6static-route-next-hop <destination, next-
hop> --ipv4static-route-next-hop <destination, next-hop> efa tenant vrf update [ --
name vrf name | --tenant tenant name | --operation < local-asn-add | local-asn-delete
| static-route-bfd-add | static-route-bfd-delete | static-route-add | static-route-
delete | max-path-add | max-path-delete | redistribute-add | redistributedelete | rh-
max-path-add | rh-max-path-delete | rh-ecmp-update > | --local-asn | --ipv4-static-
route-bfd < device IP,dest-ipv4-addr,source-ipv4-addr[interval,min-rx,multiplielr] > |
--ipv6-staticroute-bfd < device ip,dest-ipv6-addr,source-ipv6-
addr[interval,minrx,multiplielr] > | --ipv4-static-route-next-hop < device-ip,destipv4-
addr,source-ipv4-addr[interval,min-rx,multiplier] > | --ipv6-static-route-next-hop <
device-ip,dest-ipv4-addr,source-ipv4-addr[interval,min-rx,multiplier] > | --max-path
uint <1-64> | --redistribute < static | connected > | --rh-max-path uint | --rh-
ecmpenable | --help ]
```

```
# efa tenant vrf create --name red --tenant tenant11 --ipv6-static-route-next-hop
10.24.80.134,2000::/64,1001::2 --ipv6-static-route-next-hop
10.24.80.134,2000::/64,1002::2 --ipv6-static-route-next-hop
10.24.80.134,2000::/64,1003::2 --ipv6-static-route-next-hop
10.24.80.134,2000::/64,1004::2 --ipv6-static-route-next-hop
10.24.80.135,2001::/64,1001::2,4 --ipv6-static-route-next-hop
10.24.80.135,2001::/64,1002::2 --ipv6-static-route-next-hop
10.24.80.135,2001::/64,1003::2 --ipv6-static-route-next-hop
10.24.80.135,2001::/64,1004::2 --ipv4-static-route-next-hop
10.24.80.134,22.0.0.0/24,13.0.0.1,2 --ipv4-static-route-next-hop
10.24.80.134,22.0.0.0/24,13.0.0.2 --ipv4-static-route-next-hop
10.24.80.134,22.0.0.0/24,13.0.0.3 --ipv4-static-route-next-hop
10.24.80.134,22.0.0.0/24,13.0.0.4 --ipv4-static-route-next-hop
10.24.80.135,23.0.0.0/24,13.0.0.1 --ipv4-static-route-next-hop
10.24.80.135,23.0.0.0/24,13.0.0.2 --ipv4-static-route-next-hop
10.24.80.135,23.0.0.0/24,13.0.0.3 --ipv4-static-route-next-hop
10.24.80.135,23.0.0.0/24,13.0.0.4
# efa tenant vrf show --name red --tenant tenant11

================================================================================Name
          : red
Tenant Name     : tenant11
L3 VNI          :
Route Target    :
Static Route    : Switch-IP, Network->Nh1[distance],Nh2[distance],..
                : 10.24.80.134, 22.0.0.0/24->13.0.0.1[2],13.0.0.2,13.0.0.3,13.0.0.4
                  10.24.80.135, 23.0.0.0/24->13.0.0.1,13.0.0.2,13.0.0.3,13.0.0.4
                  10.24.80.134, 2000::/64->1001::2,1002::2,1003::2,1004::2
                  10.24.80.135, 2001::/64->1001::2[4],1002::2,1003::2,1004::2
```

```
Local Asn       :
================================================================================

efa tenant epg create --name ten1epg1 --tenant tenant1 --port
10.24.80.134[0/11],10.24.80.135[0/11] --switchport-mode trunk -ctag-range 11 --vrf red -
anycast-ip 11:10.10.11.1/24
```

Configure BFD on Static VRF Route

At the tenant VRF level, Bidirectional Forwarding Detection (BFD) configuration provides an option to use static route BFD timers.

Note
For more information on BFD timers, see BFD timers for router-bgp BFD and static-route BFD sessions on page 190.

1. To configure BFD when you create a tenant VRF, run the following commands:

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

2. To configure BFD on an existing tenant VRF, run the following commands:

```
# efa tenant vrf update -name <vrf-name> --tenant <tenant-name> --operation <static-
route-bfd-add|static-route-bfd-delete> --ipv6-static-route-bfd <switch-ip, destination-
ip, source-ip, bfd-min-tx, bfd-min-rx, bfd-multiplier> --ipv4static-route-bfd  <switch-
ip, destination-ip, source-ip, bfd-min-tx, bfd-min-rx, bfd-multiplier>
```

```
# efa tenant vrf create --name red --tenant tenant11 --ipv6-static-route-bfd
10.24.80.134,1001::2,1001::1,100,200,5 --ipv6-static-route-bfd
10.24.80.135,1011::2,1011::1,100,200,5 --ipv6-static-route-bfd 10.24.80.134,1002::2,
1002::1 --ipv6-static-route-bfd 10.24.80.135,1012::2, 1012::1 --ipv4-static-route-bfd
10.24.80.134,13.0.0.1,13.0.0.9,200,300,6 --ipv4-static-route-bfd
10.24.80.135,13.0.1.1,13.0.1.9,200,300,6 --ipv4-static-route-bfd
10.24.80.134,13.0.0.2,13.0.0.10 --ipv4-static-route-bfd 10.24.80.135,13.0.1.2,13.0.1.10

# efa tenant epg create --name ten1epg1 --tenant tenant1 --port
10.24.80.134[0/11],10.24.80.135[0/11] --switchport-mode trunk -ctag-range 11 --vrf red -
anycast-ip 11:10.10.11.1/24
```

Configure Local ASN on Tenant VRF

At the tenant VRF level, this option enables you to provide local-asn per tenant VRF.

1. To configure local-asn when you create a tenant VRF, run the following commands:

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

2. To configure local-asn on an existing tenant VRF, run the following commands:

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

Note
You must ensure that the local-asn support on IPv6 AF is checked.

Update Local ASN on VRF

EFA 2.5.0 supports update of local-asn on a VRF which is already used in an endpoint group.

Backup Routing

- EFA automates the backup routing configuration among the MCT nodes by configuring ipv4 or ipv6 IBGP neighborship between the MCT nodes.
- When the local-asn for a VRF (used in an endpoint group) is updated using the local-asn-add operation, the remote-asn of the backup routing IPv4 or IPv6 IBGP neighbors also gets updated.
- When the remote-asn of an existing backup routing BGP neighbor is updated, the corresponding BGP session is reset using `clear ip bgp neighbor <neighbor-ip> vrf <tenant-vrf-name>`, which lead to traffic disruption till the session is up.
- When the local-asn for a VRF (used in an endpoint group) is deleted using local-asn-delete operation, the remote-asn of the backup routing IPv4 or IPv6 IBGP neighbors also gets updated to the local-asn configured at the global router bgp level followed by the backup routing bgp session reset.

BGP neighbors (established with the EFA managed fabric devices and established by the devices not managed by EFA) must modify their remote-asn based on the modified local-asn.

Configure Local ASN during VRF Create

1. Run the following command:

```
efa fabric show
Fabric Name: default, Fabric Description: Default Fabric, Fabric Stage: 3, Fabric
Type: clos, Fabric Status: created
+------------+-----+-----------+-----+------+--------------+-----------
+------------------+-----------------+---------+-------+
| IP ADDRESS | POD | HOST NAME | ASN | ROLE | DEVICE STATE | APP STATE | CONFIG GEN
REASON | PENDING CONFIGS | VTLB ID | LB ID |
+------------+-----+-----------+-----+------+--------------+-----------
+------------------+-----------------+---------+-------+
+------------+-----+-----------+-----+------+--------------+-----------
+------------------+-----------------+---------+-------+

Fabric Name: fs, Fabric Description: , Fabric Stage: 3, Fabric Type: clos, Fabric
Status: settings-updated

Updated Fabric Settings: BGP-LL

+-------------+-----+-----------+-------+-------+--------------------+-------------
+------------------+-----------------------------------------------------+---------+-------+
| IP ADDRESS | POD | HOST NAME | ASN | ROLE | DEVICE STATE | APP STATE | CONFIG GEN
REASON | PENDING CONFIGS | VTLB ID | LB ID |
+-------------+-----+-----------+-------+-------+--------------------+-------------
+------------------+-----------------------------------------------------+---------+-------+
| 10.20.246.1 | | SLX-1 | 64512 | Spine | provisioned | cfg in-sync | NA | NA | NA | 1
|
| 10.20.246.7 | | SLX | 65000 | Leaf | provisioning failed | cfg ready | IA,IU,MD,DA |
SYSP-C,MCT-C,MCT-PA,BGP-C,INTIP-C,EVPN-C,O-C | 2 | 1 |
| 10.20.246.8 | | slx-8 | 65000 | Leaf | provisioned | cfg in-sync | NA | NA | 2 | 1 |
+-------------+-----+-----------+-------+-------+--------------------+-------------
+------------------+-----------------------------------------------------+---------+-------+
FABRIC SETTING:
BGPLL - BGP Dynamic Peer Listen Limit, BGP-MD5 - BGP MD5 Password , BFD-RX - Bfd Rx
Timer, BFD-TX - Bfd Tx Timer, BFD-MULTIPLIER - Bfd multiplier,
```

```
BFD-ENABLE - Enable Bfd, BGP-MULTIHOP - BGP ebgp multihop, P2PLR - Point-to-Point Link
Range, MCTLR - MCT Link Range, LOIP - Loopback IP Range


CONFIG GEN REASON:
LA/LD - Link Add/Delete, IA/ID/IU - Interface Add/Delete/Update, PLC/PLD/PLU -
IPPrefixList Create/Delete/Update
MD/MU - MCT Delete/Update, OD/OU - Overlay Gateway Delete/Update, EU/ED - Evpn Delete/
Update, PC/PD/PU - RouterPim Create/Delete/Update
DD - Dependent Device Update, DA/DR - Device Add/ReAdd, ASN - Asn Update, SYS - System
Properties Update
MD5 - BGP MD5 Password, BGPU - Router BGP Update, BGPLL - BGP Listen Limit, POU - Port
Channel Update, NA - Not Applicable

PENDING CONFIGS:
MCT - MCT Cluster, O - Overlay Gateway, SYSP - System Properties, INTIP - Interface
IP, BGP - Router BGP
C/D/U - Create/Delete/Update, PA/PD - Port Add/Port Delete


efa fabric setting show --name fabric1 --advanced | grep -i "backup routing"
| Backup Routing Enable          | Yes                     |
| Backup Routing IPv4 Range      | 10.40.40.0/24           |
| Backup Routing IPv6 Range      | fd40:4040:4040:1::/120 |

efa tenant show
+------+---------+------------+-------------+-------------+-----------+-----------
+----------------------+
| Name |  Type   | VLAN Range | L2VNI Range | L3VNI Range | VRF Count | Enable BD
|        Ports         |
+------+---------+------------+-------------+-------------+-----------+-----------
+----------------------+
| ten1 | private |   11-20    |             |             |    10     |   false   |
10.20.246.16[0/1-10] |
|      |         |            |             |             |           |           |
10.20.246.15[0/1-10] |
+------+---------+------------+-------------+-------------+-----------+-----------
+----------------------+

efa tenant vrf show
+----------+--------+-------------+--------------------+-------------+----------
+-----------+-----------+--------------------+-------------+-------------+
|   Name   | Tenant | Routing Type | Centralized Routers | Redistribute | Max Path |
Local Asn | Enable GR |       State        |  Dev State  |  App State  |
+----------+--------+-------------+--------------------+-------------+----------
+-----------+-----------+--------------------+-------------+-------------+
| ten1vrf1 |  ten1  | distributed  |                    |  connected  |    8     |
5000    |   false   | vrf-device-created | provisioned | cfg-in-sync |
+----------+--------+-------------+--------------------+-------------+----------
+-----------+-----------+--------------------+-------------+-------------+

efa tenant epg show -detail
================================================================================================
================================================================================
Name          : ten1epg1
Tenant        : ten1
Type          : extension
State         :
Ports         : 10.20.246.15[0/1]
POs           :
Port Property : SwitchPort Mode     : trunk
              : Native Vlan Tagging : false
NW Policy     : Ctag Range          : 11
              : VRF                 : ten1vrf1
```

```
              : L3Vni               : 8192
+------+-------------------------+-------+---------+--------------+--------------
+----------------------+---------+----------------+--------------+-------------
+-------------+
| Ctag |          Ctag           | L2Vni | BD Name | Anycast IPv4 | Anycast IPv6
|       Local IP        | IPv6 ND |    IPv6 ND     |   IPv6 ND    |  Dev State  |  App
State  |
|      |       Description       |       |         |              |              |
[Device-IP->Local-IP] |   Mtu   | Managed Config | Other Config |
|             |
+------+-------------------------+-------+---------+--------------+--------------
+----------------------+---------+----------------+--------------+-------------
+-------------+
|  11  | Tenant L3 Extended VLAN |  11   |         | 10.0.11.1/24 |
|                      |         |     false      |    false     | provisioned | cfg-
in-sync |
+------+-------------------------+-------+---------+--------------+--------------
+----------------------+---------+----------------+--------------+-------------
+-------------+
================================================================================================
================================================================================================
==
```

2. Complete the following configuration on SLX:

```
L1# show running-config bridge-
domain 4095
bridge-domain 4095 p2mp
 description Tenant L3 Extended BR
BD
 pw-profile Tenant-profile
 router-interface Ve 8191
!

L1# show running-config interface
Ve 8191
interface Ve 8191
 vrf forwarding ten1vrf1
 ip address 10.40.40.252/31
 ipv6 address
fd40:4040:4040:1::fe/127
 no shutdown
!

L1# do show running-config router
bgp
router bgp
 local-as 4200000000
 capability as4-enable
 fast-external-fallover
 neighbor 10.20.20.3 remote-as
4200000000
 neighbor 10.20.20.3 next-hop-self
 address-family ipv4 unicast
  network 172.31.254.203/32
  network 172.31.254.226/32
  maximum-paths 8
  graceful-restart
 !
 address-family ipv4 unicast vrf
ten1vrf1
  local-as 5000
  redistribute connected
  neighbor 10.40.40.253 remote-as
5000
  neighbor 10.40.40.253 next-hop-
self
  maximum-paths 8
 !
 address-family ipv6 unicast vrf
ten1vrf1
  redistribute connected
  neighbor fd40:4040:4040:1::ff
remote-as 5000
  neighbor fd40:4040:4040:1::ff
next-hop-self
  neighbor fd40:4040:4040:1::ff
activate
  maximum-paths 8
 !
!
```

```
L1# show running-config bridge-
domain 4095
bridge-domain 4095 p2mp
 description Tenant L3 Extended BR
BD
 pw-profile Tenant-profile
 router-interface Ve 8191
!

L1# show running-config interface
Ve 8191
interface Ve 8191
 vrf forwarding ten1vrf1
 ip address 10.40.40.252/31
 ipv6 address
fd40:4040:4040:1::fe/127
 no shutdown
!

L1# do show running-config router
bgp
router bgp
 local-as 4200000000
 capability as4-enable
 fast-external-fallover
 neighbor 10.20.20.3 remote-as
4200000000
 neighbor 10.20.20.3 next-hop-self
 address-family ipv4 unicast
  network 172.31.254.203/32
  network 172.31.254.226/32
  maximum-paths 8
  graceful-restart
 !
 address-family ipv4 unicast vrf
ten1vrf1
  local-as 5000
  redistribute connected
  neighbor 10.40.40.253 remote-as
5000
  neighbor 10.40.40.253 next-hop-
self
  maximum-paths 8
 !
 address-family ipv6 unicast vrf
ten1vrf1
  redistribute connected
  neighbor fd40:4040:4040:1::ff
remote-as 5000
  neighbor fd40:4040:4040:1::ff
next-hop-self
  neighbor fd40:4040:4040:1::ff
activate
  maximum-paths 8
 !
!
```

Configure Local ASN During VRF Update

1. Update a tenant VRF using the **efa tenant vrf update** command.

```
efa tenant vrf update --name ten1vrf1 --tenant ten1 --operation local-asn-add --local-
asn 6000
WARNING : This operation will result in the reset of the backup routing bgp neighbors
of the VRF. Do you want to proceed (Y/n)?

efa tenant vrf show
+----------+--------+-------------+--------------------+-------------+----------
+-----------+-----------+--------------------+-------------+-------------+
|   Name   | Tenant | Routing Type | Centralized Routers | Redistribute | Max Path |
Local Asn | Enable GR |       State        |  Dev State  |  App State  |
+----------+--------+-------------+--------------------+-------------+----------
+-----------+-----------+--------------------+-------------+-------------+
| ten1vrf1 |  ten1  | distributed  |                    |  connected  |    8     |
6000    |   false   | vrf-device-created | provisioned | cfg-in-sync |
+----------+--------+-------------+--------------------+-------------+----------
+-----------+-----------+--------------------+-------------+-------------+
```

2. Complete the following configuration on SLX:

```
L1# do show running-config router
bgp
router bgp
 local-as 4200000000
 capability as4-enable
 fast-external-fallover
 neighbor 10.20.20.3 remote-as
4200000000
 neighbor 10.20.20.3 next-hop-self
 address-family ipv4 unicast
  network 172.31.254.203/32
  network 172.31.254.226/32
  maximum-paths 8
  graceful-restart
 !
 address-family ipv4 unicast vrf
ten1vrf1
  local-as 6000
  redistribute connected
  neighbor 10.40.40.253 remote-as
6000
  neighbor 10.40.40.253 next-hop-
self
  maximum-paths 8
 !
 address-family ipv6 unicast vrf
ten1vrf1
  redistribute connected
  neighbor fd40:4040:4040:1::ff
remote-as 6000
  neighbor fd40:4040:4040:1::ff
next-hop-self
  neighbor fd40:4040:4040:1::ff
activate
  maximum-paths 8
 !
!
```

```
L2# show running-config router bgp
router bgp
 local-as 4200000000
 capability as4-enable
 fast-external-fallover
 neighbor 10.20.20.2 remote-as
4200000000
 neighbor 10.20.20.2 next-hop-self
 address-family ipv4 unicast
  network 172.31.254.226/32
  network 172.31.254.243/32
  maximum-paths 8
  graceful-restart
 !
 address-family ipv4 unicast vrf
ten1vrf1
  local-as 6000
  redistribute connected
  neighbor 10.40.40.252 remote-as
6000
  neighbor 10.40.40.252 next-hop-
self
  maximum-paths 8
 !
 address-family ipv6 unicast vrf
ten1vrf1
  redistribute connected
  neighbor fd40:4040:4040:1::fe
remote-as 6000
  neighbor fd40:4040:4040:1::fe
next-hop-self
  neighbor fd40:4040:4040:1::fe
activate
  maximum-paths 8
 !
```

Deconfigure Local ASN during VRF Update

1. Update a tenant VRF using the **efa tenant vrf update** command.

```
efa tenant vrf update --name ten1vrf1 --tenant ten1 --operation local-asn-delete
WARNING : This operation will result in the reset of the backup routing bgp neighbors
of the VRF. Do you want to proceed (Y/n)?


efa tenant vrf show
+----------+--------+--------------+---------------------+--------------+----------
+-----------+-----------+--------------------+-------------+--------------+
|   Name   | Tenant | Routing Type | Centralized Routers | Redistribute | Max Path |
Local Asn | Enable GR |       State        |  Dev State  |  App State  |
+----------+--------+--------------+---------------------+--------------+----------
+-----------+-----------+--------------------+-------------+--------------+
| ten1vrf1 |  ten1  | distributed  |                     |  connected   |    8
|           |   false   | vrf-device-created | provisioned | cfg-in-sync |
+----------+--------+--------------+---------------------+--------------+----------
+-----------+-----------+--------------------+-------------+--------------+
```

2. Complete the following configuration on SLX:

```
L1# do show running-config router
bgp
router bgp
 local-as 4200000000
 capability as4-enable
 fast-external-fallover
 neighbor 10.20.20.3 remote-as
4200000000
 neighbor 10.20.20.3 next-hop-self
 address-family ipv4 unicast
  network 172.31.254.203/32
  network 172.31.254.226/32
  maximum-paths 8
  graceful-restart
 !
 address-family ipv4 unicast vrf
ten1vrf1
  redistribute connected
  neighbor 10.40.40.253 remote-as
4200000000
  neighbor 10.40.40.253 next-hop-
self
  maximum-paths 8
 !
 address-family ipv6 unicast vrf
ten1vrf1
  redistribute connected
  neighbor fd40:4040:4040:1::ff
remote-as 4200000000
  neighbor fd40:4040:4040:1::ff
next-hop-self
  neighbor fd40:4040:4040:1::ff
activate
  maximum-paths 8
 !
!
```

```
L2# show running-config router bgp
router bgp
 local-as 4200000000
 capability as4-enable
 fast-external-fallover
 neighbor 10.20.20.2 remote-as
4200000000
 neighbor 10.20.20.2 next-hop-self
 address-family ipv4 unicast
  network 172.31.254.226/32
  network 172.31.254.243/32
  maximum-paths 8
  graceful-restart
 !
 address-family ipv4 unicast vrf
ten1vrf1
  redistribute connected
  neighbor 10.40.40.252 remote-as
4200000000
  neighbor 10.40.40.252 next-hop-
self
  maximum-paths 8
 !
 address-family ipv6 unicast vrf
ten1vrf1
  redistribute connected
  neighbor fd40:4040:4040:1::fe
remote-as 4200000000
  neighbor fd40:4040:4040:1::fe
next-hop-self
  neighbor fd40:4040:4040:1::fe
activate
  maximum-paths 8
 !
```

Enable Graceful Restart on Tenant VRF

The Tenant service configures graceful restart under each tenant VRF ipv4 or ipv6 unicast address-family.

You can enable graceful restart per tenant VRF during VRF creation and VRF update operations. EFA provisions the user input graceful restart on the switches during the instantiation and update of VRF on the switches.

- VRF is instantiated on the switches during a user-triggered L3 EPG create or L2 EPG transition to L3 EPG, based on the endpoints present in the EPG.
- VRF is updated on the switches during a user-triggered VRF update operation, based on the endpoints present in the EPGs, using this VRF.

Currently, the enable or disable value of graceful restart depends on the per-user VRF configuration.

Note

- By default, graceful restart is disabled. The default value of graceful restart is the switch default.
- EFA versions earlier than v2.4 have graceful restart enabled by default.
- An upgrade from a version earlier than v2.4 to v2.4 will retain the graceful restart as enabled for all the VRFs that were configured before the upgrade.

1. To enable graceful restart when you create a tenant VRF (**`efa tenant vrf create`**), add the **`--graceful-restart`** parameter.

   For more information, see the **`efa tenant vrf create`** command in the Extreme Fabric Automation Command Reference, 2.4.0.
2. To enable graceful restart on an existing tenant VRF (**`efa tenant vrf update`**), add the **`graceful-restart-update`** option for the **`--operation`** parameter and **`--graceful-restart`** parameter.

   For more information, see the **`efa tenant vrf update`** command in the Extreme Fabric Automation Command Reference, 2.4.0.

Configure Resilient Hashing on Tenant VRF

As a load-balancing method, resilient hashing helps to lessen the possibility that a destination path is remapped when a LAG (Link Aggregation Group) link fails.

When you create or update a tenant VRF, you can enable ECMP resilient hashing and you can configure the maximum number of resilient hashing paths allowed (8, 16, or 64 paths). Resilient hashing is disabled by default. The default number of allowed paths is the same as the default value for the SLX devices.

1. To enable resilient hashing when you create a tenant VRF, use the **`efa tenant vrf create`** command with the **`--rh-ecmp-enable`** and **`--rh-max-path`** parameters, and set the maximum number of allowed paths. For example,

```
# efa tenant vrf create --name <vrf-name> --tenant <tenant-name>
--rh-ecmp-enable=true/false --rh-max-path <8 | 16 | 64>
```

2. To enable resilient hashing on an existing tenant VRF, use the **efa tenant vrf update** command with the **--rh-ecmp-enable** and **--rh-max-path** parameters, and set the maximum number of allowed paths. For example,

```
# efa tenant vrf update --name <vrf-name> --tenant <tenant-name>
--operation <rh-max-path-add | rh-max-path-delete | rh-ecmp-update>
--rh-ecmp-enable=true/false --rh-max-path <8 | 16 | 64>
```

Note

- The **--max-path** and **--rh-max-path** parameters can co-exist.
- You cannot choose the specific devices on which to configure resilient hashing. Configuration applies to all SLX devices in the tenant VRF.
- For more information about the commands, including usage examples, see the *Extreme Fabric Automation Command Reference, 2.7.0*.

Enable Graceful Restart on Tenant VRF

You can enable graceful restart per tenant VRF during VRF create and VRF update operations. EFA provisions the user input graceful restart on the switches during the instantiation and update of VRF.

Based on the endpoints present in the EPG, VRF is instantiated on the switches when you trigger L3 endpoint group create or L2 endpoint group transition to L3 endpoint group.

VRF is updated on the switches when you trigger VRF update operation, based on the endpoints present in the endpoint groups using this VRF.

Currently, graceful restart is automatically configured when you configure the VRF.

Note

1. By default, graceful restart is disabled. Default value of graceful restart is the switch default.
2. Pre-EFA 2.4.0 versions have graceful restart enabled by default.
3. Upgrade from pre-2.4.0 to 2.4.x, and 2.5.0 retains the graceful restart enabled for all the pre-2.4.0 VRFs.

1. To enable graceful restart when you create a tenant VRF, run the following command:

```
efa tenant vrf create --name <vrf-name> --tenant <tenant-name>
            --graceful-restart=true/false
```

2. To enable graceful restart on an existing tenant VRF, run the following command:

```
efa tenant vrf update --name <vrf-name> --tenant <tenant-name>
            --operation graceful-restart-update --graceful-restart=true/false
```

Example

```
efa tenant vrf create --name vrf1 --tenant tenant1
efa tenant vrf create -name vrf10 -tenant tenant1 --graceful-restart=true

efa tenant epg create --name ten1epg1 --tenant tenant1
           --port 10.24.80.134[0/11],10.24.80.135[0/11]
           --switchport-mode trunk -ctag-range 11 --vrf vrf1 -anycast-ip
```

```
11:10.10.11.1/24
efa tenant epg create --name ten1epg2 --tenant tenant1
           --port 10.24.80.134[0/12],10.24.80.135[0/12]
           --switchport-mode trunk -ctag-range 12 --vrf vrf10 -anycast-ip
12:10.10.12.1/24
efa tenant vrf update --name vrf1 --tenant tenant1
           --operation graceful-restart-update --graceful-restart=true
efa tenant vrf update --name vrf10 --tenant tenant1
           --operation graceful-restart-update -graceful-restart=false
```

Configure MaxPaths on Tenant VRF

EFA allows provisioning of maximum-paths per tenant VRF during VRF create and VRF update operations.

VRF is updated on the switches when you trigger VRF update operation based on the endpoints present in the endpoint groups using the VRF.

Note

- Default value of `max-path` is 8.
- Choosing specific devices for `max-path` provisioning is not allowed.

To configure MaxPaths when you create a tenant VRF, run the following commands:

```
# efa tenant vrf create --name <vrf-name> --tenant <tenant-name> --max-path <value>

# efa tenant vrf update --name <vrf-name> --tenant <tenant-name> --operation <max-path-
add|max-path-delete> --max-path <value>

# efa tenant vrf create --name vrf1 --tenant tenant1

# efa tenant vrf create -name vrf10 -tenant tenant1

# efa tenant epg create --name ten1epg1 --tenant tenant1 --
port10.24.80.134[0/11],10.24.80.135[0/11] --switchport-mode trunk -ctag-range 11 --vrf
vrf1 --anycast-ip 11:10.10.11.1/24

# efa tenant epg create --name ten1epg2 --tenant tenant1 --
port10.24.80.134[0/12],10.24.80.135[0/12] --switchport-mode trunk -ctag-range 12 --vrf
vrf10 --anycast-ip 12:10.10.12.1/24

# efa tenant vrf update --name vrf10 -tenant tenant1 --operation max-paths-add --max-path
13 Switch Config
```

Configure Redistribute Attribute on Tenant VRF

EFA allows provisioning of redistribute attribute per tenant VRF during VRF create and VRF update operations.

VRF is updated on the switches during the user triggered VRF update operation based on the endpoints present in the endpoint groups using the VRF.

Note

- Default value of `redistribute` is `connected`.
- Choosing specific devices for `redistribute` provisioning is not allowed.

1. To configure redistribute attribute when you create a tenant VRF, run the following commands:

```
# efa tenant vrf create --name <vrf-name> --tenant <tenant-name> --redistribute <list>
```

2. To configure redistribute attribute on an existing tenant VRF, run the following commands:

```
# efa tenant vrf update --name <vrf-name> --tenant <tenant-name> --operation
<redistribute-add|redistribute-delete> --redistribute <static | connected>
```

Example

```
# efa tenant vrf create --name vrf1 --tenant tenant1 --redistribute static

# efa tenant vrf create –name vrf10 –tenant tenant1

# efa tenant epg create --name ten1epg1 --tenant tenant1 --
port10.24.80.134[0/11],10.24.80.135[0/11] --switchport-mode trunk –ctag-range 11 --vrf
vrf1 --anycast-ip 11:10.10.11.1/24

# efa tenant epg create --name ten1epg2 --tenant tenant1 --
port10.24.80.134[0/12],10.24.80.135[0/12] --switchport-mode trunk --ctag-range 12 --
vrf vrf10 --anycast-ip 12:10.10.12.1/24

# efa tenant vrf update --name vrf10 –tenant tenant1 --operation redistribute-add --
redistribute static

Device1# sh run router bgp
router bgp
local-as 4200000000
address-family ipv4unicast vrf vrf1
redistribute static
!
address-family ipv4unicast vrf vrf10
redistribute connected
redistribute static
!

Device2# sh run router bgp
router bgp
local-as 4200000000
address-family ipv4unicast vrf vrf1
redistribute static
!
address-family ipv4unicast vrf vrf10
redistribute connected
redistribute static
!
address-family ipv6unicast vrf vrf1
redistribute static
!
!address-family ipv6unicast vrf vrf10
redistribute connected
```

```
redistribute static
!
```

Configure Advertise Network and Static-Network on Tenant VRF

You can configure “network” and “static-network” attributes (to be advertised by BGP) on a tenant VRF (and device) when you create and update VRF. EFA provisions the “network” and “static-network” attributes on switches when you initiate or update VRF the switches. EFA supports only static-network with IPv4.

1. To configure “network” and “static-network” when you create a tenant VRF, run the following commands:

   When you create L3 EPG or transition L2 EPG to L3 EPG, VRF gets instantiated on the switches based on the endpoints present in the EPG.

```
efa tenant vrf create --name <vrf-name> --tenant <tenant-name>

                  --ipv4-network <device-ip,network> --ipv4-network-backdoor < device-
ip,network,true|false>
                  --ipv4-network-weight <device-ip,network,0-65535> --ipv4-network-
route-map <device-ip,network,route-map>

                  --ipv4-static-network <device-ip,static-network> --ipv4-static-
network-distance <device-ip,static-network,1-255>

                  --ipv6-network <device-ip,network> --ipv6-network-backdoor <device-
ip,network,true|false>
                  --ipv6-network-weight <device-ip,network,0-65535> --ipv6-network-
route-map <device-ip,network,route-map>
```

2. To configure “network” and “static-network” when you update a tenant VRF, run the following commands:

   When you trigger VRF update operation, VRF gets updated on the switches based on the endpoints present in the EPGs using the VRF.

```
efa tenant vrf update --name <vrf-name> --tenant <tenant-name>

                  --operation network-add|network-delete|static-network-add|static-
network-delete

                  --ipv4-network <device-ip,network> --ipv4-network-backdoor < device-
ip,network,true|false>
                  --ipv4-network-weight <device-ip,network,0-65535> --ipv4-network-
route-map <device-ip,network,route-map>

                  --ipv4-static-network <device-ip,static-network> --ipv4-static-
network-distance <device-ip,static-network,1-255>

                  --ipv6-network <device-ip,network> --ipv6-network-backdoor <device-
ip,network,true|false>
                  --ipv6-network-weight <device-ip,network,0-65535> --ipv6-network-
route-map <device-ip,network,route-map>
```

Example

```
efa tenant vrf create --name vrf1 --tenant tenant1
           --ipv4-network 10.24.80.134,10.20.30.40/30
           --ipv4-network 10.24.80.134,10.21.30.40/30 --ipv4-network-backdoor
10.24.80.134,10.21.30.40/30,true

           --ipv4-static-network 10.24.80.134,11.10.30.40/30
```

```
           --ipv4-static-network 10.24.80.134,11.20.30.40/30 --ipv4-static-network-
distance 10.24.80.134,11.20.30.40/30,169

           --ipv6-network 10.24.80.135,11::22/128
           --ipv6-network 10.24.80.135,11::23/128 --ipv6-network-backdoor
10.24.80.134,11::23/128,true
           --ipv6-network 10.24.80.135,11::24/128 --ipv6-network-weight
10.24.80.134,11::24/128,144
           --ipv6-network 10.24.80.135,11::25/128 --ipv6-network-route-map
10.24.80.134,11::25/128,rmap1

efa tenant epg create --name ten1epg1 --tenant tenant1
           --port 10.24.80.134[0/11],10.24.80.135[0/11]
           --switchport-mode trunk -ctag-range 11 --vrf vrf1 -anycast-ip
11:10.10.11.1/24

efa tenant vrf update --name vrf1 --tenant tenant1
           --operation network-add
           --ipv4-network 10.24.80.134,10.22.30.40/30 --ipv4-network-weight
10.24.80.134,10.22.30.40/30,144
           --ipv4-network 10.24.80.134,10.23.30.40/30 --ipv4-network-route-map
10.24.80.134,10.23.30.40/30,rmap1
```

3. Verify the switch configuration on the SLX device.

```
Rack1-Device1# sh run router bgp address-
family ipv4 unicast vrf vrf1

router bgp
  address-family ipv4 unicast vrf vrf1
    redistribute connected
    static-network 11.10.30.40/30
    static-network 11.20.30.40/30
distance 169
    network 10.20.30.40/30
    network 10.21.30.40/30 backdoor
    network 10.22.30.40/30 weight 144
    network 10.23.30.40/30 route-map rmap1
  !
  address-family ipv6 unicast vrf vrf1
    redistribute connected
  !
!
```

```
Rack1-Device2# sh run router bgp address-
family ipv4 unicast vrf vrf1

router bgp
  address-family ipv4 unicast vrf vrf1
    redistribute connected
  !
  address-family ipv6 unicast vrf vrf1
    redistribute connected
    network 11::22/128
    network 11::23/128 backdoor
    network 11::24/128 weight 144
    network 11::25/128 route-map rmap1
  !
!
```

Configure Aggregate Address on Tenant VRF

You can configure “aggregate-address” (to be advertised by BGP) per tenant VRF (and device) during VRF create and VRF update operations. EFA provisions the “aggregate-address” on switches when VRF is instantiated or updated on the switches.

1. To configure “aggregate-address” when you create a tenant VRF, run the following commands:

   When you trigger L3 EPG create or L2 EPG transition to L3 EPG, VRF is instantiated on the switches based on the endpoints present in the EPG.

```
efa tenant vrf create --name <vrf-name> --tenant <tenant-name>

                  --ipv4-aggregate-address <device-ip,aggregate-address>
                  --ipv4-aggregate-summary-only <device-ip,aggregate-address,true|false>
                  --ipv4-aggregate-as-set <device-ip,aggregate-address,true|false>
                  --ipv4-aggregate-advertise-map <device-ip,aggregate-address,route-map>
                  --ipv4-aggregate-suppress-map <device-ip,aggregate-address,route-map>
```

2. To configure “aggregate-address” when you update a tenant VRF, run the following commands:

   When you trigger VRF update operation, VRF is updated on the switches based on the endpoints present in the EPGs using the VRF.

```
efa tenant vrf update --name <vrf-name> --tenant <tenant-name>

                  --operation aggregate-address-add| aggregate-address-delete

                  --ipv4-aggregate-address <device-ip,aggregate-address>
                  --ipv4-aggregate-summary-only <device-ip,aggregate-address,true|false>
                  --ipv4-aggregate-as-set <device-ip,aggregate-address,true|false>
                  --ipv4-aggregate-advertise-map <device-ip,aggregate-address,route-map>
                  --ipv4-aggregate-suppress-map <device-ip,aggregate-address,route-map>
```

   Example

```
efa tenant vrf create --name vrf1 --tenant tenant1

            --ipv4-aggregate-address 10.24.80.134,10.20.21.40/30
            --ipv4-aggregate-summary-only 10.24.80.134,10.20.21.40/30,true
            --ipv4-aggregate-as-set 10.24.80.134,10.20.21.40/30,true
            --ipv4-aggregate-advertise-map 10.24.80.134,10.20.21.40/30,some
            --ipv4-aggregate-suppress-map 10.24.80.134,10.20.21.40/30,some

            --ipv6-aggregate-address 10.24.80.135,10::20/126
            --ipv6-aggregate-summary-only 10.24.80.135,10::20/126,true
            --ipv6-aggregate-as-set 10.24.80.135,10::20/126,true
            --ipv6-aggregate-advertise-map 10.24.80.135,10::20/126,some
            --ipv6-aggregate-suppress-map 10.24.80.135,10::20/126,some

efa tenant epg create --name ten1epg1 --tenant tenant1
            --port 10.24.80.134[0/11],10.24.80.135[0/11]
            --switchport-mode trunk -ctag-range 11 --vrf vrf1 -anycast-ip
11:10.10.11.1/24

efa tenant vrf update --name vrf1 --tenant tenant1
            --operation aggregate-address-add

            --ipv4-aggregate-address 10.24.80.134,10.21.21.40/30
            --ipv4-aggregate-summary-only 10.24.80.134,10.21.21.40/30,true
            --ipv4-aggregate-as-set 10.24.80.134,10.21.21.40/30,true
            --ipv4-aggregate-advertise-map 10.24.80.134,10.21.21.40/30,some
            --ipv4-aggregate-suppress-map 10.24.80.134,10.21.21.40/30,some

            --ipv6-aggregate-address 10.24.80.135,11::20/126
            --ipv6-aggregate-summary-only 10.24.80.135,11::20/126,true
            --ipv6-aggregate-as-set 10.24.80.135,11::20/126,true
            --ipv6-aggregate-advertise-map 10.24.80.135,11::20/126,some
            --ipv6-aggregate-suppress-map 10.24.80.135,11::20/126,some
```

3. Verify the switch configuration on the SLX device.

```
Rack1-Device1# sh run router bgp address-
family ipv4 unicast vrf vrf1

router bgp
  address-family ipv4 unicast vrf vrf1
    redistribute connected
    aggregate-address 10.20.21.40/30
advertise-map some
    aggregate-address 10.20.21.40/30 as-
set
    aggregate-address 10.20.21.40/30
summary-only
```

```
Rack1-Device2# sh run router bgp address-
family ipv4 unicast vrf vrf1

router bgp
  address-family ipv4 unicast vrf vrf1
    redistribute connected
  !
  address-family ipv6 unicast vrf vrf1
    redistribute connected
    aggregate-address 10::20/126
advertise-map some
    aggregate-address 10::20/126 as-set
```

```
    aggregate-address 10.20.21.40/30
suppress-map some
    aggregate-address 10.21.21.40/30
advertise-map some
    aggregate-address 10.21.21.40/30 as-
set
    aggregate-address 10.21.21.40/30
summary-only
    aggregate-address 10.21.21.40/30
suppress-map some
  !
  address-family ipv6 unicast vrf vrf1
    redistribute connected
  !
!
```

```
    aggregate-address 10::20/126 summary-
only
    aggregate-address 10::20/126 suppress-
map some
    aggregate-address 11::20/126
advertise-map some
    aggregate-address 11::20/126 as-set
    aggregate-address 11::20/126 summary-
only
    aggregate-address 11::20/126 suppress-
map some
  !
!
```

EPG: Network Property Description

The EPG (endpoint group) Network Property Description enables you to configure “description” per EFA tenant ctag which gets configured on the SLX as VLAN or BD description.

The default value of “description” is provided in the following table:

Network Type	Description
L2 Extension	Tenant L2 Extended VLAN or BD
L3 Extension	Tenant L3 Extended VLAN or BD
L3 Handoff	Tenant L3 Hand-off VLAN or BD
L3 Extension EVPN IRB	Tenant L3 Extended IRB BD

You can provide the EPG network “description” during EPG create and EPG update (ctag-range-add) operations.

2.3.2 to 2.4.0 Upgrade Handling

During the upgrade from EFA 2.3.2 to EFA 2.4.x, and 2.5.0 for all the EPG networks, the default “description” is determined and the app-state is set to “cfg-refreshed” so that the user triggers DRC (in EFA 2.4.0) to push the derived network “description” on to the SLX.

EPG Create

```
efa tenant epg create --name <epg-name> --tenant string <tenant-name> --port <list-of-
phy> --po <list-of-po>
               --switchport-mode <access |trunk | trunk-no-default-native>
               --ctag-range string <ctag-range> --ctag-description <ctag:description>
```

EPG Update

```
efa tenant epg update --name <epg-name> --tenant <tenant-name> --operation <ctag-range-
add>
                --ctag-range <ctag-range> --ctag-description <ctag:description>
```

Example

```
efa tenant show
+----------+---------+------------+-------------+-------------+-----------+-----------
+------------------------+
|   Name   |  Type   | VLAN Range | L2VNI Range | L3VNI Range | VRF Count | Enable BD
|         Ports          |
+----------+---------+------------+-------------+-------------+-----------+-----------
+------------------------+
|  bdTen1  | private |   21-30    |             |             |    10     |   true    |
10.20.246.15[0/11-20] |
|          |         |            |             |             |           |           |
10.20.246.16[0/11-20] |
+----------+---------+------------+-------------+-------------+-----------+-----------
+------------------------+
| vlanTen1 | private |   11-20    |             |             |    10     |   false   |
10.20.246.16[0/1-10]  |
|          |         |            |             |             |           |           |
10.20.246.15[0/1-10]  |
+----------+---------+------------+-------------+-------------+-----------+-----------
+------------------------+
efa tenant vrf create -name ten1vrf1 -tenant vlanTen1

efa tenant vrf create -name ten2vrf2 -tenant bdTen1
```

VLAN Based L2 Extension EPG

```
efa tenant epg create --name ten1epg1 --tenant vlanTen1 --port
10.20.246.15[0/1],10.20.246.16[0/1]
--switchport-mode trunk --ctag-range 11-12 --ctag-description 12:Ten1VLANNW1
```

VLAN Based L3 Extension EPG

```
efa tenant epg create --name ten1epg2 --tenant vlanTen1 --port
10.20.246.15[0/1],10.20.246.16[0/1]
--switchport-mode trunk --ctag-range 13-14 --ctag-description 14:Ten1VLANNW2 --anycast-ip
13:10.0.13.1/24 --anycast-ip 14:10.0.14.1/24 --vrf ten1vrf1
```

VLAN Based L3 Handoff EPG

```
efa tenant epg create --name ten1epg3 --tenant vlanTen1 --type l3-hand-off
--port 10.20.246.15[0/1],10.20.246.16[0/1] --switchport-mode trunk --ctag-range 15-16 --
ctag-description 16:Ten1VLANNW3
```

BD Based L2 Extension EPG

```
efa tenant epg create --name ten2epg1 --tenant bdTen1 --port
10.20.246.15[0/11],10.20.246.16[0/11]
--switchport-mode trunk --ctag-range 21-22 --ctag-description 22:Ten2BDNW1
```

BD Based L3 Extension EPG

```
efa tenant epg create --name ten2epg2 --tenant bdTen1 --port
10.20.246.15[0/11],10.20.246.16[0/11]
--switchport-mode trunk --ctag-range 23-24 --ctag-description 24:Ten2BDNW2 --anycast-ip
23:10.0.23.1/24 --anycast-ip 24:10.0.24.1/24 --vrf ten2vrf
```

efa tenant epg show --detail

Switch Config

```
Rack1-Device1# show running-
config bridge-domain
bridge-omain 1 p2mp
 description Tenant L2 Extended BD
 pw-profile default
 logical-interface ethernet 0/11.21
 bpdu-drop-enable
 local-switching
!
bridge-domain 2 p2mp
 description Ten2BDNW1
 pw-profile default
 logical-interface ethernet 0/11.22
 bpdu-drop-enable
 local-switching
!
bridge-domain 3 p2mp
 description Tenant L3 Extended BD
 pw-profile Tenant-profile
 router-interface Ve 4099
 !
 logical-interface ethernet 0/11.23
 bpdu-drop-enable
 local-switching
 suppress-arp
!
bridge-domain 4 p2mp
 description Ten2BDNW2
 pw-profile Tenant-profile
 router-interface Ve 4100
 !
 logical-interface ethernet 0/11.24
 bpdu-drop-enable
 local-switching
 suppress-arp
!
bridge-domain 4093 p2mp
 description Tenant L3 Extended IRB BD
 pw-profile Tenant-profile
 router-interface Ve 8189
 !
 bpdu-drop-enable
 local-switching
!
bridge-domain 4094 p2mp
 description Tenant L3 Extended IRB BD
 pw-profile Tenant-profile
 router-interface Ve 8190
 !
 bpdu-drop-enable
 local-switching
!
```

```
Rack1-Device2# show running-
config bridge-domain
bridge-domain 1 p2mp
 description Tenant L2 Extended BD
 pw-profile default
 logical-interface ethernet 0/11.21
 bpdu-drop-enable
 local-switching
!
bridge-domain 2 p2mp
 description Ten2BDNW1
 pw-profile default
 logical-interface ethernet 0/11.22
 bpdu-drop-enable
 local-switching
!
bridge-domain 3 p2mp
 description Tenant L3 Extended BD
 pw-profile Tenant-profile
 router-interface Ve 4099
 !
 logical-interface ethernet 0/11.23
 bpdu-drop-enable
 local-switching
 suppress-arp
!
bridge-domain 4 p2mp
 description Ten2BDNW2
 pw-profile Tenant-profile
 router-interface Ve 4100
 !
 logical-interface ethernet 0/11.24
 bpdu-drop-enable
 local-switching
 suppress-arp
!
bridge-domain 4093 p2mp
 description Tenant L3 Extended IRB BD
 pw-profile Tenant-profile
 router-interface Ve 8189
 !
 bpdu-drop-enable
 local-switching
!
bridge-domain 4094 p2mp
 description Tenant L3 Extended IRB BD
 pw-profile Tenant-profile
 router-interface Ve 8190
 !
 bpdu-drop-enable
 local-switching
!
```

```
Rack1-Device1#show running-config vlan
vlan 11
 description Tenant L2 Extended VLAN
!
vlan 12
 description Ten1VLANNW1
!
```

```
Rack1-Device2# show running-config vlan
vlan 11
 description Tenant L2 Extended VLAN
!
vlan 12
 description Ten1VLANNW1
!
```

```
vlan 13
 router-interface Ve 13
 suppress-arp
 description Tenant L3 Extended VLAN
!
vlan 14
 router-interface Ve 14
 suppress-arp
 description Ten1VLANNW2
!
vlan 15
 description Tenant L3 Hand-off VLAN
!
vlan 16
 description Ten1VLANNW3
!

Rack1-Device1#
```

```
vlan 13
 router-interface Ve 13
 suppress-arp
 description Tenant L3 Extended VLAN
!
vlan 14
 router-interface Ve 14
 suppress-arp
 description Ten1VLANNW2
!
vlan 15
 description Tenant L3 Hand-off VLAN
!
vlan 16
 description Ten1VLANNW3
!
Rack1-Device2#
```

Update anycast-ip on an Existing Tenant Network

You can add or delete anycast-ipv4 and anycast-ipv6 to or from an existing tenant network using the endpoint group (EPG) update anycast-ip-add or delete operations on an L3 EPG.

- You can provide ipv6 nd attributes (ipv6 nd mtu, ipv6 nd prefix, ipv6 nd m or o flags) along with anycast-ipv6 during anycast-ip-add operation.
- You can provide only one anycast-ipv4 and one anycast-ipv6 per tenant network even though SLX supports multiple anycast-ipv4 or ipv6 per VE.

Typical usage of the API from Openstack Integration is as follows:

1. EPG create with a ctag.
2. EPG update port-group-add with endpoints (po or phy).
3. EPG update vrf-add with anycast-ipv4.
4. EPG update anycast-ip-add with anycast-ipv6.

1. Add anycast-ip on existing EPG tenant network.

```
efa tenant epg update --name <epg-name> --tenant <tenant-name>
                --operation <anycast-ip-add | anycast-ip-delete>
                --anycast-ip <ctag:anycast-ipv4> --anycast-ipv6 <ctag:anycast-ipv6>
```

Example

```
efa tenant epg show --detail
================================================================================
================================================================================
Name          : ten1epg1
Tenant        : vlanTen1
Description   :
Type          : extension
Ports         : 10.20.246.15[0/1]
              : 10.20.246.16[0/1]
POs           :
Port Property : switchport mode     : trunk
              : native-vlan-tagging : false
NW Policy     : ctag-range          : 11
```

```
              : vrf                : ten1vrf1
              : l3-vni             : 8188
Network Property [Flags : * - Native Vlan]
+------+--------+-------------+-------------+---------
+--------------------------------+-------------------------+-------------
+----------------------+--------------------+-------------+-------------+
| Ctag | L2-Vni | Anycast-IPv4 | Anycast-IPv6 | BD-name | Local IP (Device-IP->Local-
IP) |    Ctag-Description     | Mtu-IPv6-ND | ManagedConfig-IPv6-ND | OtherConfig-IPv6-
ND |  Dev-state  |  App-state  |
+------+--------+-------------+-------------+---------
+--------------------------------+-------------------------+-------------
+----------------------+--------------------+-------------+-------------+
| 11   | 11     | 10.0.11.1/24 |             |
|                                | Tenant L3 Extended VLAN |             |
False                 | False               | provisioned | cfg-in-sync |
+------+--------+-------------+-------------+---------
+--------------------------------+-------------------------+-------------
+----------------------+--------------------+-------------+-------------+
================================================================================================
================================================================================

efa tenant epg update --name ten1epg1 --tenant vlanTen1 --operation anycast-ip-add --
anycast-ipv6 11:10::1/123

efa tenant epg show --detail
================================================================================================
================================================================================
Name          : ten1epg1
Tenant        : vlanTen1
Description   :
Type          : extension
Ports         : 10.20.246.15[0/1]
              : 10.20.246.16[0/1]
POs           :
Port Property : switchport mode     : trunk
              : native-vlan-tagging : false
NW Policy     : ctag-range          : 11
              : vrf                 : ten1vrf1
              : l3-vni              : 8188
Network Property [Flags : * - Native Vlan]
+------+--------+-------------+-------------+---------
+--------------------------------+-------------------------+-------------
+----------------------+--------------------+-------------+-------------+
| Ctag | L2-Vni | Anycast-IPv4 | Anycast-IPv6 | BD-name | Local IP (Device-IP->Local-
IP) |    Ctag-Description     | Mtu-IPv6-ND | ManagedConfig-IPv6-ND | OtherConfig-IPv6-
ND |  Dev-state  |  App-state  |
+------+--------+-------------+-------------+---------
+--------------------------------+-------------------------+-------------
+----------------------+--------------------+-------------+-------------+
| 11   | 11     | 10.0.11.1/24 | 10::1/123    |
|                                | Tenant L3 Extended VLAN |             |
False                 | False               | provisioned | cfg-in-sync |
+------+--------+-------------+------+---------+---------------------------------
+-------------------------+-------------+----------------------+--------------------
+-------------+-------------+
================================================================================================
================================================================================
```

2. Delete anycast-ip on existing EPG tenant network.

```
efa tenant epg update --name ten1epg1 --tenant vlanTen1 --operation anycast-ip-delete
--anycast-ipv6 11:10::1/123

efa tenant epg show
================================================================================================
```

```
================================================================================
Name          : ten1epg1
Tenant        : vlanTen1
Description   :
Type          : extension
Ports         : 10.20.246.15[0/1]
              : 10.20.246.16[0/1]
POs           :
Port Property : switchport mode     : trunk
              : native-vlan-tagging : false
NW Policy     : ctag-range          : 11
              : vrf                 : ten1vrf1
              : l3-vni              : 8188
Network Property [Flags : * - Native Vlan]
+------+--------+-------------+-------------+---------
+---------------------------------+-------------+----------------------
+--------------------+-------------+-------------+
| Ctag | L2-Vni | Anycast-IPv4 | Anycast-IPv6 | BD-name | Local IP (Device-IP->Local-
IP) | Mtu-IPv6-ND | ManagedConfig-IPv6-ND | OtherConfig-IPv6-ND |  Dev-state  |  App-
state  |
+------+--------+-------------+-------------+---------
+---------------------------------+-------------+----------------------
+--------------------+-------------+-------------+
| 11   | 11     | 10.0.11.1/24 |              |
|                                 |             | False                |
False               | provisioned | cfg-in-sync |
+------+--------+-------------+-------------+---------
+---------------------------------+-------------+----------------------
+--------------------+-------------+-------------+
For 'unstable' entities, run 'efa tenant po/vrf show' for details
================================================================================
================================================================================
```

Configure Multiple Anycast IP

You can provide multiple anycast IP address per tenant ctag when you create or update an EPG, add ctag-range, add VRF, and add or delete anycast IP. The multiple anycast IP address is configured under the `interface Ve` of the switching hardware.

1. To configure multiple anycast IP when you create an EPG, run the following command:

```
efa tenant epg create --name <epg-name> --tenant <tenant-name>

           --switchport-mode <mode>
        --port <port-list> --po <po-list>

          --ctag-range <ctag-range>
          --vrf <vrf-name>

       --anycast-ip <value> --anycast-ip <value>
       --anycast-ipv6 <value> --anycast-ipv6 <value>
```

2. To configure multiple anycast IP when you update an EPG, run the following command:

```
efa tenant epg update --name <epg-name> --tenant <tenant-name>

              --operation vrf-add | ctag-range-add |anycast-ip-add | anycast-ip-
delete|
              --ctag-range <ctag-range> --vrf  <vrf-name>

              --anycast-ip <value> --anycast-ip <value>
              --anycast-ipv6 <value> --anycast-ipv6 <value>
```

Example:

Multiple anycast IP configuration example when you create an EPG:

```
efa tenant epg create --name e1 --tenant tenant11 --po po1 --switchport-mode trunk-no-
default-native --vrf v1 --ctag-range 101-102  --anycast-ip 101:1.1.1.254/24 --anycast-ip
101:2.1.1.254/24 --anycast-ip 101:3.1.1.254/24 --anycast-ipv6 101:1::1/124 --anycast-ipv6
102:2::1/124
efa tenant epg show -detail
================================================================================================
========================
Name          : e1
Tenant        : tenant11
Type          : extension
State         : epgf-with-port-group-and-ctag-range
Description   :

Ports         :
POs           : po1
Port Property : SwitchPort Mode               : trunk-no-default-native
              : Single-Homed BFD Session Type : Auto
NW Policy     : Ctag Range                    : 101-103
              : VRF                           : v1
              : L3Vni                         : 8192

+------------+-------------+-----------+------------+-------------+
| MAC ACL IN | MAC ACL OUP | IP ACL IN | IP ACL OUT | IPv6 ACL IN |
+------------+-------------+-----------+------------+-------------+
POrt Property ACLs

+------+--------------+-------------+
| Port |  Dev State   |  App State  |
+------+--------------+-------------+
| po1  | provisionied | cfg-in-sync |
+------+--------------+-------------+
Port Property States

+------+---------------+-------+---------+--------------+--------------+----------
+--------------------------+--------+---------+---------------+--------------
+-------------+-------------+
| Ctag |    Ctag       | L2Vni | BD Name | Anycast IPv4 | Anycast IPv6 | Suppress
|         Local IP         | IP MTU | IPv6 ND |    IPv6 ND     |     IPv6 ND  |  Dev
State  |  App State  |
|      |  Description  |       |         |              |              |  ARP/ND  |
[Device-IP->Local-IP]  |        |   MTU   | Managed Oonfig | Other Config |
|             |
+------+---------------+-------+---------+--------------+--------------+----------
+--------------------------+--------+---------+---------------+--------------
+-------------+-------------|
| 101  |   Tenant L3   |10000  |         | 2.1.1.254/24 |   1::1/124   |    T/T   |
10.20.246.3->1.10.1.1/24 |        |         |      false     |     false    | porvisioned
| cfg-in-sync |
|      | Extended VLAN |       |         | 3.1.1.254/24 |              |
|                          |        |         |                |
|             |             |
|      |               |       |         |              |              |
|                          |        |         |                |
|             |             |
|      |               |       |         | 1.1.1.254/24 |              |
|                          |        |         |                |
|             |             |
+------+---------------+-------+---------+--------------+--------------+----------
+--------------------------+--------+---------+---------------+--------------
+-------------+-------------|
```

```
| 102  |   Tenant L3   |10001  |         |              |   2::1/124   |    T/T
|                          |        |         |      false     |     false    |
porvisioned | cfg-in-sync |
|      | Extended VLAN |       |         |              |              |
|                          |        |         |                |
|             |             |
+------+---------------+-------+---------+--------------+--------------+----------
+--------------------------+--------+---------+----------------+--------------
+-------------+-------------+
Network Property [Flags : * - Native Vlan]

+------+----------------+--------------+----------------+--------------------
+-------------+
| Ctag | IPv6 ND Prefix | No Advertise | Valid Lifetime | Preferred Lifetime | Config
Type |
+------+----------------+--------------+----------------+--------------------
+-------------+
IPv6 ND Prefix Flags

+------+----------------+-------------+-----------+------------+-------------+
| Ctag | MAC ACL Prefix | MAC ACL OUT | IP ACL IN | IP ACL OUT | IPv6 ACL IN |
+------+----------------+-------------+-----------+------------+-------------+
Network Property ACLs

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

10.20.246.25

```
SLX# show running-config interface
Ve
interface Ve 101
 vrf forwarding v1
 ip anycast-address 1.1.1.254/24
 ip anycast-address 2.1.1.254/24
 ip anycast-address 3.1.1.254/24
 ipv6 anycast-address 1::1/124
 no shutdown
!
interface Ve 102
 vrf forwarding v1
 ipv6 anycast-address 2::1/124
 no shutdown
!
interface Ve 4090
 ip address 10.20.20.2/31
 no shutdown
!
interface Ve 5120
 vrf forwarding v1
 ipv6 address use-link-local-only
 no shutdown
!
```

10.20.246.26

```
SLX# show running-config interface
Ve
interface Ve 101
 vrf forwarding v1
 ip anycast-address 1.1.1.254/24
 ip anycast-address 2.1.1.254/24
 ip anycast-address 3.1.1.254/24
 ipv6 anycast-address 1::1/124
 no shutdown
!
interface Ve 102
 vrf forwarding v1
 ipv6 anycast-address 2::1/124
 no shutdown
!
interface Ve 4090
 ip address 10.20.20.3/31
 no shutdown
!
interface Ve 5120
 vrf forwarding v1
 ipv6 address use-link-local-only
 no shutdown
!
```

Multiple anycast IP configuration example when you update an EPG:

```
efa  tenant epg update --name e1 --tenant tenant11 --operation anycast-ip-add --anycast-ip 101:4.1.1.1/24
```

10.20.246.25	10.20.246.26
SLX# show running-config interface Ve **interface Ve 101** vrf forwarding v1 ip anycast-address 1.1.1.254/24 ip anycast-address 2.1.1.254/24 ip anycast-address 3.1.1.254/24 ip anycast-address 4.1.1.1/24 ipv6 anycast-address 1::1/124 no shutdown ! interface Ve 102 vrf forwarding v1 ipv6 anycast-address 2::1/124 no shutdown ! interface Ve 4090 ip address 10.20.20.2/31 no shutdown ! interface Ve 5120 vrf forwarding v1 ipv6 address use-link-local-only no shutdown !	**SLX# show running-config interface Ve** **interface Ve 101** vrf forwarding v1 ip anycast-address 1.1.1.254/24 ip anycast-address 2.1.1.254/24 ip anycast-address 3.1.1.254/24 ip anycast-address 4.1.1.1/24 ipv6 anycast-address 1::1/124 no shutdown ! interface Ve 102 vrf forwarding v1 ipv6 anycast-address 2::1/124 no shutdown ! interface Ve 4090 ip address 10.20.20.3/31 no shutdown ! interface Ve 5120 vrf forwarding v1 ipv6 address use-link-local-only no shutdown !

Configure IPv6 ND on a Tenant Network

The feature lets you configure ipv6 nd attributes per tenant network (ctag). IPv6 ND attributes are MTU, M flag, O flag and Prefixes.

You can configure IPv6 ND attributes using EPG (Endpoint Group) create or EPG update ctag-range-add, vrf-add, and anycast-ip-add operations.

1. Configure IPv6 ND during EPG Create.

```
efa tenant epg create --name <epg-name> --tenant <tenant-name>
                --ipv6-nd-mtu <ipv6-mtu>
                --ipv6-nd-managed-config <true | false>
                --ipv6-nd-other-config <true | false>
                --ipv6-nd-prefix <ctag:list-of-prefix>
                --ipv6-nd-prefix-valid-lifetime <ctag,prefix:validTime>
                --ipv6-nd-prefix-preferred-lifetime <ctag,prefix:preferredTime>
                --ipv6-nd-prefix-no-advertise <ctag,prefix:noadvertiseflag>
                --ipv6-nd-prefix-config-type <ctag,prefix:configType(no-autoconfig| no-onlink | off-link)>
```

2. Configure IPv6 ND during EPG Update.

```
efa tenant epg update --name <epg-name> --tenant <tenant-name>
                --operation <ctag-range-add | vrf-add | anycast-ip-add> --ctag-range <ctag-range> --vrf <vrf-name>
                --anycast-ip <ctag:anycast-ip> --anycast-ipv6 <ctag:anycast-ipv6>
                --ipv6-nd-mtu <ipv6-mtu>
                --ipv6-nd-managed-config <true | false>
                --ipv6-nd-other-config <true | false>
```

```
              --ipv6-nd-prefix <ctag:list-of-prefix>
              --ipv6-nd-prefix-valid-lifetime <ctag,prefix:validTime>
              --ipv6-nd-prefix-preferred-lifetime <ctag,prefix:preferredTime>
              --ipv6-nd-prefix-no-advertise <ctag,prefix:noadvertiseflag>
              --ipv6-nd-prefix-config-type <ctag,prefix:configType(no-autoconfig| no-
onlink | off-link)>
```

Example

```
efa tenant show
+----------+---------+------------+-------------+-------------+-----------+-----------
+----------------------+
|   Name   |  Type   | VLAN Range | L2VNI Range | L3VNI Range | VRF Count | Enable BD
|        Ports         |
+----------+---------+------------+-------------+-------------+-----------+-----------
+----------------------+
| vlanTen1 | private |   11-20    |             |             |    10     |   false
| 10.20.246.15[0/1-10] |
|          |         |            |             |             |           |
| 10.20.246.16[0/1-10] |
+----------+---------+------------+-------------+-------------+-----------+-----------
+----------------------+


efa tenant vrf show
+----------+----------+--------------+---------------------+--------------+----------
+-----------+-----------+--------------------+-------------+--------------+
|   Name   |  Tenant  | Routing Type | Centralized Routers | Redistribute | Max Path |
Local Asn | Enable GR |       State        |  Dev State  |  App State  |
+----------+----------+--------------+---------------------+--------------+----------
+-----------+-----------+--------------------+-------------+--------------+
| ten1vrf1 | vlanTen1 | distributed  |                     |  connected   |    8
|           |   false   | vrf-device-created | provisioned | cfg-in-sync |
+----------+----------+--------------+---------------------+--------------+----------
+-----------+-----------+--------------------+-------------+--------------+


efa tenant epg create --name ten1epg1 --tenant vlanTen1 --port
10.20.246.15[0/1],10.20.246.16[0/1] --switchport-mode trunk --ctag-range 11-13
          --anycast-ip 11:10.0.11.1/24 --anycast-ip 12:10.0.12.1/24 --anycast-ip
13:10.0.13.1/24
          --ipv6-nd-mtu 12:1600 --ipv6-nd-managed-config 12:true --ipv6-nd-other-
config 12:true
          --ipv6-nd-prefix 12:1:5::/64 --ipv6-nd-prefix-valid-lifetime
12,1:5::/64:2000 --ipv6-nd-prefix-preferred-lifetime 12,1:5::/64:2000
          --ipv6-nd-prefix 12:1:6::/64 --ipv6-nd-prefix-valid-lifetime
12,1:6::/64:2001 --ipv6-nd-prefix-preferred-lifetime 12,1:6::/64:2001
          --vrf ten1vrf1

efa tenant epg show
+----------+----------+-----------+--------------------+----+------------+-------------
+------------+----------+-------+-------+
|   Name   |  Tenant  |   Type    |       Ports        | PO | SwitchPort | Native Vlan
| Ctag Range |   Vrf    | L3Vni | State |
|          |          |           |                    |    |    Mode    |   Tagging
|            |          |       |       |
+----------+----------+-----------+--------------------+----+------------+-------------
+------------+----------+-------+-------+
| ten1epg1 | vlanTen1 | extension | 10.20.246.15[0/1] |    |   trunk    |    false
|   11-13    | ten1vrf1 | 8192  |       |
|          |          |           | 10.20.246.16[0/1] |    |            |
|            |          |       |       |
+----------+----------+-----------+--------------------+----+------------+-------------
+------------+----------+-------+-------+
```

```
efa tenant epg show --detail

================================================================================================
================================================================================
Name          : ten1epg1
Tenant        : vlanTen1
Type          : extension
State         :
Description   :
Ports         : 10.20.246.15[0/1]
              : 10.20.246.16[0/1]
POs           :
Port Property : SwitchPort Mode     : trunk
              : Native Vlan Tagging : false
NW Policy     : Ctag Range          : 11-13
              : VRF                 : ten1vrf1
              : L3Vni               : 8192
+------+-------------------------+-------+---------+--------------+--------------
+----------------------+---------+----------------+--------------+-------------
+-------------+
| Ctag |          Ctag           | L2Vni | BD Name | Anycast IPv4 | Anycast IPv6
|       Local IP        | IPv6 ND |    IPv6 ND     |   IPv6 ND    |  Dev State  |  App
State  |
|      |       Description       |       |         |              |              |
[Device-IP->Local-IP] |   Mtu   | Managed Config | Other Config |
|             |
+------+-------------------------+-------+---------+--------------+--------------
+----------------------+---------+----------------+--------------+-------------
+-------------+
|  11  | Tenant L3 Extended VLAN |  11   |         | 10.0.11.1/24 |
|                      |         |     false      |    false     | provisioned | cfg-
in-sync |
+------+-------------------------+-------+---------+--------------+--------------
+----------------------+---------+----------------+--------------+-------------
+-------------+
|  12  | Tenant L3 Extended VLAN |  12   |         | 10.0.12.1/24 |
|                      |  1600   |      true      |     true     | provisioned | cfg-
in-sync |
+------+-------------------------+-------+---------+--------------+--------------
+----------------------+---------+----------------+--------------+-------------
+-------------+
|  13  | Tenant L3 Extended VLAN |  13   |         | 10.0.13.1/24 |
|                      |         |     false      |    false     | provisioned | cfg-
in-sync |
+------+-------------------------+-------+---------+--------------+--------------
+----------------------+---------+----------------+--------------+-------------
+-------------+
Network Property [Flags : * - Native Vlan]
+------+----------------+--------------+----------------+--------------------
+-------------+
| Ctag | IPv6 ND Prefix | No Advertise | Valid Lifetime | Preferred Lifetime | Config
Type |
+------+----------------+--------------+----------------+--------------------
+-------------+
|  12  |    1:5::/64    |    false     |      2000      |        2000
|             |
+------+----------------+--------------+----------------+--------------------
+-------------+
|  12  |    1:6::/64    |    false     |      2001      |        2001
|             |
+------+----------------+--------------+----------------+--------------------
+-------------+
```

```
IPv6 ND Prefix Flags
For 'unstable' entities, run 'efa tenant po/vrf show' for details
================================================================================================
================================================================================
```

Configure the BFD Session Type for an Endpoint Group

You can determine the session type for a Bidirectional Forwarding Detection (BFD) session formed over a Cluster Edge Port (CEP) port.

You can assign a session type as you create an endpoint group. You can also assign a session type to an existing endpoint group.

The default is auto, which means that the BFD session type is automatically determined based on whether the service type is set to extension (software) or Layer 3 hand-off (hardware).

The value of `--single-homed-bfd-session-type` is configured for one endpoint group and then propagated to all Ethernet and single-homed port channel interfaces defined for that endpoint group.

EFA does not distinguish between SRIOV (single-root input/output virtualization) and non-SRIOV connections. Therefore, it treats both connections the same way. If you want to use hardware-based BFD sessions for CEP non-SRIOV connections, then create an endpoint group that contains all the CEP non-SRIOV connections and set the `--single-homed-bfd-session-type` to `hardware`.

1. To assign a BFD session type as you create an endpoint group, run the following command.

   ```
   $ efa tenant epg create --name epg5 --tenant tenant11 --port 10.20.216.15[0/11]
   ,10.20.216.16[0/11] --po po1 --switchport-mode trunk --single-homed-bfd-session-type
   auto
   ```

   In this example, the session type is set to 'auto'.
2. To assign a BFD session type to an existing endpoint group, run the following command.

   ```
   $ efa tenant epg update --name epg5 --tenant tenant11 --operation port-group-add
   --port 10.20.216.15[0/11],10.20.216.16[0/11] --po po1 --switchport-mode trunk
   --single-homed-bfd-session-type hardware
   ```

   In this example, the session type is set to 'hardware'.

Configure CEP Cluster Tracking for Endpoint Groups

EFA does not provision `reload-delay 90` configuration on Cluster Edge Port (CEP) interfaces. EFA instead provisions `cluster-track` configuration on CEP interfaces when the CEP is configured as a member of an EPG (endpoint group) during the creation or updating of endpoint groups.

During upgrade from EFA 2.3.0 to 2.4.0 or 2.5.0, EFA marks all the CEPs with the intended `cluster-track` configuration and shows it as configuration drift. On reconciliation of the drift, EFA pushes the `cluster-track` configuration to the CEP ports. Before the `cluster-track` configuration push, EFA automatically removes the `reload-delay` configuration from the CEP ports.

To configure CEP Cluster Tracking for EPG, run the following command:

```
# efa tenant po create --name po11 --tenant tenant1 –speed 10Gbps --negotiation active
--port 10.24.80.134[0/15] => CEP # efa tenant po create --name po12 --tenant tenant1
--speed 10Gbps --negotiation active --port 10.24.80.134[0/25],10.24.80.135[0/25] => CCEP

# efa tenant epg create --name ten1epg1 --tenant tenant1 --switchport-mode trunk --ctag-
```

```
range 1001
--port 10.24.80.134[0/35] --po po11,po12
```

Enable Cluster Tracking on CEP Interfaces

By default, EFA enables reload-delay configuration on all Cluster Edge Port (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 interfaces, you must remove the reload-delay configuration.

1. Remove the reload-delay configuration on CEP 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
```

Suppress Address Resolution Protocol and Neighbor Discovery on VLAN or Bridge Domain

Provide an option to enable or disable suppress-arp or nd (supress address resolution protocol or neighbor discovery) at the tenant network level so that you can choose to enable suppress-arp or nd per tenant network.

This option is mainly useful for a single rack small data center deployment where the suppress-arp or nd configuration on tenant network is not needed.

- EFA configures the suppress-arp or nd on the VLAN or BD (Bridge Domain) associated with the L3 tenant networks belonging to the distributed router.
- EFA does not configure the suppress-arp or nd on the VLAN or BD associated with the L3 tenant networks belonging to the centralized router.
- EFA doesn't configure the suppress-arp or nd on the VLAN or BD associated with the L2 tenant networks.

Note

- For all the L3 tenant networks belonging to the distributed router, suppress-arp or nd is displayed as "true" after the upgrade from 2.4.x to 2.5.0.
- For all the L3 tenant networks belonging to the centralized router, suppress-arp or nd is displayed as "false" after the upgrade from 2.4.x to 2.5.0.
- For all the L2 tenant networks, suppress-arp or nd is displayed as "false" after the upgrade from 2.4.x to 2.5.0.

To configure suppress ARP or ND for performance tuning when you create a VLAN or BD, run the following command:

```
efa tenant epg create --name <epg-name> --tenant <tenant-name>
                --port <port-list> --switchport-mode <trunk|access> --ctag-range <ctag-
```

```
range>
                --anycast-ip <ctag:anycast-ipv4> --anycast-ipv6 <ctag:anycast-ipv6> --
vrf ten1vrf1
                --suppress-arp <ctag:true|false> --suppress-nd <ctag:true|false>
 efa tenant epg update --name <epg-name> --tenant <tenant-name> --operation vrf-add
                 --anycast-ip <ctag:anycast-ipv4> --anycast-ipv6 <ctag:anycast-ipv6>  --
vrf ten1vrf1
                 --suppress-arp <ctag:true|false> --suppress-nd <ctag:true|false>
efa tenant epg update --name <epg-name> --tenant <tenant-name> --operation ctag-range-add
                --ctag-range <ctag-range> --anycast-ip <ctag:anycast-ipv4> --anycast-
ipv6 <ctag:anycast-ipv6>
                --suppress-arp <ctag:true|false> --suppress-nd <ctag:true|false>
 efa tenant epg update --name <epg-name> --tenant <tenant-name> --operation anycast-ip-
add
      --anycast-ip <ctag:anycast-ipv4> --anycast-ipv6 <ctag:anycast-ipv6>
                 --suppress-arp <ctag:true|false> --suppress-nd <ctag:true|false>
efa fabric show
Fabric Name: default, Fabric Description: Default Fabric, Fabric Stage: 3, Fabric Type:
clos, Fabric Status: created
+------------+-----+-----------+-----+------+--------------+-----------
+-------------------+-----------------+---------+-------+
| IP ADDRESS | POD | HOST NAME | ASN | ROLE | DEVICE STATE | APP STATE | CONFIG GEN
REASON | PENDING CONFIGS | VTLB ID | LB ID |
+------------+-----+-----------+-----+------+--------------+-----------
+-------------------+-----------------+---------+-------+
+------------+-----+-----------+-----+------+--------------+-----------
+-------------------+-----------------+---------+-------+

Fabric Name: fs, Fabric Description: , Fabric Stage: 3, Fabric Type: clos, Fabric Status:
settings-updated

Updated Fabric Settings: BGP-LL

+-------------+-----+-----------+-------+-------+---------------------+-------------
+-------------------+-----------------------------------------------------+---------+-------+
| IP ADDRESS | POD | HOST NAME | ASN | ROLE | DEVICE STATE | APP STATE | CONFIG GEN
REASON | PENDING CONFIGS | VTLB ID | LB ID |
+-------------+-----+-----------+-------+-------+---------------------+-------------
+-------------------+-----------------------------------------------------+---------+-------+
| 10.20.246.1 | | SLX-1 | 64512 | Spine | provisioned | cfg in-sync | NA | NA | NA | 1 |
| 10.20.246.7 | | SLX | 65000 | Leaf | provisioning failed | cfg ready | IA,IU,MD,DA |
SYSP-C,MCT-C,MCT-PA,BGP-C,INTIP-C,EVPN-C,O-C | 2 | 1 |
| 10.20.246.8 | | slx-8 | 65000 | Leaf | provisioned | cfg in-sync | NA | NA | 2 | 1 |
+-------------+-----+-----------+-------+-------+---------------------+-------------
+-------------------+-----------------------------------------------------+---------+-------+
FABRIC SETTING:
BGPLL - BGP Dynamic Peer Listen Limit, BGP-MD5 - BGP MD5 Password , BFD-RX - Bfd Rx
Timer, BFD-TX - Bfd Tx Timer, BFD-MULTIPLIER - Bfd multiplier,
BFD-ENABLE - Enable Bfd, BGP-MULTIHOP - BGP ebgp multihop, P2PLR - Point-to-Point Link
Range, MCTLR - MCT Link Range, LOIP - Loopback IP Range


CONFIG GEN REASON:
LA/LD - Link Add/Delete, IA/ID/IU - Interface Add/Delete/Update, PLC/PLD/PLU -
IPPrefixList Create/Delete/Update
MD/MU - MCT Delete/Update, OD/OU - Overlay Gateway Delete/Update, EU/ED - Evpn Delete/
Update, PC/PD/PU - RouterPim Create/Delete/Update
DD - Dependent Device Update, DA/DR - Device Add/ReAdd, ASN - Asn Update, SYS - System
Properties Update
MD5 - BGP MD5 Password, BGPU - Router BGP Update, BGPLL - BGP Listen Limit, POU - Port
Channel Update, NA - Not Applicable

PENDING CONFIGS:
MCT - MCT Cluster, O - Overlay Gateway, SYSP - System Properties, INTIP - Interface IP,
```

```
BGP - Router BGP
C/D/U - Create/Delete/Update, PA/PD - Port Add/Port Delete

efa tenant show
+------+---------+------------+-------------+-------------+-----------+-----------
+-----------------------+
| Name |  Type   | VLAN Range | L2VNI Range | L3VNI Range | VRF Count | Enable BD
|         Ports         |
+------+---------+------------+-------------+-------------+-----------+-----------
+-----------------------+
| ten1 | private |   11-20    |             |             |    10     |   false   |
10.20.246.15[0/1-10]  |
|      |         |            |             |             |           |           |
10.20.246.16[0/1-10]  |
+------+---------+------------+-------------+-------------+-----------+-----------
+-----------------------+
| ten2 | private |   21-30    |             |             |    10     |   true    |
10.20.246.15[0/11-20] |
|      |         |            |             |             |           |           |
10.20.246.16[0/11-20] |
+------+---------+------------+-------------+-------------+-----------+-----------
+-----------------------+
efa tenant vrf show
+----------+--------+--------------+---------------------+--------------+----------
+-----------+-----------+--------------------+-------------+-------------+
|   Name   | Tenant | Routing Type | Centralized Routers | Redistribute | Max Path |
Local Asn | Enable GR |       State        |  Dev State  |  App State  |
+----------+--------+--------------+---------------------+--------------+----------
+-----------+-----------+--------------------+-------------+-------------+
| ten1vrf1 |  ten1  | distributed  |                     |  connected   |    8
|           |   false   | vrf-device-created | provisioned | cfg-in-sync |
+----------+--------+--------------+---------------------+--------------+----------
+-----------+-----------+--------------------+-------------+-------------+
| ten2vrf1 |  ten2  | distributed  |                     |  connected   |    8
|           |   false   | vrf-device-created | provisioned | cfg-in-sync |
+----------+--------+--------------+---------------------+--------------+----------
+-----------+-----------+--------------------+-------------+-------------+
efa tenant epg create --name ten1epg1 --tenant ten1 --port
10.20.246.15[0/1],10.20.246.16[0/1]
           --switchport-mode trunk --ctag-range 11-12
           --anycast-ip 11:10.0.11.1/24 --anycast-ip 12:10.0.12.1/24 --anycast-ipv6
11:11::1/127 --anycast-ipv6 12:12::1/127 --vrf ten1vrf1
           --suppress-arp 11:false -suppress-nd 12:false

efa tenant epg create --name ten2epg1 --tenant ten2 --port
10.20.246.15[0/11],10.20.246.16[0/11]
           --switchport-mode trunk --ctag-range 21-22
           --anycast-ip 21:10.0.21.1/24 --anycast-ipv6 21:21::1/127 --anycast-ip
22:10.0.22.1/24 --anycast-ipv6 22:22::1/127 --vrf ten2vrf1
           --suppress-arp 21:false -suppress-nd 22:false

efa tenant epg show -detail

================================================================================================
=============================================================
Name          : ten1epg1
Tenant        : ten1
Type     : extension
State         :
Description   :
Ports         : 10.20.246.15[0/1]
              : 10.20.246.16[0/1]
POs           :
Port Property : SwitchPort Mode     : trunk
```

```
              : Native Vlan Tagging : false
NW Policy     : Ctag Range          : 11-12
              : VRF            : ten1vrf1
              : L3Vni               : 8192

+------+-------------------------+-------+---------+--------------+--------------+--------
+----------------------+---------+----------------+--------------+-------------
+------------------+
| Ctag |          Ctag           | L2Vni | BD Name | Anycast IPv4 | Anycast IPv6 |
Suppress|        Local IP      | IPv6 ND |    IPv6 ND     |   IPv6 ND    |  Dev State  |
App State  |
|      |       Description       |       |         |              |              | ARP/ND
|[Device-IP->Local-IP] |   Mtu   | Managed Config | Other Config |
|             |
+------+-------------------------+-------+---------+--------------+--------------
+-------------+----------------------+---------+----------------+--------------
+-------------+-------------+
|  11  | Tenant L3 Extended VLAN |  11   |         | 10.0.11.1/24 |  11::1/127   |
F/T        |                      |         |     false      |    false     | provisioned
| cfg-in-sync |
+------+-------------------------+-------+---------+--------------+--------------
+-------------+----------------------+---------+----------------+--------------
+-------------+-------------+
|  12  | Tenant L3 Extended VLAN |  12   |         | 10.0.12.1/24 |  12::1/127   |
T/F        |                      |         |     false      |    false     | provisioned
| cfg-in-sync |
+------+-------------------------+-------+---------+--------------+--------------
+-------------+----------------------+---------+----------------+--------------
+-------------+-------------+
================================================================================================
=============================================================================
================================================================================================
=============================================================================
Name          : ten2epg1
Tenant        : ten2
Type     : extension
State         :
Description   :
Ports         : 10.20.246.15[0/11]
              : 10.20.246.16[0/11]
POs           :
Port Property : SwitchPort Mode     : trunk
              : Native Vlan Tagging : false
NW Policy     : Ctag Range          : 21-22
              : VRF            : ten2vrf1
              : L3Vni               : 8191
+------+-----------------------+-------+--------------+--------------+--------------
+--------+----------------------+---------+----------------+--------------+-------------
+----------------+
| Ctag |          Ctag         | L2Vni | BD Name      | Anycast IPv4 | Anycast IPv6 |
Suppress|        Local IP      | IPv6 ND |    IPv6 ND     |   IPv6 ND    |  Dev State  |
App State|
|      |       Description     |       |              |              |              |
ARP/ND |[Device-IP->Local-IP] |   Mtu   | Managed Config | Other Config |
|           |
+------+-----------------------+-------+--------------+--------------+--------------
+-------------+----------------------+---------+----------------+--------------
+-------------+-----------+
|  21  | Tenant L3 Extended BD | 4097  | Auto-BD-4097 | 10.0.21.1/24 |  21::1/127   |
F/T        |                       |         |     false      |    false     |
provisioned |cfg-in-sync|
+------+-----------------------+-------+--------------+--------------+--------------
+-------------+----------------------+---------+----------------+--------------
+-------------+-----------+
```

```
|  22  | Tenant L3 Extended BD | 4098  | Auto-BD-4098 | 10.0.22.1/24 |  22::1/127   |
T/F         |                       |         |     false      |    false     |
provisioned |cfg-in-sync|
+------+----------------------+-------+--------------+--------------+--------------
+-------------+----------------------+---------+----------------+--------------
+-------------+-----------+
================================================================================================
================================================================
```

Configure Local IP for Endpoint Group

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

- Creating endpoint groups (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.

1. To configure local IP during EPG create operations, run the following commands:

```
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       |             |               |
```

2. To delete local IP during EPG update operations, run the following commands:

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

3. To add local IP during EPG update operations, run the following commands:

```
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            |             |             |
```

EPG: Network Property: IP MTU

You can configure the maximum transmission unit (MTU) when you create or update an endpoint group.

You use the `--ip-mtu` parameter (in the format `ctag:value`) to configure the MTU for the tenant network. This value is then configured on the interface VE on the SLX device.

1. To configure MTU when you create an endpoint group, run the **`efa tenant epg create`** command.

   ```
   $ efa tenant epg create --name ten1epg1 --tenant ten1 --port 10.20.246.17[0/1],
   10.20.246.18[0/1] --switchport-mode trunk --ctag-range 11-12 --anycast-
   ip11:10.0.11.1/24
   --anycast-ip12:10.0.12.1/24 --anycast-ipv6 11:11::1/127 --anycast-ipv6 12:12::1/127
   --vrf ten1vrf1 --ip-mtu 11:7900 --ip-mtu 12:8900
   ```

   This example creates an endpoint group with MTU values for Ctag 11 and Ctag 12.
2. To configure MTU for an existing endpoint group, run the **`efa tenant epg update`** command during `vrf-add` or `ctag-range-add` operations.

   ```
   $ efa tenant epg update --name ten1epg1 --tenant ten1 --operation ctag-range-add
   --ctag-range 12 --anycast-ip12:10.0.12.1/24 --anycast-ipv6 12:12::1/127 --ip-mtu
   12:6990
   ```

   This example configures the MTU during a `ctag-range-add` operation.
3. To view the configured MTU for an endpoint group, run the **`efa tenant epg show --detail`** command.

BGP as a Service

BGP as a service allows creation and deletion of BGP peer and peer-group neighbors on a given fabric device.

The VRF must be created on the fabric device using EPG (Endpoint Group) create or update prior to the BGP neighbors.

BGP Peer Group

BGP Peer Group Instance Create

```
# efa tenant service bgp peer-group create --name <peer-group-name> --tenant <tenant-
name> --description <description> --pg-name <switch-ip:pg-name> --pg-asn <switch-ip:pg-
name,remote-asn> --pg-bfd <switch-ip:pg-name,bfd-enable(true/false),interval,min-
rx,multiplier> --pg-next-hop-self <switch-ip:pg-name,next-hop-self(true/false)> --pg-
update-source-ip <switch-ip:pg-name,update-source-ip>
```

```
# efa tenant service bgp peer-group create -name ten1BgpPG1 --tenant tenant1 --pg-name
10.24.80.134:pg1 --pg-asn 10.24.80.134:pg1,6000 --pg-bfd 10.24.80.134:pg1,true,100,200,5
```

```
--pg-next-hop-self 10.24.80.134:pg1,true --pg-update-source-ip
10.24.80.134:pg1,10.20.30.40
```

BGP Peer Group Instance Update - Peer Group Add

```
# efa tenant service bgp peer-group update --name <peer-group-name> --tenant <tenant-
name> --operation <peer-group-add|peer-group-delete|peer-group-desc-update> --description
<description> --pg-name <switch-ip:pg-name> --pg-asn <switch-ip:pg-name,remote-asn> --pg-
bfd <switch-ip:pg-name,bfd-enable(true/false),interval,min-rx,multiplier> --pg-next-hop-
self <switch-ip:pg-name,next-hop-self(true/false)> --pg-update-source-ip <switch-ip:pg-
name,update-source-ip>
```

```
efa tenant service bgp peer-group update --name ten1BgpPG1 --tenant tenant1 --operation
peer-group-add --pg-name 10.24.80.134:pg2 -pg-asn 10.24.80.134:pg2,7000 --pg-bfd
10.24.80.134:pg2,true,200,300,6 --pg-next-hop-self 10.24.80.134:pg2,true --pg-update-
source-ip 10.24.80.134:pg2,10.20.30.41
```

BGP Peer Group Instance Show

```
efa tenant service bgp peer-group show
================================================================================================
========================================
Name : ten1BgpPG1
Tenant : tenant1
State : bs-state-created
+--------------+-------------+------------+-------------+--------------+--------
+----------------+-------------+-------------+
| Device IP | PeerGroup | REMOTE ASN | BFD Enabled | BFD Interval | BFD Rx | BFD
Multiplier | Dev-state | App-state |
+--------------+-------------+------------+-------------+--------------+--------
+----------------+-------------+-------------+
| 10.24.80.134 | pg1 | 6000 | true | 100 | 200 | 5 | provisioned | cfg-in-sync |
+--------------+-------------+------------+-------------+--------------+--------
+----------------+-------------+-------------+
| 10.24.80.134 | pg2 | 7000 | true | 200 | 300 | 6 | provisioned | cfg-in-sync |
+--------------+-------------+------------+-------------+--------------
+-------------------------+-------------+-------------+
================================================================================================
========================================
```

Switch Config

```
Rack1-Device1# show running-config router bgp
router bgp
local-as 100
neighbor pg1 peer-group
neighbor pg1 remote-as 6000
neighbor pg1 update-source 10.20.30.40
neighbor pg1 next-hop-self neighbor pg1 bfd
neighbor pg1 bfd interval 100 min-rx 200 multiplier 5
neighbor pg2 peer-group
neighbor pg2 remote-as 7000
neighbor pg2 update-source 10.20.30.41
neighbor pg2 next-hop-self neighbor pg2 bfd
neighbor pg2 bfd interval 200 min-rx 300 multiplier 6
address-family ipv4 unicast
!
address-family ipv6 unicast
!
address-family l2vpn evpn
!
```

BGP Peer Group Instance Update - Peer Group Delete

```
# efa tenant service bgp peer-group update --name ten1BgpPG1 --tenant tenant1 --operation
peer-group-delete --pg-name 10.24.80.134:pg2
```

```
# efa tenant service bgp peer-group show
================================================================================================
=======================================
Name : ten1BgpPG1
Tenant : tenant1
State : bs-state-created
+--------------+-------------+------------+-------------+--------------+--------
+----------------+-------------+-------------+
| Device IP | PeerGroup | REMOTE ASN | BFD Enabled | BFD Interval | BFD Rx | BFD
Multiplier | Dev-state | App-state |
+--------------+-------------+------------+-------------+--------------+--------
+----------------+-------------+-------------+
| 10.24.80.134 | pg1 | 6000 | true | 100 | 200 | 5 | provisioned | cfg-in-sync |
+--------------+-------------+------------+-------------+--------------
+------------------------+-------------+-------------+
================================================================================================
=======================================
```

BGP Peer Group Instance Delete

```
# efa tenant service bgp peer-group delete --name ten1BgpPG1 --tenant tenant1
```

BGP Static Peer

BGP Peer Instance Create - Static Peer

```
# efa tenant service bgp peer create --name <peer-name> --tenant <tenant-name> --ipv4-uc-
nbr <switch-ip,vrf-name:ipv4-neighbor,remote-as> --ipv4-uc-nbr-bfd <switch-ip,vrf-
name:ipv4-neighbor,bfd-enable(true/false),bfd-interval,bfd-rx,bfd-mult> --ipv4-uc-nbr-
update-source-ip <switch-ip,vrf-name:ipv4-neighbor,update-source-ip> --ipv4-uc-nbr-next-
hop-self <switch-ip,vrf-name:ipv4-neighbor,next-hop-self(true/false)> --ipv6-uc-nbr
<switch-ip,vrf-name:ipv6-neighbor,remote-as> --ipv6-uc-nbr-bfd <switch-ip,vrf-name:ipv6-
neighbor,bfd-enable(t/f),bfd-interval,bfd-rx,bfd-mult> --ipv6-uc-nbr-update-source-ip
<switch-ip,vrf-name:ipv6-neighbor,update-source-ip> --ipv6-uc-nbr-next-hop-self <switch-
ip,vrf-name:ipv6-neighbor,next-hop-self(true/false)>
```

```
# efa tenant service bgp peer create --name bgpservice1 --tenant tenant1 --ipv4-uc-nbr
10.24.80.134,red:10.20.30.40,5000 --ipv4-uc-nbr-bfd
10.24.80.134,red:10.20.30.40,true,100,200,5 --ipv4-uc-nbr-update-source-ip
10.24.80.134,red:10.20.30.40,11.22.20.33 --ipv4-uc-nbr-next-hop-self
10.24.80.134,red:10.20.30.40,true
```

BGP Peer Instance Show - Static Peer

```
# efa tenant service bgp show
```

```
# efa tenant service bgp peer 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 |
+--------------+-----+------+---------+-------------+------------+-------------
+--------------+--------+----------------+-------------+-------------+
```

```
| 10.24.80.134 | red | ipv4 | unicast | 10.20.30.40 | 5000 | true | 100 | 200 | 5 |
provisioned | cfg-in-sync |
+--------------+-----+------+---------+-------------+------------+-------------
+--------------+--------+----------------+-------------+-------------+
================================================================================================
================================================================
```

BGP Peer Instance Update - Static Peer Add

```
# efa tenant service bgp peer update --name <peer-name> --tenant <tenant-name> --
operation peer-add --ipv4-uc-nbr <switch-ip,vrf-name:ipv4-neighbor,remote-as> --ipv4-uc-
nbr-bfd <switch-ip,vrf-name:ipv4-neighbor,bfd-enable(t/f),bfd-interval,bfd-rx,bfd-mult> --
ipv4-uc-nbr-update-source-ip <switch-ip,vrf-name:ipv4-neighbor,update-source-ip> --ipv4-
uc-nbr-next-hop-self <switch-ip,vrf-name:ipv4-neighbor,next-hop-self(true/false)> --ipv6-
uc-nbr <switch-ip,vrf-name:ipv6-neighbor,remote-as> --ipv6-uc-nbr-bfd <switch-ip,vrf-
name:ipv6-neighbor,bfd-enable(t/f),bfd-interval,bfd-rx,bfd-mult> --ipv6-uc-nbr-update-
source-ip <switch-ip,vrf-name:ipv6-neighbor,update-source-ip> --ipv6-uc-nbr-next-hop-self
<switch-ip,vrf-name:ipv6-neighbor,next-hop-self(true/false)>
```

```
# efa tenant service bgp peer update --name bgpservice1 --tenant tenant1 --operation peer-
add --ipv6-uc-nbr 10.24.80.134,red:10::40,5000 --ipv6-uc-nbr-bfd
10.24.80.134,red:10::40,true,100,200,5 --ipv6-uc-nbr-update-source-ip
10.24.80.134,red:10::40,11::22 --ipv6-uc-nbr-next-hop-self 10.24.80.134,red:10::40,true
efa tenant service bgp peer 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 |
+--------------+-----+------+---------+-------------+------------+-------------
+--------------+--------+----------------+-------------+-------------+
| 10.24.80.134 | red | ipv6 | unicast | 10::40 | 5000 | false | 100 | 200 | 5 |
provisioned | cfg-in-sync |
+--------------+-----+------+---------+-------------+------------+-------------
+--------------+--------+----------------+-------------+-------------+
| 10.24.80.134 | red | ipv4 | unicast | 10.20.30.40 | 5000 | false | 100 | 200 | 5 |
provisioned | cfg-in-sync |
+--------------+-----+------+---------+-------------+------------+-------------
+--------------+--------+----------------+-------------+-------------+
================================================================================================
================================================================
```

Switch Config

```
Rack1-Device1# sh run router bgp
router bgp
local-as 100
address-family ipv4 unicast
!
address-family ipv4 unicast vrf red
neighbor 10.20.30.40 remote-as 5000
neighbor 10.20.30.40 next-hop-self
neighbor 10.20.30.40 bfd
neighbor 10.20.30.40 bfd interval 100 min-rx 200 multiplier 5
neighbor 10.20.30.40 update-source 11.22.20.33
!
address-family ipv6 unicast
!
address-family ipv6 unicast vrf red
neighbor 10::40 remote-as 5000
```

```
neighbor 10::40 next-hop-self
neighbor 10::40 bfd
neighbor 10::40 bfd interval 100 min-rx 200 multiplier 5
neighbor 10::40 update-source 11::22
!
address-family l2vpn evpn
!
```

BGP Peer Instance Update - Static Peer Delete

```
# efa tenant service bgp peer update --name <peer-name> --tenant <tenant-name> --
operation peer-delete --ipv4-unicast-neighbor <switch-ip,vrf-name:ipv4-neighbor> --ipv6-
unicast-neighbor <switch-ip,vrf-name:ipv4-neighbor >
```

```
# efa tenant service bgp peer update --name bgpservice1 --tenant tenant1 --operation peer-
delete --ipv4-unicast-neighbor 10.24.80.134,red:10.20.30.40
# 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 |
+--------------+-----+------+---------+-------------+------------+-------------
+--------------+--------+----------------+-------------+-------------+
| 10.24.80.134 | red | ipv6 | unicast | 10::40 | 5000 | false | 0 | 0 | 0 | provisioned |
cfg-in-sync |
+--------------+-----+------+---------+-------------+------------+-------------
+--------------+--------+----------------+-------------+-------------+
================================================================================================
================================================================
```

BGP Dynamic Peer

BGP Peer Instance Create - Dynamic Peer

```
# efa tenant service bgp peer create --name <peer-name> --tenant <tenant-name> --ipv4-uc-
dyn-nbr <switch-ip,vrf-name:listen-range,peer-group-name,listen-limit> --ipv6-uc-dyn-nbr
<switch-ip,vrf-name:listen-range,peer-group-name,listen-limit>
```

```
# efa tenant service bgp peer create --name bgpservice1 --tenant tenant1 --operation peer-
add –ipv4-uc-dyn-nbr 10.24.80.134,red:11::22/127,pg1,20
```

BGP Peer Instance Show - Dynamic Peer

```
# efa tenant service bgp peer show
================================================================================================
================================================================
Name : bgpservice1
Tenant : tenant1
State : bs-state-created
+--------------+-----+------+---------+----------------+------------+-------------
+-------------+-------------+
| Device IP | VRF | AFI | SAFI | LISTEN RANGE | Peer Group | LISTEN LIMIT | Dev-state |
App-state |
+--------------+-----+------+---------+----------------+------------+-------------
+-------------+-------------+
| 10.24.80.134 | red | ipv4 | unicast | 11.22.33.44/30 | pg1 | 10 | provisioned | cfg-in-
sync |
```

```
+--------------+-----+------+---------+----------------+------------+--------------
+-------------+-------------+ | 10.24.80.134 | red | ipv6 | unicast | 11::22/127 | pg1 |
20 | provisioned | cfg-in-sync |
+--------------+-----+------+---------+----------------+------------+--------------
+-------------+-------------+
================================================================================================
================================================================
```

Switch Config

```
Rack1-Device1# sh run router bgp
router bgp
local-as 100
neighbor pg1 peer-group
neighbor pg1 remote-as 6000
address-family ipv4 unicast
!
address-family ipv4 unicast vrf red
listen-range 11.22.33.44/30 peer-group pg1 limit 10
!
address-family ipv6 unicast
!
address-family ipv6 unicast vrf red
listen-range 11::22/127 peer-group pg1 limit 20
!
address-family l2vpn evpn
!
!
```

BGP Peer Instance Update - Dynamic Peer Delete

```
# efa tenant service bgp peer update --name <peer-name> --tenant <tenant-name> --
operation peer-delete --ipv4-uc-dyn-nbr <switch-ip,vrf-name:listen-range,peer-group-name>
--ipv6-uc-dyn-nbr <switch-ip,vrf-name:listen-range,peer-group-name>
```

```
# efa tenant service bgp peer create --name bgpservice1 --tenant tenant1 --operation peer-
delete -ipv4-uc-dyn-nbr 10.24.80.134,red:11::22/127,pg1

# efa tenant service bgp peer show
================================================================================================
================================================================
Name : bgpservice1
Tenant : tenant1
State : bs-state-created
+--------------+-----+------+---------+----------------+------------+--------------
+-------------+-------------+
| Device IP | VRF | AFI | SAFI | LISTEN RANGE | Peer Group | LISTEN LIMIT | Dev-state |
App-state |
+--------------+-----+------+---------+----------------+------------+--------------
+-------------+-------------+
| 10.24.80.134 | red | ipv4 | unicast | 11.22.33.44/30 | pg1 | 10 | provisioned | cfg-in-
sync |
+--------------+-----+------+---------+----------------+------------+--------------
+-------------+-------------+
================================================================================================
================================================================
```

BGP Peer Instance Delete

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

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

Configure Listen Limit on BGP Dynamic Peer

EFA enables configuration of global router BGP listen-limit which depends on the router BGP dynamic peer scale requirements. Listen-limit configuration defined under the “global router bgp” context signifies the maximum number of dynamic BGP peers that can be operational across the VRFs in the SLX. Default value of the global router BGP listen-limit is 100, which you can modify with any value in the range of 1-2400.

Note

- For the SLX version 20.3.4 or lower, the supported listen-limit range is <1-1024>.
- For the SLX version 20.4.1 or higher, the supported listen-limit range is <1-2400>.

Ensure that the listen-limit configuration defined at the “dynamic peer listen-range” level is less than or equal to the “listen-limit” configuration defined under the “global router bgp”. Pre-2.5.0 versions of EFA do not support configuration of global router bgp listen-limit.

The maximum number of dynamic BGP peers is limited to the SLX default (100), when provisioned through EFA.

Configure the new fabric setting for each fabric with `bgp-dynamic-peer-listen-limit` command. The fabric setting is applicable for Clos and small data center fabrics, and for all types of devices (leaf, border-leaf, spine, super-spine).

Configure the `bgp-dynamic-peer-listen-limit` value on all the devices of the fabric when you configure the fabric.

You can configure the `bgp-dynamic-peer-listen-limit` on an already provisioned fabric. For the new value to be effective, run the **`fabric configure`** command, followed by the **`efa fabric setting update`** command. This enables you to configure **`bgp-dynamic-peer-listen-limit`** on the existing pre-2.5.0 deployments.

Note

You can only increase (but not decrease) the value of `bgp-dynamic-peer-listen-limit` on an already provisioned fabric.

1. Run the following command:

```
efa fabric setting update –name <fabric-name> --bgp-dynamic-peer-listen-limit <1-2400>
```

2. Verify the switch configuration on the SLX device.

```
Rack1-Device1# show running-config
router bgp
router bgp
 local-as 4200000000
 capability as4-enable
 fast-external-fallover
 listen-limit 200
 neighbor 10.20.20.5 remote-as
4200000000
 neighbor 10.20.20.5 next-hop-self
 address-family ipv4 unicast
  network 172.31.254.139/32
  network 172.31.254.226/32
  maximum-paths 8
  graceful-restart
 !
 address-family ipv6 unicast
 !
 address-family l2vpn evpn
  graceful-restart
 !
!
```

```
Rack1-Device2# show running-config
router bgp
router bgp
 local-as 4200000000
 capability as4-enable
 fast-external-fallover
 listen-limit 200
 neighbor 10.20.20.4 remote-as
4200000000
 neighbor 10.20.20.4 next-hop-self
 address-family ipv4 unicast
  network 172.31.254.115/32
  network 172.31.254.226/32
  maximum-paths 8
  graceful-restart
 !
 address-family ipv6 unicast
 !
 address-family l2vpn evpn
  graceful-restart
 !
!
```

Getting the operational state of the BGP peers

Use this procedure to get the operational state of the BGP peers belonging to the tenant VRF (non default-vrf).

1. Run the following command to create a BGP peer on tenant VRF:

```
efa inventory device register --ip 10.20.246.23,10.20.246.24 --username admin --
password password
efa inventory device register --ip 10.20.246.21,10.20.246.22 --username admin --
password password
efa inventory device register --ip 10.20.246.14 --username admin --password password
efa fabric create --name fabric4 --type clos
efa fabric setting update --name fabric4 --vni-auto-map No --backup-routing-enable yes
efa fabric device add --ip 10.20.246.14 --role spine --name fabric4 --username admin --
password password
efa fabric device add --ip 10.20.246.23 --role leaf  --name fabric4 --username admin --
password password
efa fabric device add --ip 10.20.246.24 --role leaf  --name fabric4 --username admin --
password password
efa fabric device add --ip 10.20.246.22 --role border-leaf --name fabric4 --username
admin --password password
efa fabric device add --ip 10.20.246.21 --role border-leaf --name fabric4 --username
admin --password password
efa fabric configure --name fabric4


efa tenant create --name tenant2 --port 10.20.246.23[0/21-24],10.20.246.24[0/21-24] --
vlan-range 100-200 --l2-vni-range 12000-13000  --vrf-count 25 --l3-vni-range 8000-9000

efa tenant po create --name po101 --port 10.20.246.23[0/22],10.20.246.24[0/22] --speed
10Gbps --negotiation active --tenant tenant2

efa tenant vrf create --name vrf101 --tenant tenant2 --rt-type import --rt 101:102 --
rt-type export --rt 105:104
efa tenant vrf create --name vrf102 --tenant tenant2 --rt-type import --rt 104:105 --
rt-type export --rt 200:108 --local-asn 34566 --ipv4-static-route-next-hop
10.20.246.23,50.0.0.0/24,20.0.0.2 --ipv6-static-route-next-hop
```

```
10.20.246.23,3001:1::/64,01::2

efa tenant epg create --name epg1 --ctag-range  100-102 --po po101 --port
10.20.246.23[0/23] --switchport-mode trunk --tenant tenant2 --vrf vrf102 --anycast-ip
100:10.10.10.254/24 --anycast-ip 101:10.10.1.254/24 --anycast-ip 102:10.10.2.254/24 --
anycast-ipv6 100:3001:10:0:1::1/64  --anycast-ipv6 101:3001:10:0:2::1/64 --anycast-
ipv6 102:3002:10:0:3::1/64 --local-ip 100,10.20.246.23:121.10.1.2/24 --local-ip
101,10.20.246.23:121.10.2.2/24 --local-ipv6 102,10.20.246.23:121:a::2/64 --local-ip
100,10.20.246.24:121.10.1.3/24 --local-ip 101,10.20.246.24:121.10.2.3/24 --local-ipv6
102,10.20.246.24:121:a::3/64

efa tenant service bgp peer-group create --tenant tenant2 --name "pg1" --pg-name
10.20.246.23:peerb1 --pg-asn 10.20.246.23,peerb1:4294967295 --pg-bfd-enable
10.20.246.23,peerb1:true --pg-bfd 10.20.246.23,peerb1:30000,30000,50 --pg-next-hop-
self 10.20.246.23,peerb1:true --pg-update-source-ip 10.20.246.23,peerb1:3.3.3.3 --pg-
name 10.20.246.24:peerb1 --pg-asn 10.20.246.24,peerb1:4294967295 --pg-bfd-enable
10.20.246.24,peerb1:true --pg-bfd 10.20.246.24,peerb1:30000,30000,50 --pg-next-hop-
self 10.20.246.24,peerb1:true --pg-update-source-ip 10.20.246.24,peerb1:3.3.3.3

efa tenant service bgp peer create --name B3 --tenant tenant2 --ipv4-uc-nbr
10.20.246.23,vrf102:121.10.1.3,34566 --ipv4-uc-nbr
10.20.246.23,vrf102:121.10.2.3,34566 --ipv6-uc-nbr 10.20.246.23,vrf102:121:a::3,34566
--ipv4-uc-nbr 10.20.246.24,vrf102:121.10.1.2,34566 --ipv4-uc-nbr
10.20.246.24,vrf102:121.10.2.2,34566 --ipv6-uc-nbr 10.20.246.24,vrf102:121:a::2,34566


efa tenant epg create --name epg2 --ctag-range  105-107 --po po101 --port
10.20.246.23[0/24] --switchport-mode trunk --tenant tenant2 --vrf vrf101 --anycast-ip
105:11.11.10.254/24 --anycast-ip 106:11.11.1.254/24 --anycast-ip 107:11.11.2.254/24 --
anycast-ipv6 105:1001:11:0:1::1/64  --anycast-ipv6 106:1001:11:0:2::1/64 --anycast-
ipv6 107:1002:11:0:3::1/64 --local-ip 105,10.20.246.23:141.10.1.2/24 --local-ip
106,10.20.246.23:141.10.2.2/24 --local-ipv6 107,10.20.246.23:141:a::2/64 --local-ip
105,10.20.246.24:141.10.1.3/24 --local-ip 106,10.20.246.24:141.10.2.3/24 --local-ipv6
107,10.20.246.24:141:a::3/64

efa tenant service bgp peer create --name B2 --tenant tenant2 --ipv4-uc-nbr
10.20.246.23,vrf101:141.10.1.3,65000 --ipv4-uc-nbr
10.20.246.23,vrf101:141.10.2.3,65000 --ipv6-uc-nbr 10.20.246.23,vrf101:141:a::3,65000
--ipv4-uc-nbr 10.20.246.24,vrf101:141.10.1.2,65000 --ipv4-uc-nbr
10.20.246.24,vrf102:141.10.2.2,65000 --ipv6-uc-nbr 10.20.246.24,vrf102:141:a::2,65000

efa tenant create --name tenant3 --port 10.20.246.23[0/11-14],10.20.246.24[0/11-14] --
vlan-range 30-40 --l2-vni-range 2000-3000  --vrf-count 25 --l3-vni-range 5000-6000

efa tenant po create --name po3 --port 10.20.246.23[0/11],10.20.246.24[0/11] --speed
10Gbps --negotiation active --tenant tenant3

efa tenant vrf create --name vrf31 --tenant tenant3 --rt-type import --rt 301:302 --rt-
type export --rt 305:304
efa tenant vrf create --name vrf32 --tenant tenant3 --rt-type import --rt 304:305 --rt-
type export --rt 300:308 --local-asn 34566 --ipv4-static-route-next-hop
10.20.246.23,30.0.0.0/24,30.0.0.2 --ipv6-static-route-next-hop
10.20.246.23,5001:1::/64,01::2

efa tenant epg create --name epg3 --ctag-range  30-32 --po po3 --port
10.20.246.23[0/13] --switchport-mode trunk --tenant tenant3 --vrf vrf32 --anycast-ip
30:30.30.10.254/24 --anycast-ip 32:30.10.1.254/24 --anycast-ip 31:30.10.2.254/24 --
anycast-ipv6 30:5001:10:0:1::1/64  --anycast-ipv6 32:5001:10:0:2::1/64 --anycast-ipv6
31:5002:10:0:3::1/64 --local-ip 30,10.20.246.23:131.10.1.1/24 --local-ip
32,10.20.246.24:131.10.1.2/24 --local-ipv6 31,10.20.246.23:131:a::1/64
efa tenant epg create --name epg32 --ctag-range  35-37 --po po3 --port
10.20.246.23[0/14] --switchport-mode trunk --tenant tenant3 --vrf vrf31 --anycast-ip
35:11.11.10.254/24 --anycast-ip 36:11.11.1.254/24 --anycast-ip 37:11.11.2.254/24 --
anycast-ipv6 35:301:11:0:1::1/64  --anycast-ipv6 36:301:11:0:2::1/64 --anycast-ipv6
```

```
37:302:11:0:3::1/64 --local-ip 35,10.20.246.23:131.10.1.1/24 --local-ip
36,10.20.246.24:131.10.1.2/24 --local-ipv6 37,10.20.246.23:131:a::1/64

efa tenant service bgp peer-group create --tenant tenant3 --name "pg3" --pg-name
10.20.246.23:peerb3 --pg-asn 10.20.246.23,peerb3:4294967295 --pg-bfd-enable
10.20.246.23,peerb3:true --pg-bfd 10.20.246.23,peerb3:30000,30000,50 --pg-next-hop-
self 10.20.246.23,peerb3:true --pg-update-source-ip 10.20.246.23,peerb3:31.3.3.3 --pg-
name 10.20.246.24:peerb3 --pg-asn 10.20.246.24,peerb3:4294967295 --pg-bfd-enable
10.20.246.24,peerb3:true --pg-bfd 10.20.246.24,peerb3:30000,30000,50 --pg-next-hop-
self 10.20.246.24,peerb3:true --pg-update-source-ip 10.20.246.24,peerb3:3.3.3.3
efa tenant service bgp peer create --name B2 --tenant tenant3 --ipv4-uc-nbr
10.20.246.23,vrf31:131.10.1.2,4200000000 --ipv4-uc-nbr
10.20.246.24,vrf31:131.10.1.1,4200000000  --ipv6-uc-nbr
10.20.246.24,vrf31:131:a::1,4200000000
```

2. Run the following command to get the operational state of the BGP peers belonging to both default VRF and Tenant VRF:

```
efa tenant service bgp peer operational show --tenant <tenant-name> --vrf <vrf-name>
```

3. Run the following command to get the operational state of the BGP peers for a given tenant VRF:

```
efa tenant service bgp peer operational show --tenant tenant11 --vrf v1
```

4. Run the following command to get the operational state of the BGP peers for a given tenant:

```
efa tenant service bgp peer operational show --tenant tenant11
```

5. Run the following command to get the operational state of the BGP peers for all tenant:

```
efa tenant service bgp peer operational show
```

Configure route map attribute

To enable external connectivity, configure the route map attribute when you create or update BGP peer.

Note
For information about commands and supported parameters to configure route map attribute, see *Extreme Fabric Automation Command Reference, 2.7.0*.

1. Run the following command to configure route map when you create BGP peer:

```
efa tenant service bgp peer create --name <bgp-peer-name> --tenant <tenant-name>
           --ipv4-uc-nbr <device-ip,vrf-name:neighbor-ip,remote-asn>
           --ipv4-uc-nbr-bfd <device-ip,vrf-name:neighbor-ip,true|false>
           --ipv4-uc-nbr-route-map-in <device-ip,vrf-name:neighbor-ip,route-map-name>
           --ipv4-uc-nbr-route-map-out <device-ip,vrf-name:neighbor-ip,route-map-name>
```

2. Run the following command to configure route map when you update BGP peer:

```
efa tenant service bgp peer update --name <bgp-peer-name> --tenant <tenant-name>
           --operation peer-add
           --ipv4-uc-nbr <device-ip,vrf-name:neighbor-ip,remote-asn>
           --ipv4-uc-nbr-bfd <device-ip,vrf-name:neighbor-ip,true|false>
```

```
            --ipv4-uc-nbr-route-map-in <device-ip,vrf-name:neighbor-ip,route-map-name>
            --ipv4-uc-nbr-route-map-out <device-ip,vrf-name:neighbor-ip,route-map-name>
```

Example

```
efa tenant service bgp peer update --name ten1bgppeer1 --tenant ten1
             --operation peer-add
             --ipv4-uc-nbr 10.20.246.15,ten1vrf1:10.20.30.50,50000
             --ipv4-uc-nbr-bfd 10.20.246.15,ten1vrf1:10.20.30.50,true
             --ipv4-uc-nbr-route-map
10.20.246.15,ten1vrf1:10.20.30.50,rmap2,in
             --ipv4-uc-nbr 10.20.246.16,ten1vrf1:10.20.30.50,50000
             --ipv4-uc-nbr-bfd 10.20.246.16,ten1vrf1:10.20.30.50,true
             --ipv4-uc-nbr-route-map
10.20.246.16,ten1vrf1:10.20.30.50,rmap2,out
```

```
efa tenant service bgp peer show --detail
================================================================================
============

Name              : bgp173-2501
Tenant            : tenant11
State             : bgp-peer-created
Description       :

Static Peer
-----------
        Device IP                   : 10.20.246.23
        VRF                         : v1
        AFI                         : ipv4
        SAFI                        : unicast
        Remote IP                   : 10.40.40.252
        Remote ASN                  : 6500
        Next Hop Self               : false
        Update Source IP            :
        BFD Enabled                 : true
        BFD Interval                : 0
        BFD Rx                      : 0
        BFD Multiplier              : 0
        MD5 Password                : $9$MCgKGaNt6OASX68/7TC6Lw==
        Remove Private AS           : false
        Default Originate           : false
        Default Originate Route Map :
        Prefix List In              :
        Prefix List Out             :
        Route Map In                : tt
        Route Map Out               : rmap1
        Dev State                   : provisioned
        App State                   : cfg-in-sync

        Device IP                   : 10.20.246.23
        VRF                         : v1
        AFI                         : ipv4
        SAFI                        : unicast
        Remote IP                   : fd40:4040:4040:1::fe
        Remote ASN                  : 6500
        Next Hop Self               : false
        Update Source IP            :
        BFD Enabled                 : true
        BFD Interval                : 0
        BFD Rx                      : 0
        BFD Multiplier              : 0
        MD5 Password                : $9$MCgKGaNt6OASX68/7TC6Lw==
```

```
        Remove Private AS           : false
        Default Originate           : false
        Default Originate Route Map :
        Prefix List In              :
        Prefix List Out             :
        Route Map In                : rmap1
        Route Map Out               :
        Dev State                   : provisioned
        App State                   : cfg-in-sync


Dynamic Peer
------------
        0 Records
=============================================================================
```

3. Verify the switch configuration on the SLX device.

```
Rack1-Device1# show running-config
router bgp
router bgp
 local-as 4200000000
 capability as4-enable
 fast-external-fallover
 neighbor 10.20.20.4 remote-as
4200000000
 neighbor 10.20.20.4 next-hop-self
 address-family ipv4 unicast
  network 172.31.254.46/32
  network 172.31.254.123/32
  maximum-paths 8
  graceful-restart
 !
 address-family ipv4 unicast vrf
ten1vrf1
  redistribute connected
  neighbor 10.20.30.40 remote-as
50000
  neighbor 10.20.30.40 route-map
in rmap1
  neighbor 10.20.30.40 bfd
  neighbor 10.20.30.50 remote-as
50000
  neighbor 10.20.30.50 route-map
in rmap2
  neighbor 10.20.30.50 bfd
  maximum-paths 8
 !
 address-family ipv4 unicast
 !
 address-family ipv4 unicast vrf
ten1vrf1
  redistribute connected
  maximum-paths 8
 !
 address-family l2vpn evpn
  graceful-restart
 !
!
```

```
Rack1-Device2# show running-config
router bgp
router bgp
 local-as 4200000000
 capability as4-enable
 fast-external-fallover
 neighbor 10.20.20.5 remote-as
4200000000
 neighbor 10.20.20.5 next-hop-self
 address-family ipv4 unicast
  network 172.31.254.46/32
  network 172.31.254.176/32
  maximum-paths 8
  graceful-restart
 !
 address-family ipv4 unicast vrf
ten1vrf1
  redistribute connected
  neighbor 10.20.30.40 remote-as
50000
  neighbor 10.20.30.40 route-map
out rmap1
  neighbor 10.20.30.40 bfd
  neighbor 10.20.30.50 remote-as
50000
  neighbor 10.20.30.50 route-map
out rmap2
  neighbor 10.20.30.50 bfd
  maximum-paths 8
 !
 address-family ipv4 unicast
 !
 address-family ipv4 unicast vrf
ten1vrf1
  redistribute connected
  maximum-paths 8
 !
 address-family l2vpn evpn
  graceful-restart
 !
!
```

Configure remove-private-as on BGP Peer

To enable external connectivity, you can configure the remove-private-as attribute when you create or update BGP peer. By default, remove-private-as is disabled and it is an IPv4 attribute only.

1. Run the following command to create a BGP Peer on a tenant VRF:

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

                        --ipv4-uc-nbr <device-ip,vrf-name:neighbor-ip,remote-asn>
                        --ipv4-uc-nbr-bfd <device-ip,vrf-name:neighbor-ip,true|false>

                        --ipv4-uc-nbr-remove-private-as <device-ip,vrf-name:neighbor-
ip,true|false>
```

2. Run the following command to update a BGP Peer on a tenant VRF:

```
efa tenant service bgp peer update --name <bgp-peer-name> --tenant <tenant-name>

             --operation peer-add
             --ipv4-uc-nbr 10.20.246.25,v1:10.20.30.50,50000
        --ipv4-uc-nbr-bfd 10.20.246.25,v1:10.20.30.50,true
        --ipv4-uc-nbr-remove-private-as 10.20.246.25,v1:10.20.30.50,true
        --ipv4-uc-nbr 10.20.246.26,v1:10.20.30.50,50000
        --ipv4-uc-nbr-bfd 10.20.246.26,v1:10.20.30.50,true
        --ipv4-uc-nbr-remove-private-as 10.20.246.26,v1:10.20.30.50,false
```

Example:

```
efa tenant service bgp peer create --name ten1bgppeer1 --tenant tenant11
        --ipv4-uc-nbr 10.20.246.25,v1:10.20.30.40,50000
        --ipv4-uc-nbr-bfd 10.20.246.25,v1:10.20.30.40,true
        --ipv4-uc-nbr-remove-private-as 10.20.246.25,v1:10.20.30.40,true
        --ipv4-uc-nbr 10.20.246.26,v1:10.20.30.40,50000
        --ipv4-uc-nbr-bfd 10.20.246.26,v1:10.20.30.40,true
        --ipv4-uc-nbr-remove-private-as 10.20.246.26,v1:10.20.30.40,true
```

```
10.20.246.25
ORCA_01# show running-config
router bgp
address-family ipv4 unicast vrf v1
  redistribute connected
  neighbor 10.20.30.40 remote-as
50000
  neighbor 10.20.30.40 remove-
private-as
  neighbor 10.20.30.40 bfd
  neighbor 10.40.40.253 remote-as
4200000000
  neighbor 10.40.40.253 next-hop-
self
  maximum-paths 8
 !
```

```
10.20.246.26
ORCA_02# show running-config
router bgp
address-family ipv4 unicast vrf v1
  redistribute connected
  neighbor 10.20.30.40 remote-as
50000
  neighbor 10.20.30.40 remove-
private-as
  neighbor 10.20.30.40 bfd
  neighbor 10.40.40.252 remote-as
4200000000
  neighbor 10.40.40.252 next-hop-
self
  maximum-paths 8
 !
```

```
efa tenant service bgp peer show --detail
```

```
Name              :
ten1bgppeer1
Tenant            :
tenant11
State             : bgp-
peer-created
Description       :

Static Peer
-----------
Device
IP                    :
10.20.246.25
VRF
     : v1
AFI
     : ipv4
SAFI
     : unicast
Remote
IP                    :
10.20.30.40
Remote
ASN                   :
50000
Next Hop
Self                :
false
Update Source
IP             :
BFD
Enabled
 : true
BFD
Interval
 : 0
BFD
Rx
 : 0
BFD
```

```
Device
IP                    :
10.20.246.25
VRF
     : v1
AFI
     : ipv4
SAFI
     : unicast
Remote
IP                    :
10.20.30.50
Remote
ASN                   :
50000
Next Hop
Self                :
false
Update Source
IP             :
BFD
Enabled
 : true
BFD
Interval
 : 0
BFD
Rx
 : 0
BFD
Multiplier
 : 0
MD5
Password
 :
Remove Private
AS            : true
Default
Originate           :
false
```

```


BFD
Multiplier
 : 0
MD5
Password
 :
Remove Private
AS            : false
Default
Originate           :
false
Default Originate
Route Map :
Prefix List
In              :
Prefix List
Out             :
Route Map
In                :
Route Map
Out               :
Dev
State
 : provisioned
App
State
 : cfg-in-sync

Device
IP                    :
10.20.246.26
VRF
     : v1
AFI
     : ipv4
SAFI
     : unicast
Remote
```

```
Multiplier
 : 0
MD5
Password
 :
Remove Private
AS            : true
Default
Originate           :
false
Default Originate
Route Map :
Prefix List
In               :
Prefix List
Out              :
Route Map
In                 :
Route Map
Out                :
Dev
State
 : provisioned
App
State
 : cfg-in-sync
```

```
Default Originate
Route Map :
Prefix List
In               :
Prefix List
Out              :
Route Map
In                 :
Route Map
Out                :
Dev
State
 : provisioned
App
State
 : cfg-in-sync

Device
IP                     :
10.20.246.26
VRF
     : v1
AFI
     : ipv4
SAFI
     : unicast
Remote
IP                     :
10.20.30.50
Remote
ASN                    :
50000
Next Hop
Self                 :
false
Update Source
IP             :
BFD
Enabled
 : true
BFD
Interval
 : 0
BFD
Rx
 : 0
```

```
IP                     :
10.20.30.40
Remote
ASN                    :
50000
Next Hop
Self                 :
false
Update Source
IP             :
BFD
Enabled
 : true
BFD
Interval
 : 0
BFD
Rx
 : 0
BFD
Multiplier
 : 0
MD5
Password
 :
Remove Private
AS            : true
Default
Originate           :
false
Default Originate
Route Map :
Prefix List
In               :
Prefix List
Out              :
Route Map
In                 :
Route Map
Out                :
Dev
State
 : provisioned
App
State
 : cfg-in-sync
```

3. Verify the switch configuration on the SLX device.

```
Rack1-Device1# show running-config
router bgp
router bgp
 local-as 4200000000
 capability as4-enable
 fast-external-fallover
 neighbor 10.20.20.4 remote-as
4200000000
 neighbor 10.20.20.4 next-hop-self
 address-family ipv4 unicast
  network 172.31.254.46/32
  network 172.31.254.123/32
  maximum-paths 8
  graceful-restart
 !
 address-family ipv4 unicast vrf
ten1vrf1
  redistribute connected
  neighbor 10.20.30.40 remote-as
50000
  neighbor 10.20.30.40 remove-
private-as
  neighbor 10.20.30.40 bfd
  neighbor 10.20.30.50 remote-as
50000
  neighbor 10.20.30.50 remove-
private-as
  neighbor 10.20.30.50 bfd
  maximum-paths 8
 !
 address-family ipv6 unicast
 !
 address-family ipv6 unicast vrf
ten1vrf1
  redistribute connected
  maximum-paths 8
 !
 address-family l2vpn evpn
  graceful-restart
 !
!
```

```
Rack1-Device2# show running-config
router bgp
router bgp
 local-as 4200000000
 capability as4-enable
 fast-external-fallover
 neighbor 10.20.20.5 remote-as
4200000000
 neighbor 10.20.20.5 next-hop-self
 address-family ipv4 unicast
  network 172.31.254.46/32
  network 172.31.254.176/32
  maximum-paths 8
  graceful-restart
 !
 address-family ipv4 unicast vrf
ten1vrf1
  redistribute connected
  neighbor 10.20.30.40 remote-as
50000
  neighbor 10.20.30.40 remove-
private-as
  neighbor 10.20.30.40 bfd
  neighbor 10.20.30.50 remote-as
50000
  neighbor 10.20.30.50 remove-
private-as
  neighbor 10.20.30.50 bfd
  maximum-paths 8
 !
 address-family ipv6 unicast
 !
 address-family ipv6 unicast vrf
ten1vrf1
  redistribute connected
  maximum-paths 8
 !
 address-family l2vpn evpn
  graceful-restart
 !
!
```

Note

For information about commands and supported parameters to configure remove-private-as attribute, see *Extreme Fabric Automation Command Reference, 2.7.0*.

Configure remove-private-as on BGP Peer Group

To enable external connectivity, you can configure the remove-private-as attribute when you create or update BGP peer group. By default, remove-private-as is disabled.

1. Run the following command to create a BGP Peer-Group on a tenant VRF:

```
efa tenant service bgp peer-group create --name <bgp-pg-name> --tenant <tenant-name>
                   --pg-name <device-ip:pg-name> --pg-asn <device-ip,pg-name:remote-
asn>
                   --pg-bfd-enable <device-ip,pg-name:true|false>
                   --pg-remove-private-as <device-ip,pg-name:true|false>
```

2. Run the following command to update a BGP Peer-Group on a tenant VRF:

```
efa tenant service bgp peer-group update --name <bgp-pg-name> --tenant <tenant-name>
                   --operation peer-group-add
                   --pg-name <device-ip:pg-name> --pg-asn <device-ip,pg-name:remote-
asn>
                   --pg-bfd-enable <device-ip,pg-name:true|false>
                   --pg-remove-private-as <device-ip,pg-name:true|false>
```

Example:

```
efa tenant service bgp peer-group create --name ten1bgppg1 --tenant ten1
                   --pg-name 10.20.246.15:pg1 --pg-asn 10.20.246.15,pg1:55001
                   --pg-bfd-enable 10.20.246.15,pg1:true
                   --pg-remove-private-as 10.20.246.15,pg1:true
                   --pg-name 10.20.246.16:pg1 --pg-asn 10.20.246.16,pg1:55001
                   --pg-bfd-enable 10.20.246.16,pg1:true
                   --pg-remove-private-as 10.20.246.16,pg1:true

efa tenant service bgp peer-group update --name ten1bgppg1 --tenant ten1
                   --operation peer-group-add
                   --pg-name 10.20.246.15:pg2 --pg-asn 10.20.246.15,pg2:55002
                   --pg-bfd-enable 10.20.246.15,pg2:true
                   --pg-remove-private-as 10.20.246.15,pg2:true
                   --pg-name 10.20.246.16:pg2 --pg-asn 10.20.246.16,pg2:55002
                   --pg-bfd-enable 10.20.246.16,pg2:true
                   --pg-remove-private-as 10.20.246.16,pg2:true

efa tenant service bgp peer-group show --detail
==========================================================================
Name             : ten1bgppg1
Tenant           : ten1
State            : bgp-pg-state-created

Peer Group
----------
        Device IP        : 10.20.246.15
        Peer Group       : pg1
        Remote ASN       : 55001
        Next Hop Self    : false
        BFD Enabled      : true
        BFD Interval     :
        BFD Rx           :
        BFD Multiplier   :
        Remove Private AS: true
        Dev State        : provisioned
        App State        : cfg-in-sync

        Device IP        : 10.20.246.15
        Peer Group       : pg2
        Remote ASN       : 55002
        Next Hop Self    : false
        BFD Enabled      : true
        BFD Interval     :
        BFD Rx           :
        BFD Multiplier   :
        Remove Private AS: true
        Dev State        : provisioned
        App State        : cfg-in-sync

        Device IP        : 10.20.246.16
        Peer Group       : pg1
        Remote ASN       : 55001
        Next Hop Self    : false
        BFD Enabled      : true
```

```
        BFD Interval     :
        BFD Rx           :
        BFD Multiplier   :
        Remove Private AS: true
        Dev State        : provisioned
        App State        : cfg-in-sync

        Device IP        : 10.20.246.16
        Peer Group       : pg2
        Remote ASN       : 55002
        Next Hop Self    : false
        BFD Enabled      : true
        BFD Interval     :
        BFD Rx           :
        BFD Multiplier   :
        Remove Private AS: false
        Dev State        : provisioned
        App State        : cfg-in-sync
================================================================================
```

3. Verify the switch configuration on the SLX device.

```
Rack1-Device1# show running-config
router bgp
router bgp
 local-as 4200000000
 capability as4-enable
 fast-external-fallover
 neighbor pg1 peer-group
 neighbor pg1 remote-as 55001
 neighbor pg1 remove-private-as
 neighbor pg1 bfd
 neighbor pg2 peer-group
 neighbor pg2 remote-as 55002
 neighbor pg2 remove-private-as
 neighbor pg2 bfd
 neighbor 10.20.20.4 remote-as
4200000000
 neighbor 10.20.20.4 next-hop-self
 address-family ipv4 unicast
  network 172.31.254.46/32
  network 172.31.254.123/32
  maximum-paths 8
  graceful-restart
 !
 address-family ipv4 unicast vrf
ten1vrf1
  redistribute connected
  maximum-paths 8
 !
 address-family ipv6 unicast
 !
 address-family ipv6 unicast vrf
ten1vrf1
  redistribute connected
  maximum-paths 8
 !
 address-family l2vpn evpn
  graceful-restart
 !
!
```

```
Rack1-Device2# show running-config
router bgp
router bgp
 local-as 4200000000
 capability as4-enable
 fast-external-fallover
 neighbor pg1 peer-group
 neighbor pg1 remote-as 55001
 neighbor pg1 remove-private-as
 neighbor pg1 bfd
 neighbor pg2 peer-group
 neighbor pg2 remote-as 55002
 neighbor pg2 password remove-
private-as
 neighbor pg2 bfd
 neighbor 10.20.20.5 remote-as
4200000000
 neighbor 10.20.20.5 next-hop-self
 address-family ipv4 unicast
  network 172.31.254.46/32
  network 172.31.254.176/32
  maximum-paths 8
  graceful-restart
 !
 address-family ipv4 unicast vrf
ten1vrf1
  redistribute connected
  maximum-paths 8
 !
 address-family ipv6 unicast
 !
 address-family ipv6 unicast vrf
ten1vrf1
  redistribute connected
  maximum-paths 8
 !
 address-family l2vpn evpn
  graceful-restart
 !
!
```

Note

For information about commands and supported parameters to configure remove-private-as attribute, see *Extreme Fabric Automation Command Reference, 2.7.0*.

Configure default-originate to advertise default route on BGP Peer

To enable external connectivity, you can configure the default-originate attribute when you create or update BGP peer (IPv4 and IPv6). By default, default-originate is disabled.

1. Run the following command to create a BGP Peer on a tenant VRF:

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

                   --ipv4-uc-nbr <device-ip,vrf-name:neighbor-ip,remote-asn>
                   --ipv4-uc-nbr-bfd <device-ip,vrf-name:neighbor-ip,true|false>

                   --ipv4-uc-nbr-default-originate <device-ip,vrf-name:neighbor-
ip,true/false>
```

```
                    --ipv4-uc-nbr-default-originate-route-map <device-ip,vrf-
name:neighbor-ip,route-map>
```

2. Run the following command to update a BGP Peer on a tenant VRF:

```
efa tenant service bgp peer update --name <bgp-peer-name> --tenant <tenant-name>

                    --operation peer-add

                    --ipv4-uc-nbr <device-ip,vrf-name:neighbor-ip,remote-asn>
                    --ipv4-uc-nbr-bfd <device-ip,vrf-name:neighbor-ip,true|false>

                    --ipv4-uc-nbr-default-originate <device-ip,vrf-name:neighbor-
ip,true/false>
                    --ipv4-uc-nbr-default-originate-route-map <device-ip,vrf-
name:neighbor-ip,route-map>
```

Note

The `ipv4-uc-nbr-default-originate-route-map` attribute is an optional attribute.

Example:

```
efa tenant service bgp peer create --name ten1bgppeer1 --tenant tenant11
        --ipv4-uc-nbr 10.20.246.3,v1:10.20.30.40,50000
        --ipv4-uc-nbr-bfd 10.20.246.3,v1:10.20.30.40,true
        --ipv4-uc-nbr-default-originate 10.20.246.3,v1:10.20.30.40,true
        --ipv4-uc-nbr 10.20.246.4,v1:10.20.30.40,50000
        --ipv4-uc-nbr-bfd 10.20.246.4,v1:10.20.30.40,true
        --ipv4-uc-nbr-default-originate 10.20.246.4,v1:10.20.30.40,true
```

```
--ipv4-uc-nbr-default-originate-route-map 10.20.246.4,v1:10.20.30.40,rmap1
```

§efa tenant service bgp peer-group show --detail

```
Name      : ten1bgppeer1
Tenant          :
tenant11
State             : bgp-
peer-created
Description       :

Static Peer
-----------
Device
IP                    :
10.20.246.4
VRF
     : v1
AFI
     : ipv4
SAFI
     : unicast
Remote
IP                    :
10.20.30.40
Remote
ASN                   :
50000
Next Hop
Self                  :
false
Update Source
IP             :
BFD
Enabled
 : true
BFD
Interval
 : 0
BFD
Rx
 : 0
BFD
Multiplier
 : 0
MD5
Password
 :
Remove Private
AS            : false
Default
Originate             :
true
Default Originate
Route Map : rmap1
Prefix List
In                :
Prefix List
Out               :
Route Map
In                  :
Route Map
```

```
Device IP          :
10.20.246.4
VRF
     : v1
AFI
     : ipv4
SAFI
     : unicast
Remote
IP                    :
10.20.30.50
Remote
ASN                   :
50000
Next Hop
Self                  :
false
Update Source
IP             :
BFD
Enabled
 : true
BFD
Interval
 : 0
BFD
Rx
 : 0
BFD
Multiplier
 : 0
MD5
Password
 :
Remove Private
AS            : false
Default
Originate             :
false
Default Originate
Route Map :
Prefix List
In                :
Prefix List
Out               :
Route Map
In                  :
Route Map
Out                 :
Dev
State
 : provisioned
App
State
 : cfg-in-sync

Device
IP                    :
10.20.246.3
VRF
```

```
BFD Multiplier     : 0
MD5
Password
 :
Remove Private
AS            : false
Default
Originate             :
true
Default Originate
Route Map :
Prefix List
In                :
Prefix List
Out               :
Route Map
In                  :
Route Map
Out                 :
Dev
State
 : provisioned
App
State
 : cfg-in-sync

Device
IP                    :
10.20.246.3
VRF
     : v1
AFI
     : ipv4
SAFI
     : unicast
Remote
IP                    :
10.20.30.50
Remote
ASN                   :
50000
Next Hop
Self                  :
false
Update Source
IP             :
BFD
Enabled
 : true
BFD
Interval
 : 0
BFD
Rx
 : 0
BFD
Multiplier
 : 0
MD5
Password
```

Out :	: v1	:
Dev	AFI	Remove Private
State	: ipv4	AS : false
: provisioned	SAFI	Default
App	: unicast	Originate :
State	Remote	true
: cfg-in-sync	IP :	Default Originate
	10.20.30.40	Route Map : tt
	Remote	Prefix List
	ASN :	In :
	50000	Prefix List
	Next Hop	Out :
	Self :	Route Map
	false	In :
	Update Source	Route Map
	IP :	Out :
	BFD	Dev
	Enabled	State
	: true	: provisioned
	BFD	App
	Interval	State
	: 0	: cfg-in-sync
	BFD	
	Rx	
	: 0	

```
efa tenant service bgp peer show --detail
================================================================================
Name             : bgp173-2501
Tenant           : tenant11
State            : bgp-peer-created
Description      :
Static Peer
-----------
       Device IP                  : 10.20.246.3
    VRF                         : v1
    AFI                         : ipv4
    SAFI                        : unicast
    Remote IP                   : 10.20.30.40
    Remote ASN                  : 50000
    Next Hop Self               : false
    Update Source IP            :
    BFD Enabled                 : true
    BFD Interval                : 0
    BFD Rx                      : 0
    BFD Multiplier              : 0
    MD5 Password                :
    Remove Private AS           : false
    Default Originate           : true
    Default Originate Route Map :
    Prefix List In              :
    Prefix List Out             :
    Route Map In                :
    Route Map Out               :
    Dev State                   : provisioned
    App State                   : cfg-in-sync

      Device IP                  : 10.20.246.4
    VRF                         : v1
    AFI                         : ipv4
    SAFI                        : unicast
```

```
    Remote IP                   : 10.20.30.40
    Remote ASN                  : 50000
    Next Hop Self               : false
    Update Source IP            :
    BFD Enabled                 : true
    BFD Interval                : 0
    BFD Rx                      : 0
    BFD Multiplier              : 0
    MD5 Password                :
    Remove Private AS           : false
    Default Originate           : true
    Default Originate Route Map : rmap1
    Prefix List In              :
    Prefix List Out             :
    Route Map In                :
    Route Map Out               :
    Dev State                   : provisioned
    App State                   : cfg-in-sync
==============================================================================
```

3. Verify the switch configuration on the SLX device.

10.20.246.3	10.20.246.4

SLX# show running-config router bgp address-family ipv4 unicast vrf v1
  redistribute connected
  neighbor 10.20.30.40 remote-as 50000
  neighbor 10.20.30.40 default-originate
  neighbor 10.20.30.40 bfd
  neighbor 10.40.40.252 remote-as 4200000000
  neighbor 10.40.40.252 next-hop-self
  maximum-paths 8
 !
	SLX# show running-config router bgp address-family ipv4 unicast vrf v1
	  redistribute connected
	  neighbor 10.20.30.40 remote-as 50000
	  neighbor 10.20.30.40 default-originate route-map rmap1
	  neighbor 10.20.30.40 bfd
	  neighbor 10.40.40.253 remote-as 4200000000
	  neighbor 10.40.40.253 next-hop-self
	  maximum-paths 8
	 !

Note

For information about commands and supported parameters to configure default-originate attribute, see *Extreme Fabric Automation Command Reference, 2.7.0*.

Configure Backup Routing Neighbors on BGP Peer

You can configure backup routing neighbors with additional attributes, such as bidirectional forwarding detection, route-maps and prefix-lists.

By default, backup routing neighbor is configured with remote-as and next-hop-self. You can provide md5-password as a fabric setting which is applied on the neighbors.

Note

You can modify the MD5 password for the backup routing neighbors only by configuring and re-configuring the fabric.

1. To create backup routing neighbors when you create a BGP peer, run the following command:

```
efa tenant service bgp peer create --name <bgp-peer-name> --tenant <tenant-name>  --
ipv4-uc-nbr <device-ip,vrf-name:neighbor-ip,remote-asn>   --ipv4-uc-nbr-bfd <device-
ip,vrf-name:neighbor-ip,true|false>  --ipv4-uc-nbr-default-originate <device-ip,vrf-
name:neighbor-ip,true/false>  --ipv4-uc-nbr-default-originate-route-map <device-ip,vrf-
name:neighbor-ip,route-map>
```

2. To create backup routing neighbors when you update a BGP peer, run the following command:

```
efa tenant service bgp peer update --name <bgp-peer-name> --tenant <tenant-name>   --
operation peer-add --ipv4-uc-nbr <device-ip,vrf-name:neighbor-ip,remote-asn>  --ipv4-
uc-nbr-bfd <device-ip,vrf-name:neighbor-ip,true|false>  --ipv4-uc-nbr-default-
originate <device-ip,vrf-name:neighbor-ip,true/false>
```

3. To remove the backup routing neighbors association with BGP service when you delete a BGP peer, run the following command:

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

IPv6 Support

EFA supports the provisioning of IPv6 anycast gateways.

EFA Provisioning

The following commands support anycast gateways.

```
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
```

Software BFD Session Support on CEP

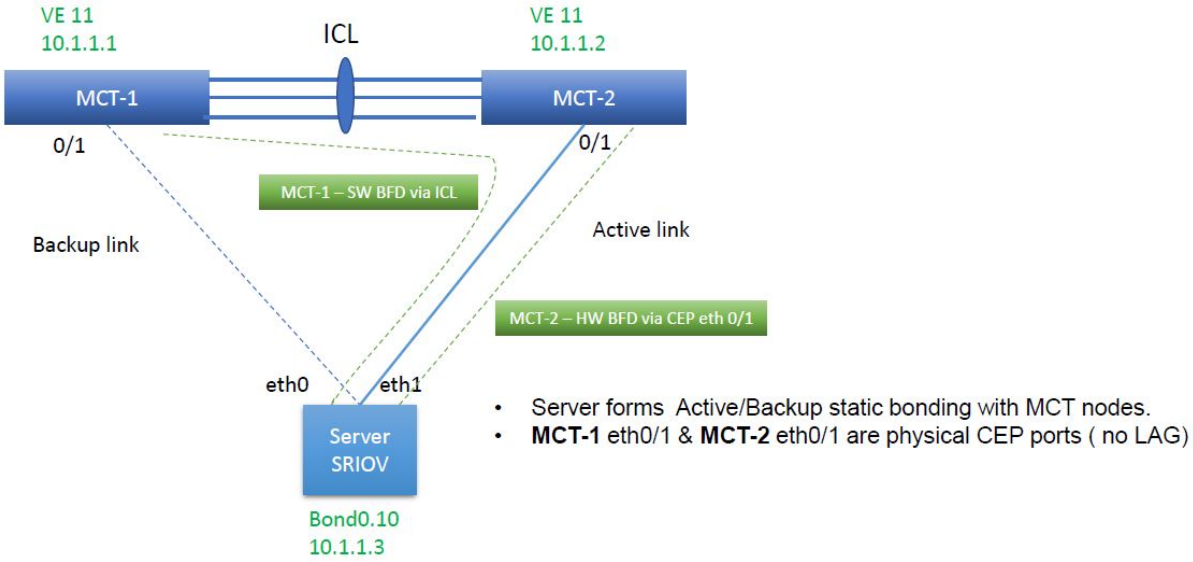
You can configure software Bidirectional Forwarding Detection (BFD) sessions on the Cluster Edge Port (CEP) on SLX 9150 and SLX 9250 platforms. `EPG : Port Property` shows the attribute `bfd-software-session`, using which you can choose a software or hardware BFD session.

BFD Session Formation with SRIOV Server

During initial state of BFD session formation with SRIOV (single-root input or output virtualization) server:

- For MCT-1:
  - The nexthop reachability for 10.1.1.3 is via ICL.
  - It forms a Software BFD session with 10.1.1.3 is via ICL. It also forms a Software BFD session with 10.1.1.3.
- For MCT-2:
  - The nexthop reachability for 10.1.1.3 is via CEP port eth 0/1.

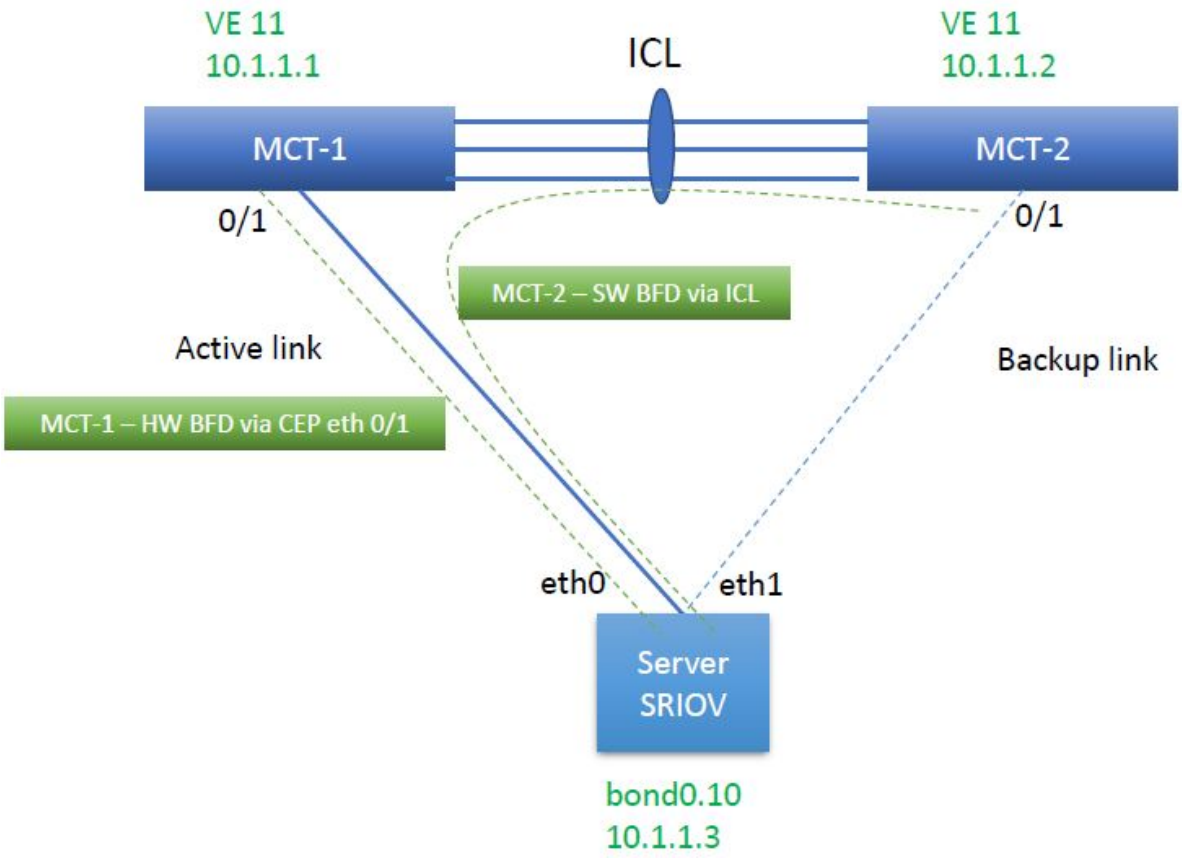
- It forms a Hardware BFD session with 10.1.1.3.



BFD Session Formation with SRIOV Server after Link Failover

During the link failover of BFD session formation with SRIOV (single-root input/output virtualization) server:

- For MCT-1:
  - The Nexthop reachability for 10.1.1.3 changes from ICL to its CEP eth 0/1.
  - The BFD session changes from Software to Hardware. BFD reachability for 10.1.1.3 changes from ICL to its CEP eth 0/1. The BFD session changes from Software to Hardware BFD.
- For MCT-2:
- ◦ The Nexthop reachability for 10.1.1.3 changes from CEP eth 0/1 to ICL.
  - The BFD session changes from Hardware to Software BFD.



BFD Session Formation with SRIOV Server

- Limitation in SLX 9250 and 8720

  Transition between software and hardware based BFD is not supported. Therefore, during session formation with SRIOV, BFD does not come up during link failover.

  Note

  - Introduce a CLI knob to change the BFD Session formed over a CEP (Ethernet or port channel) port as software BFD sessions instead of hardware BFD.
  - With the CLI knob, during the SRIOV scenario both MCT 1 and MCT 2 can form a software BFD sessions with SRIOV server.
  - During the link failover, it is SW-SW BFD session transition instead of HW-SW BFD session transition.

- SLX Configuration

```
interface Ethernet 0/1
 cluster-track
 bfd-software-session
 switchport
 switchport mode trunk
 switchport trunk allowed vlan add
11
 no switchport trunk tag native
vlan
 no shutdown
!
```

```
interface Port-channel 1
 cluster-track
 bfd-software-session
 switchport
 switchport mode trunk
 switchport trunk allowed vlan add
11
 no switchport trunk tag native
vlan
 no shutdown
!
```

cep-bfd-session-type Automation on EPG (Endpoint Group) Port Property

EFA automates the `cep-bfd-session-type` on the CEP interfaces based on the logic, with no additional input from the users.

SLX Hardware Type	EFA: Fabric Links (Leaf to Spine)	EFA: Extension EPG		EFA: L3-Handoff EPG
CEP SRIOV	CEP Non-SRIOV	CEP		
SLX 9250	Hardware	Software	Software	Hardware
SLX 8720	Hardware	Software	Software	Hardware
SLX 9740 and other SLX versions	Hardware	Hardware	Hardware	Hardware

```
(efa:root)root-2:~# efa fabric show
Fabric Name: default, Fabric Description: Default Fabric, Fabric Stage: 3, Fabric Type:
clos, Fabric Status: created
+------------+-----+-----------+-----+------+--------------+-----------
+--------------------+-----------------+---------+-------+
| IP ADDRESS | POD | HOST NAME | ASN | ROLE | DEVICE STATE | APP STATE | CONFIG GEN
REASON | PENDING CONFIGS | VTLB ID | LB ID |
+------------+-----+-----------+-----+------+--------------+-----------
+--------------------+-----------------+---------+-------+
+------------+-----+-----------+-----+------+--------------+-----------
+--------------------+-----------------+---------+-------+

Fabric Name: fs, Fabric Description: , Fabric Stage: 3, Fabric Type: clos, Fabric Status:
settings-updated

Updated Fabric Settings: BGP-LL

+-------------+-----+-----------+-------+-------+--------------------+-------------
+--------------------+-----------------------------------------------------+---------+-------+
| IP ADDRESS | POD | HOST NAME | ASN | ROLE | DEVICE STATE | APP STATE | CONFIG GEN
REASON | PENDING CONFIGS | VTLB ID | LB ID |
+-------------+-----+-----------+-------+-------+--------------------+-------------
+--------------------+-----------------------------------------------------+---------+-------+
| 10.20.246.1 | | SLX-1 | 64512 | Spine | provisioned | cfg in-sync | NA | NA | NA | 1 |
| 10.20.246.7 | | SLX | 65000 | Leaf | provisioning failed | cfg ready | IA,IU,MD,DA |
SYSP-C,MCT-C,MCT-PA,BGP-C,INTIP-C,EVPN-C,O-C | 2 | 1 |
| 10.20.246.8 | | slx-8 | 65000 | Leaf | provisioned | cfg in-sync | NA | NA | 2 | 1 |
+-------------+-----+-----------+-------+-------+--------------------+-------------
+--------------------+-----------------------------------------------------+---------+-------+
FABRIC SETTING:
BGPLL - BGP Dynamic Peer Listen Limit, BGP-MD5 - BGP MD5 Password , BFD-RX - Bfd Rx
```

```
Timer, BFD-TX - Bfd Tx Timer, BFD-MULTIPLIER - Bfd multiplier,
BFD-ENABLE - Enable Bfd, BGP-MULTIHOP - BGP ebgp multihop, P2PLR - Point-to-Point Link
Range, MCTLR - MCT Link Range, LOIP - Loopback IP Range


CONFIG GEN REASON:
LA/LD - Link Add/Delete, IA/ID/IU - Interface Add/Delete/Update, PLC/PLD/PLU -
IPPrefixList Create/Delete/Update
MD/MU - MCT Delete/Update, OD/OU - Overlay Gateway Delete/Update, EU/ED - Evpn Delete/
Update, PC/PD/PU - RouterPim Create/Delete/Update
DD - Dependent Device Update, DA/DR - Device Add/ReAdd, ASN - Asn Update, SYS - System
Properties Update
MD5 - BGP MD5 Password, BGPU - Router BGP Update, BGPLL - BGP Listen Limit, POU - Port
Channel Update, NA - Not Applicable

PENDING CONFIGS:
MCT - MCT Cluster, O - Overlay Gateway, SYSP - System Properties, INTIP - Interface IP,
BGP - Router BGP
C/D/U - Create/Delete/Update, PA/PD - Port Add/Port Delete
(efa:root)root-2:-# efa tenant show
+------+---------+------------+------------ +-------------+-----------+------------
+----------------------+
| Name |   Type  | VLAN Range | L2VNI Range | L3VNI Range | VRF Count |  Enable BD
|         Ports        |
+------+---------+------------+-------------+-------------+-----------+------------
+----------------------+
| ten1 | Private |    11-20   |             |             |     10    |    false   |
10.20.246.6[0/1-10]| |
|      |         |            |             |             |           |            |
10.20.246.5[0/1-10]| |
+------+---------+------------+-------------+-------------+-----------+------------
+----------------------+

efa tenant vrf create --name ten1vrf1 --tenant ten1
```

EPG Create

Run the following command to create a `cep-bfd-session-type` automation on EPG port property:

```
efa tenant epg create --name ten1epg1 --tenant ten1
        --port 10.20.246.5[0/1],10.20.246.6[0/1]
        --switchport-mode trunk
        --vrf ten1vrf1 --ctag-range 11
        --anycast-ip 11:20.0.11.1/24
        --local-ip 11,10.20.246.5:10.1.1.1/24 --local-ip 11,10.20.246.6:10.1.1.2/24
```

Example

```
(efa:root)root@node-2:-# efa tenant epg show  --detail
================================================================
Name          : ten1epg1
Tenant        : ten1
Type          : extension
State         :
Description   :

Ports         : 10.20.246.5[0/1]
              : 10.20.246.6[0/1]
POs           :
Port Property : SwithchPort Mode    : trunk
              : Native Vlan Tagging : false
              : BFD Session Type    : Auto
NW Policy     : Ctag Range         : 11
```

```
              : VRF                : ten1vrf1
              : L3Vni              : 8192
+------+-------------------------+-------+---------+--------------+--------------
+-------------------------+---------+---------------+--------------+-------------
+-------------+
| Ctag |            Ctag         | L2Vni | BD Name | Anycast IPv4 | Anycast IPv6
|         Local IP         | IPv6 ND |     IPv6 ND    |    IPv6 ND   |  Dev State  |  App
State  |
|      |        Description      |       |         |              |              |
[Device-IP->Local-IP]   |   Mtu   | Managed Config | Other Config |
|             |
+------+-------------------------+-------+---------+--------------+--------------
+----------------------+------------+---------------+--------------+-------------
+-------------+
|  11  | Tenant L3 Extended VLAN |  11   |         | 20/0.11.1/24 |              |
10.20.246.5->10.1.1.1/24 |         |      false     |     false    | provisioned | cfg-in-
sync |
|      |                         |       |         |              |              |
10.20.246.6->10.1.1.1/24 |         |                |              |
|             |
+------+-------------------------+-------+---------+--------------+--------------
+-------------------------+---------+---------------+--------------+-------------
+-------------+
Network Property [Flags : * - Native Vlan]
+------+----------------+--------------+----------------+--------------------
+-------------+
| CTAG | IPv6 ND Prefix | No Advertise | Valid Lifetime | Preferred Lifetime | Config
Type |
+------+----------------+--------------+----------------+--------------------
+-------------+
IPv6 ND Prefix Flags

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

VRF Update

Run the following command to update a tenant VRF on the static route BFD:

```
efa tenant vrf update --name ten1vrf1 --tenant ten1
        --operation static-route-bfd-add
        --ipv4-static-route-bfd 10.20.246.5,10.1.1.3,10.1.1.1
        --ipv4-static-route-bfd 10.20.246.6,10.1.1.3,10.1.1.2
```

Example

```
(efa:root)root@node-2:-# efa tenant vrf show  --detail
================================================================
Name                    : ten1vrf1
Tenant                  : ten1
Type                    : extension
Routing Type            : distributed
Centralized Routers     :
Redistribute            : connected
Max Path                : 8
Local ASN               :
L3VNI                   : 8192
EVPN IRB BD             : 4096
EVPN IRB VE             : 8192
BR VNI                  : 4096
BR BD                   :
BR VE                   :
RH Max Path             :
Enable RH ECMP          : false
```

```
Enable Graceful Restart : false
Route Target            : import 101:101
                        : export 101:101
Static Route            :
Static Rout BFD         : Switch-IP->[DestIP,SourceIP][Interval,Min-Rx,Multiplier], ...
                        : 10.20.246.6->10.1.1.3,10.1.1.2
                        : 10.20.246.5->10.1.1.3,10.1.1.1
State                   : vrf-device-created
Dev State               : provisioned
App State               : cfg-in-sync

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

Switch Config

```
Rack1-Device1(config)# do show
running-config vlan 11
vlan 11
 router interface Ve 11
 suppress-arp
 description Tenant L3 Extended
VLAN
!
Rack1-Device1(config)# do show
running config interface Ve 11
interface Ve 11
 vrf forwarding ten1vrf1
 ip anycast address 20.0.11.1/24
 ip address 10.1.1.1/24
 no shutdown
!
Rack1-Device1(config)# do show
running config interface Ethernet
0/1
interface Ethernet 0/1
 cluster-track
 bfd software session
 switchport
 switchport mode trunk
 switchport trunk allowed vlan add
11
 no switchport trunk tag native-
vlan
 no shutdown
!
Rack1-Device1(config)# do show
running config vrf ten1vrf1
address family
ipv4 unicast
vrf ten1vrf1
 address family ipv4 unicast
  route target export 101:101 evpn
  route target import 101:101 evpn
  ip route static bfd 10.1.1.3
10.1.1.1
 !
!
Rack1-Device1(config)#
```

```
Rack1-Device2(config)# do show
running config vlan 11
vlan 11
 router interface Ve 11
 suppress arp
 description Tenant L3 Extended
VLAN
!
Rack1Device2(config)# do show
running config in Ve 11
interface Ve 11
 vrf forwarding ten1vrf1
 ip anycast address 20.0.11.1/24
 ip address 10.1.1.2/24
 no shutdown
!
Rack1Device2(config)# do show
running config int eth 0/1
interface Ethernet 0/1
 cluster track
 bfd software session
 switchport
 switchport mode trunk
 switchport trunk allowed vlan add
11
 no switchport trunk tag native-
vlan
 no shutdown
!
Rack1-Device2(config)# do show
running config vrf ten1vrf1
address family
ipv4 unicast
vrf ten1vrf1
 address family ipv4 unicast
  route target export 101:101 evpn
  route target import 101:101 evpn
  ip route static bfd 10.1.1.3
10.1.1.2
 !
!
Rack1-Device2(config)#
```

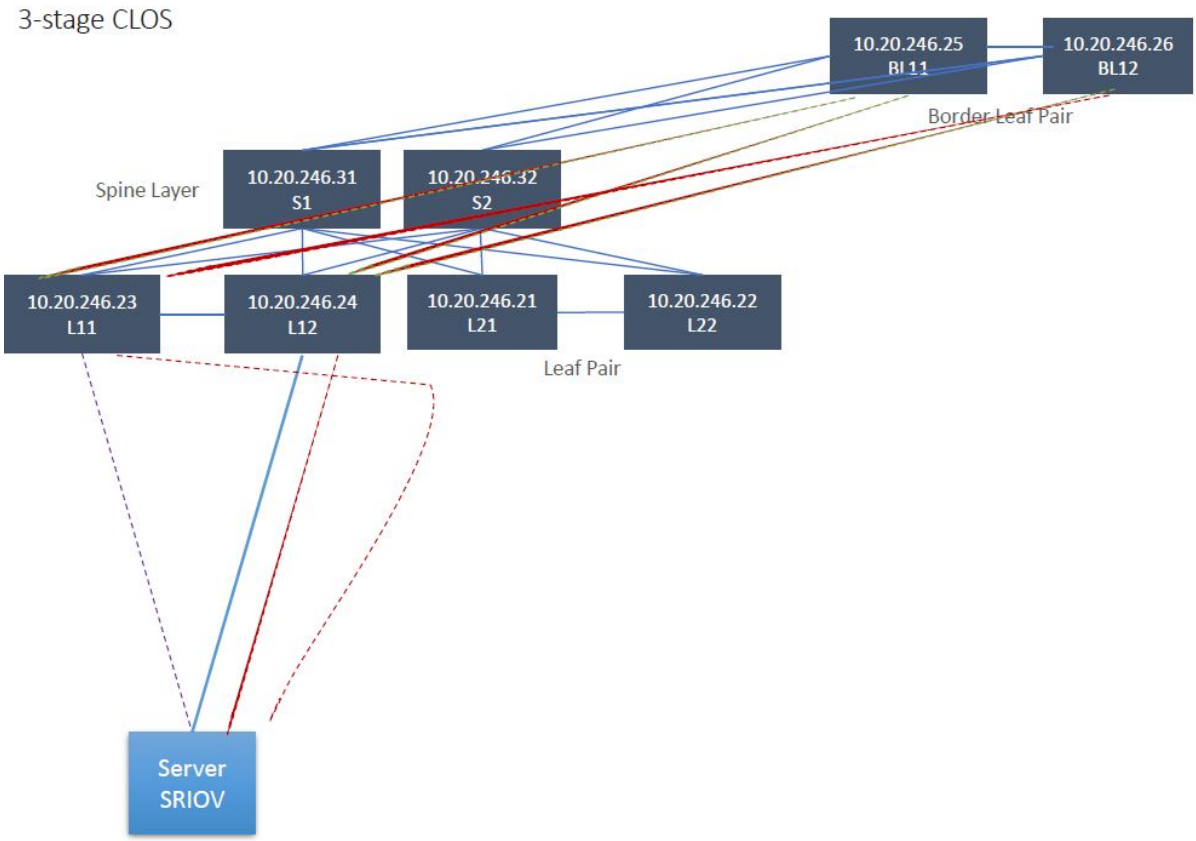
cep-bfd-session-type on EPG Port Property

- The `cep-bfd-session-type` feature enables you to provide `cep bfd session type` per EPG which gets configured on all the ethernet and single homed port channel interfaces defined in the EPG.
- The default value of `cep bfd session type` is set to “auto”. EFA automatically derives the appropriate `cep bfd session type` based on the use case (extension or l3 hand off) and endpoint type.
- You can provide the `cep bfd session type` configuration during EPG create and EPG update port group add operations.

EPG Create	EPG Update
`efa tenant epg create --name < epg-name> --tenant <tenant-name> --port <port-list> po <po-list> --switchport-mode <access | trunk>` **`--cep-bfd-session-type {auto | software | hardware}`**	`efa tenant epg update --name < epg-name> tenant <tenant-name>` **`--operation port group add`** `--port <port-list> --po <po-list> switchport-mode < access|trunk >` **`--cep-bfd-session-type {auto | software | hardware}`**

CEP SRIOV and Non SRIOV	Upgrade Handling
• EFA cannot distinguish between the SRIOV and Non-SRIOV connections. Hence both the CEP SRIOV and CEP Non-SRIOV phy or port channel are treated in same manner. • To use the “hardware” bfd sessions for the CEP Non-SRIOV connections, create an EPG containing all the Non-SRIOV CEP with `cep bfd session type=hardware`.	During the upgrade from EFA 2.4.0 to 2.4.1 and onwards, all the CEP ports (on SLX 9250 and SLX 8720 platforms) used in the “extension” EPG must have `cep bfd session type` as `software`, and you must perform an explicit DRC to reconcile the EFA configuration to synchronize with the SLX.

Co-existence of centralize and distributed routing on a CEP



Exclusion of VLANs and Bridge from Cluster Instance

EFA excludes the VLANs and bridge domains used in the Layer 3 hand-off (toward the external gateway) endpoint group from the cluster instance by configuring `member vlan remove <vlan-range>` and `member bridge-domain remove <bd-range>` under the cluster instance.

During EFA upgrade, EFA marks all the VLANs and Bridge Domains (BD) used in `l3-hand-off` EPGs with the intended `member vlan remove <vlan-range>` and `member bridge-domain remove <bd-range>` configuration and shows as configuration drift. On reconciliation of the drift, EFA pushes `member vlan remove <vlan-range>` and `member bridge-domain remove <bd-range>` configuration under the cluster.

EFA Provisioning

```
# efa tenant create --name tenant1 --port 10.24.80.134[0/1-10],10.24.80.135[0/1-10]
--vlan-range 2001-2010

# efa tenant po create --name po1 --tenant tenant1 --port
10.24.80.134[0/1],10.24.80.135[0/1]
--speed 10Gbps --negotiation active

# efa tenant epg create --name L3HandoffEPG1Ten1 --tenant tenant1 --ctag-range 2001-2003
--switchport-mode trunk --po po1 --type l3-hand-off
Device1 # show run interface Port-channel 1

interface Port-channel 1
```

```
cluster-client auto
switchport
switchport mode trunk
switchport trunk allowed vlan add 2001-2003
no switchport trunk tag native-vlan
no shutdown
!
Device1# show running config-evpn
evpn-fabric1
route-target both auto ignore-as
rd auto
duplicate-mac-timer 5 max-count 3
!
Device1# show running-config cluster
cluster fabric1-cluster-1
peer 10.20.20.5
peer-interface Port-channel 64
peer-keepalive
auto
!
member vlan-all
member vlan remove 2001-2003
member bridge-domain all
!
Device2 # show run interface Port-channel 1

interface Port-channel 1
cluster-client auto
switchport
switchport mode trunk
switchport trunk allowed vlan add 2001-2003
no switchport trunk tag native-vlan
no shutdown
!
Device2# show running config-evpn
evpn-fabric1
route-target both auto ignore-as
rd auto
duplicate-mac-timer 5 max-count 3
!
Device2# show running-config cluster
cluster fabric1-cluster-1
peer 10.20.20.5
peer-interface Port-channel 64
peer-keepalive
auto
!
member vlan-all
member vlan remove 2001-2003
member bridge-domain all
!
```

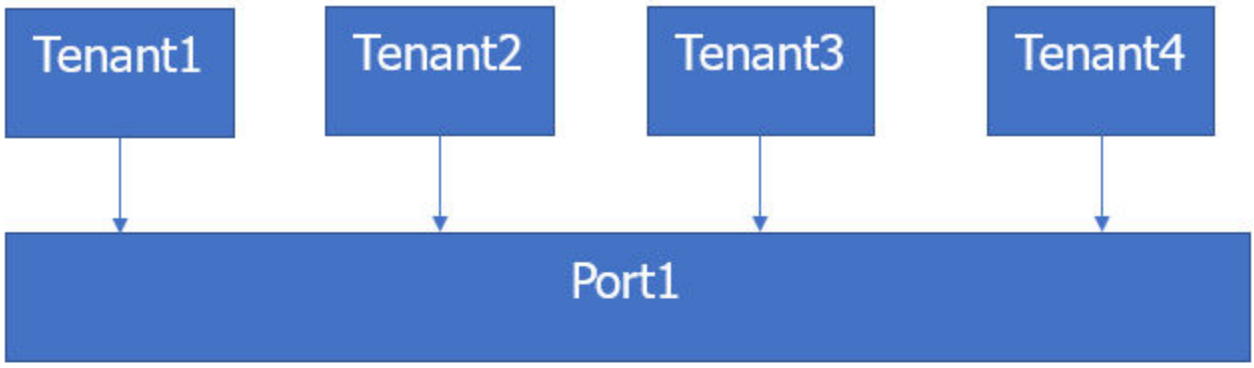
Sharing Resources Across Tenants using Shared Tenant

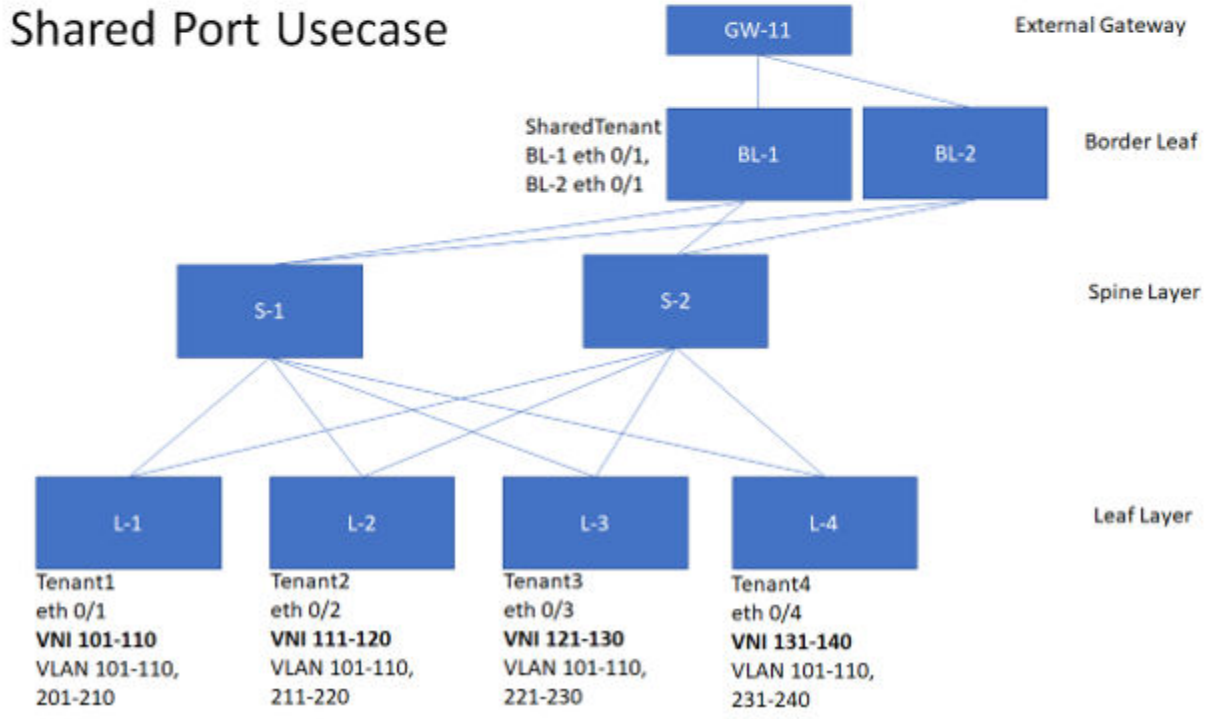
This topic offers examples of configuring shared tenant resources.

One tenant, with the `role=shared` attribute, owns the resources and entities that can be shared across all the other tenants, called non-Shared Tenant. The tenant service can have one Shared Tenant that services all the shared resources. The Shared Tenant owns the physical ports, Layer 2 and Layer 3 VNI number ranges, VLAN number ranges, and the VRF numbers. The Shared Tenant can create the endpoints and the VRFs, but not the endpoint groups.

A non-Shared Tenant cannot use the ports that the Shared Tenant owns if the ports are already part of an endpoint. A non-Shared Tenant cannot create an endpoint using the ports that the Shared Tenant owns.

Example: Shared Port use case (Layer 2 hand-off)





The following examples show the commands and syntax used to configure the Shared Port.

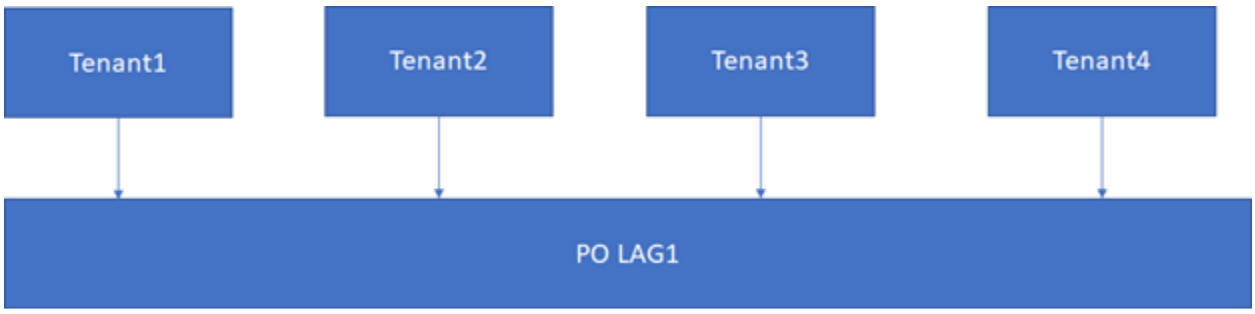
```
efa tenant create --name tenant1 --l2-vni-range 101-110 --vlan-range 101-110,201-210
--port L-1[0/1]
efa tenant create --name tenant2 --l2-vni-range 111-120 --vlan-range 101-110,211-220
--port L-2[0/2]
efa tenant create --name tenant3 --l2-vni-range 121-130 --vlan-range 101-110,221-230
--port L-3[0/3]
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] --type 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
```

Example: Shared Endpoint use case (Layer 2 hand-off)



The following examples show the commands and syntax used to configure the Shared Endpoint.

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

Example: Shared Endpoint use case (Layer 3 hand-off)

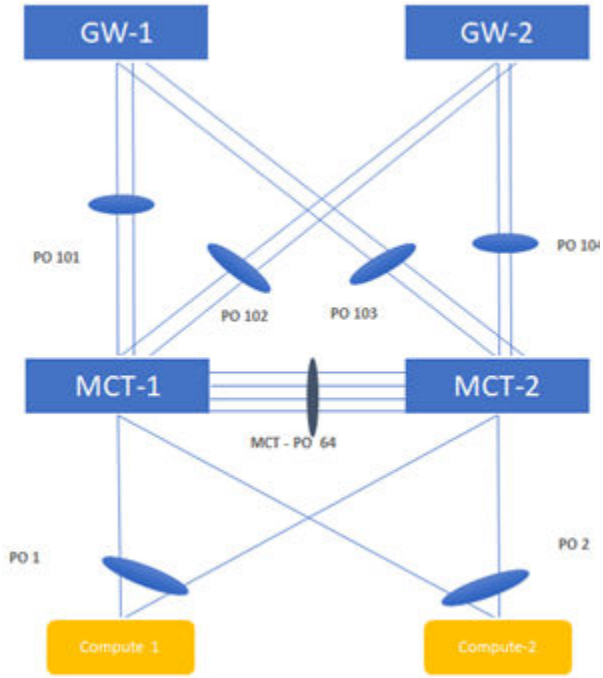


Figure 19: Topology

The following examples show the commands and syntax used to configure the Shared Endpoint.

```
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] --type 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
```

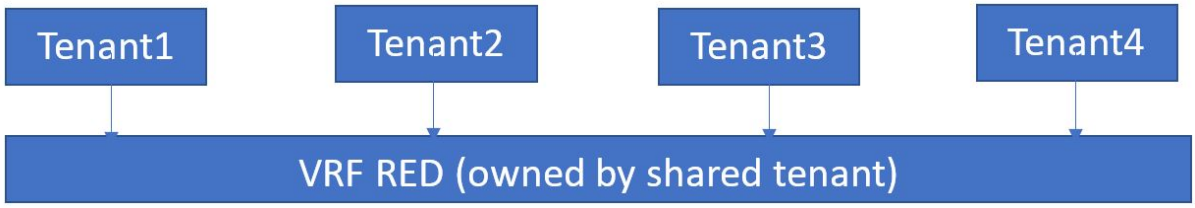
Shared VRF and Router

EFA provides a provisioning model to support sharing of VRF or Router across multiple tenants. Following models are supported:

- Inter-POD (Inter-Tenant) routing (Tenant1 routing to Tenant2 and vice versa)

- Multiple tenants accessing a shared resource (for example, storage)
- Multiple tenants using a shared VRF for l3-hand-off

Entities (VRFs) created by the shared tenant are available for the use by all the Private Tenants.



Configure Shared Tenant, Shared VRF, and Private EPG using Shared VRF

1. Configure Shared Tenant.

```
efa tenant create --name <epg-name> --type shared --port <port-list>
                  --vrf-count <num-of-vrfs> --l3-vni-range <l3-vni-range>
                  --vlan-range <vlan-range> --l2-vni-range <l2-vni-range>
```

2. Configure Shared VRF under the ownership of Shared Tenant.

```
efa tenant vrf create --name <vrf-name> --tenant <shared-tenant-name>
```

3. Configure endpoint group (EPG) under the ownership of Private Tenant using the Shared VRF.

```
efa tenant epg create --name <epg-name> --tenant <private-tenant>
                --po <po-list> --switchport-mode <trunk|access> --ctag-range <ctag-
range>
                --anycast-ip <ctag:anycast-ip> --vrf <shared-vrf-owned-by-shared-
tenant>
```

Configure L3-Hand-Off EPG and BGP Peer under Ownership of Shared Tenant

L3-Hand-Off endpoint groups (EPG) created under the ownership of Shared Tenant are exclusively meant for the handoff of the Shared VRF. You cannot create the Extension EPGs under the ownership of Shared Tenant. BGP peer created under the ownership of Shared Tenant are exclusively meant for the handoff of the Shared VRF.

1. Configure Shared Tenant.

```
efa tenant create --name <epg-name> --type shared --port <port-list>
                --vrf-count <num-of-vrfs> --l3-vni-range <l3-vni-range>
                --vlan-range <vlan-range> --l2-vni-range <l2-vni-range>
```

2. Configure Shared VRF under the ownership of Shared Tenant.

```
efa tenant vrf create --name <vrf-name> --tenant <shared-tenant-name>
```

3. Configure L3-hand-off EPG under the ownership of Shared Tenant using the Shared VRF.

```
efa tenant epg create --name <epg-name> --type l3-hand-off --tenant <shared-tenant-
name-owning-the-shared-vrf>
                --po <po-list> --switchport-mode <trunk|access> --ctag-range <ctag-
range>
                --local-ip <ctag,device-ip:local-ip> --vrf <shared-vrf-owned-by-
shared-tenant>
```

4. Configure BGP peer under the ownership of Shared Tenant using the Shared VRF.

```
efa tenant service bgp peer create --name <bgp-peer-name> --tenant <shared-tenant-
owning-the-shared-vrf>
                --ipv4-uc-nbr <device-ip,shared-vrf-owned-by-shared-tenant:bgp-peer-
ip,bgp-peer-remote-asn>
```

Shared VRF and Router Usecase with Examples

Topology

```
efa fabric create --name fabric1 --type non-clos

efa fabric setting update --name fabric1
           --vni-auto-map No --backup-routing-enable Yes

efa fabric device add-bulk --name fabric1
           --rack rack1 --ip 10.20.246.25-26 --rack rack2 --ip 10.20.246.17-18
           --border-leaf-rack rack3 --border-leaf-ip 10.20.246.15-16
           --username admin --password password

efa fabric configure --name fabric1

efa fabric show

Fabric Name: default, Fabric Description: Default Fabric, Fabric Stage: 3, Fabric Type:
clos, Fabric Status: created
+------------+-----+-----------+-----+------+--------------+-----------
+-------------------+-----------------+---------+-------+
| IP ADDRESS | POD | HOST NAME | ASN | ROLE | DEVICE STATE | APP STATE | CONFIG GEN
REASON | PENDING CONFIGS | VTLB ID | LB ID |
+------------+-----+-----------+-----+------+--------------+-----------
+-------------------+-----------------+---------+-------+
+------------+-----+-----------+-----+------+--------------+-----------
+-------------------+-----------------+---------+-------+

Fabric Name: fs, Fabric Description: , Fabric Stage: 3, Fabric Type: clos, Fabric Status:
settings-updated

Updated Fabric Settings: BGP-LL

+-------------+-----+-----------+-------+-------+--------------------+-------------
+-------------------+-----------------------------------------------------+---------+-------+
| IP ADDRESS | POD | HOST NAME | ASN | ROLE | DEVICE STATE | APP STATE | CONFIG GEN
REASON | PENDING CONFIGS | VTLB ID | LB ID |
+-------------+-----+-----------+-------+-------+--------------------+-------------
+-------------------+-----------------------------------------------------+---------+-------+
| 10.20.246.1 | | SLX-1 | 64512 | Spine | provisioned | cfg in-sync | NA | NA | NA | 1 |
| 10.20.246.7 | | SLX | 65000 | Leaf | provisioning failed | cfg ready | IA,IU,MD,DA |
SYSP-C,MCT-C,MCT-PA,BGP-C,INTIP-C,EVPN-C,O-C | 2 | 1 |
| 10.20.246.8 | | slx-8 | 65000 | Leaf | provisioned | cfg in-sync | NA | NA | 2 | 1 |
+-------------+-----+-----------+-------+-------+--------------------+-------------
+-------------------+-----------------------------------------------------+---------+-------+
FABRIC SETTING:
BGPLL - BGP Dynamic Peer Listen Limit, BGP-MD5 - BGP MD5 Password , BFD-RX - Bfd Rx
Timer, BFD-TX - Bfd Tx Timer, BFD-MULTIPLIER - Bfd multiplier,
BFD-ENABLE - Enable Bfd, BGP-MULTIHOP - BGP ebgp multihop, P2PLR - Point-to-Point Link
Range, MCTLR - MCT Link Range, LOIP - Loopback IP Range


CONFIG GEN REASON:
LA/LD - Link Add/Delete, IA/ID/IU - Interface Add/Delete/Update, PLC/PLD/PLU -
IPPrefixList Create/Delete/Update
MD/MU - MCT Delete/Update, OD/OU - Overlay Gateway Delete/Update, EU/ED - Evpn Delete/
Update, PC/PD/PU - RouterPim Create/Delete/Update
```

```
DD - Dependent Device Update, DA/DR - Device Add/ReAdd, ASN - Asn Update, SYS - System
Properties Update
MD5 - BGP MD5 Password, BGPU - Router BGP Update, BGPLL - BGP Listen Limit, POU - Port
Channel Update, NA - Not Applicable

PENDING CONFIGS:
MCT - MCT Cluster, O - Overlay Gateway, SYSP - System Properties, INTIP - Interface IP,
BGP - Router BGP
C/D/U - Create/Delete/Update, PA/PD - Port Add/Port Delete
```

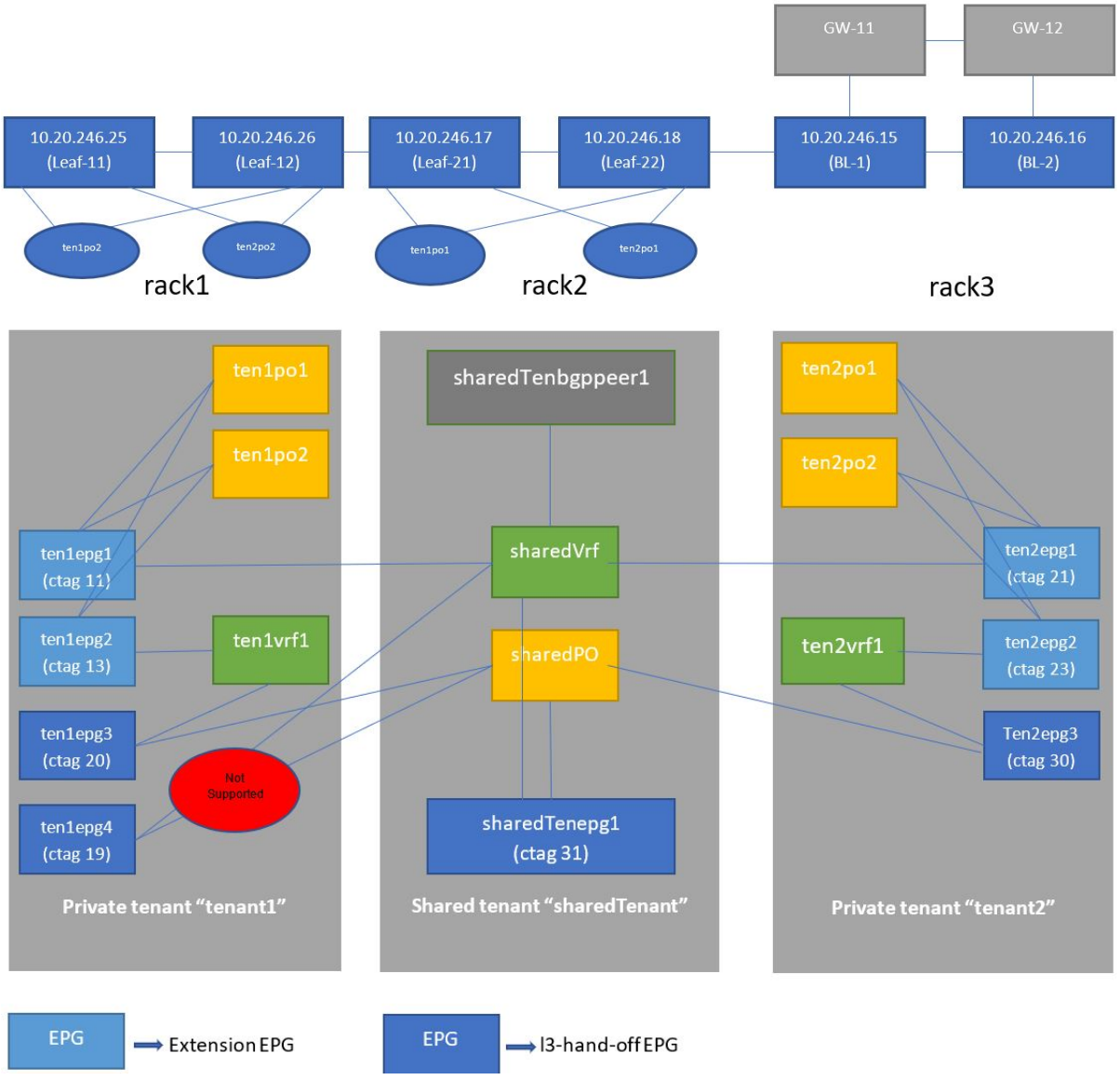


Figure 20: Shared VRF Configuration Overview

Shared Tenant and Private Tenant Configuration

```
efa tenant create --name sharedTenant --type shared --port
10.20.246.15[0/31],10.20.246.16[0/31]
    --vrf-count 10 --l3-vni-range 31001-31020
    --vlan-range 31-40 --l2-vni-range 30011-30020

efa tenant create --name tenant1 --port
10.20.246.17[0/11-20],10.20.246.18[0/11-20],10.20.246.25[0/11-20],10.20.246.26[0/11-20]
    --vlan-range 11-20 --l2-vni-range 10011-10020 --vrf-count 10 --l3-vni-range
```

```
11001-11020

efa tenant create --name tenant2 --port
10.20.246.17[0/21-30],10.20.246.18[0/21-30],10.20.246.25[0/21-30],10.20.246.26[0/21-30]
    --vlan-range 21-30 --l2-vni-range 20011-20020 --vrf-count 10 --l3-vni-range
21001-21020

efa tenant show
+--------------+---------+------------+-------------+-------------+-----------+-----------
+-----------------------+
|     Name     |  Type   | VLAN Range | L2VNI Range | L3VNI Range | VRF Count | Enable BD
|         Ports         |
+--------------+---------+------------+-------------+-------------+-----------+-----------
+-----------------------+
| sharedTenant | shared  |   31-40    | 30011-30020 | 31001-31020 |    10     |   false
|  10.20.246.15[0/31]   |
|              |         |            |             |             |           |
|  10.20.246.16[0/31]   |
+--------------+---------+------------+-------------+-------------+-----------+-----------
+-----------------------+
|   tenant1    | private |   11-20    | 10011-10020 | 11001-11020 |    10     |   false
| 10.20.246.18[0/11-20] |
|              |         |            |             |             |           |
| 10.20.246.17[0/11-20] |
|              |         |            |             |             |           |
| 10.20.246.25[0/11-20] |
|              |         |            |             |             |           |
| 10.20.246.26[0/11-20] |
+--------------+---------+------------+-------------+-------------+-----------+-----------
+-----------------------+
|   tenant2    | private |   21-30    | 20011-20020 | 21001-21020 |    10     |   false
| 10.20.246.26[0/21-30] |
|              |         |            |             |             |           |
| 10.20.246.18[0/21-30] |
|              |         |            |             |             |           |
| 10.20.246.17[0/21-30] |
|              |         |            |             |             |           |
| 10.20.246.25[0/21-30] |
+--------------+---------+------------+-------------+-------------+-----------+-----------
+-----------------------+
```

Shared PO and Private PO Configuration

```
efa tenant po create --name sharedPO --tenant sharedTenant
                  --port 10.20.246.15[0/31],10.20.246.16[0/31] --speed 10Gbps --
negotiation active

efa tenant po create --name ten1po1 --tenant tenant1
                  --port 10.20.246.17[0/11],10.20.246.18[0/11] --speed 10Gbps --
negotiation active

efa tenant po create --name ten1po2 --tenant tenant1
                  --port 10.20.246.25[0/11],10.20.246.26[0/11] --speed 10Gbps --
negotiation active
efa tenant po create --name ten2po1 --tenant tenant2
                  --port 10.20.246.17[0/21],10.20.246.18[0/21] --speed 10Gbps --
negotiation active
efa tenant po create --name ten2po2 --tenant tenant2
                  --port 10.20.246.25[0/21],10.20.246.26[0/21] --speed 10Gbps --
negotiation active
efa tenant po show
+----------+--------------+----+--------+-------------+----------+---------
+--------------------+------------+-------------+-------------+
|   Name   |    Tenant    | ID | Speed  | Negotiation | Min Link |  Lacp   |
Ports        |   State    |  Dev State  |  App State  |
```

```
|          |              |    |        |             |  Count   | Timeout
|                    |            |             |             |
+----------+--------------+----+--------+-------------+----------+---------
+--------------------+------------+-------------+-------------+
| sharedPO | sharedTenant | 1  | 10Gbps |   active    |    1     |  long   |
10.20.246.16[0/31] | po-created | provisioned | cfg-in-sync |
|          |              |    |        |             |          |         |
10.20.246.15[0/31] |            |             |             |
+----------+--------------+----+--------+-------------+----------+---------
+--------------------+------------+-------------+-------------+
| ten1po1  |   tenant1    | 1  | 10Gbps |   active    |    1     |  long   |
10.20.246.18[0/11] | po-created | provisioned | cfg-in-sync |
|          |              |    |        |             |          |         |
10.20.246.17[0/11] |            |             |             |
+----------+--------------+----+--------+-------------+----------+---------
+--------------------+------------+-------------+-------------+
| ten1po2  |   tenant1    | 1  | 10Gbps |   active    |    1     |  long   |
10.20.246.25[0/11] | po-created | provisioned | cfg-in-sync |
|          |              |    |        |             |          |         |
10.20.246.26[0/11] |            |             |             |
+----------+--------------+----+--------+-------------+----------+---------
+--------------------+------------+-------------+-------------+
| ten2po1  |   tenant2    | 2  | 10Gbps |   active    |    1     |  long   |
10.20.246.18[0/21] | po-created | provisioned | cfg-in-sync |
|          |              |    |        |             |          |         |
10.20.246.17[0/21] |            |             |             |
+----------+--------------+----+--------+-------------+----------+---------
+--------------------+------------+-------------+-------------+
| ten2po2  |   tenant2    | 2  | 10Gbps |   active    |    1     |  long   |
10.20.246.25[0/21] | po-created | provisioned | cfg-in-sync |
|          |              |    |        |             |          |         |
10.20.246.26[0/21] |            |             |             |
+----------+--------------+----+--------+-------------+----------+---------
+--------------------+------------+-------------+-------------+
```

Shared VRF and Private VRF

```
efa tenant vrf create --name sharedVrf --tenant sharedTenant

efa tenant vrf create --name ten1vrf1 --tenant tenant1

efa tenant vrf create --name ten2vrf1 --tenant tenant2

efa tenant vrf show

+-----------+--------------+--------------+---------------------+---------------+----------
+-----------+-----------+------------+-----------------+-----------+
|   Name    |    Tenant    | Routing Type | Centralized Routers | Redistribute | Max Path
| Local Asn | Enable GR |   State    |    Dev State    | App State |
+-----------+--------------+--------------+---------------------+---------------+----------
+-----------+-----------+------------+-----------------+-----------+
| sharedVrf | sharedTenant | distributed  |                     |  connected   |    8
|           |   false   | vrf-create | not-provisioned | cfg-ready |
+-----------+--------------+--------------+---------------------+---------------+----------
+-----------+-----------+------------+-----------------+-----------+
| ten1vrf1  |   tenant1    | distributed  |                     |  connected   |    8
|           |   false   | vrf-create | not-provisioned | cfg-ready |
+-----------+--------------+--------------+---------------------+---------------+----------
+-----------+-----------+------------+-----------------+-----------+
| ten2vrf1  |   tenant2    | distributed  |                     |  connected   |    8
|           |   false   | vrf-create | not-provisioned | cfg-ready |
+-----------+--------------+--------------+---------------------+---------------+----------
+-----------+-----------+------------+-----------------+-----------+
```

Shared VRF: Inter POD Routing

- Endpoint groups (EPGs) owned by different Private Tenants using the shared VRF

```
efa tenant epg create --name ten1epg1 --tenant tenant1 --po ten1po1,ten1po2 --
switchport-mode trunk
                --ctag-range 11 --anycast-ip 11:10.0.11.1/24 –vrf sharedVrf
efa tenant epg create --name ten2epg1 --tenant tenant2 --po ten2po1,ten2po2 --
switchport-mode trunk
        --ctag-range 21 --anycast-ip 21:10.0.21.1/24 --vrf sharedVrf
```

- EPGs owned by different Private Tenants using their own private VRF:

```
efa tenant epg create --name ten1epg2 --tenant tenant1 --po ten1po1,ten1po2 --
switchport-mode trunk
                --ctag-range 13 --anycast-ip 13:10.0.13.1/24 --vrf ten1vrf1
efa tenant epg create --name ten2epg2 --tenant tenant2 --po ten2po1,ten2po2 --
switchport-mode trunk
        --ctag-range 23 --anycast-ip 23:10.0.23.1/24 --vrf ten2vrf1
```

```
efa tenant epg show
+----------+---------+-----------+-------+---------+------------+-------------
+------------+-----------+-------+-------+
|   Name   | Tenant  |   Type    | Ports |   PO    | SwitchPort | Native Vlan | Ctag
Range |    Vrf    | L3Vni | State |
|          |         |           |       |         |    Mode    |   Tagging
|            |           |       |       |
+----------+---------+-----------+-------+---------+------------+-------------
+------------+-----------+-------+-------+
| ten1epg1 | tenant1 | extension |       | ten1po2 |   trunk    |    false    |
11     | sharedVrf | 31001 |       |
|          |         |           |       | ten1po1 |            |
|            |           |       |       |
+----------+---------+-----------+-------+---------+------------+-------------
+------------+-----------+-------+-------+
| ten1epg2 | tenant1 | extension |       | ten1po1 |   trunk    |    false    |
13     | ten1vrf1  | 11001 |       |
|          |         |           |       | ten1po2 |            |
|            |           |       |       |
+----------+---------+-----------+-------+---------+------------+-------------
+------------+-----------+-------+-------+
| ten2epg1 | tenant2 | extension |       | ten2po1 |   trunk    |    false    |
21     | sharedVrf | 31001 |       |
|          |         |           |       | ten2po2 |            |
|            |           |       |       |
|          |         |           |       |         |            |
|            |           |       |       |
+----------+---------+-----------+-------+---------+------------+-------------
+------------+-----------+-------+-------+
| ten2epg2 | tenant2 | extension |       | ten2po2 |   trunk    |    false    |
23     | ten2vrf1  | 21001 |       |
|          |         |           |       | ten2po1 |            |
|            |           |       |       |
|          |         |           |       |         |            |
|            |           |       |       |
+----------+---------+-----------+-------+---------+------------+-------------
+------------+-----------+-------+-------+
```

L3 Handoff Using EPG and BGP Peer Owned by Shared Tenant Using Shared VRF

- Endpoint Group (EPG) owned by Shared Tenant handling off Shared VRF

```
efa tenant epg create --name sharedTenepg1 --tenant sharedTenant --type l3-hand-off
                    --po sharedPO --switchport-mode trunk --ctag-range 31
                    --vrf sharedVrf --local-ip 31,10.20.246.15:10.0.31.2/24 --local-ip
31,10.20.246.16:10.0.31.2/24
```

- BGP peer owned by Shared Tenant handling off Shared VRF

```
efa tenant service bgp peer create --name sharedTenbgppeer1 --tenant sharedTenant
                    --ipv4-uc-nbr
10.20.246.15,sharedVrf:10.0.31.3,50000
                    --ipv4-uc-nbr 10.20.246.16,sharedVrf:10.0.31.3,50000
```

```
efa tenant epg show
+---------------+--------------+-------------+-------+----------+------------
+-------------+------------+-----------+-------+-------+
|     Name      |    Tenant    |    Type     | Ports |    PO    | SwitchPort | Native
Vlan | Ctag Range |    Vrf    | L3Vni | State |
|               |              |             |       |          |    Mode    |
Tagging   |            |           |       |       |
+---------------+--------------+-------------+-------+----------+------------
+-------------+------------+-----------+-------+-------+
| sharedTenepg1 | sharedTenant | l3-hand-off |       | sharedPO |   trunk    |
false    |     31     | sharedVrf | 31001 |       |
|               |              |             |       |          |
|             |            |           |       |       |
+---------------+--------------+-------------+-------+----------+------------
+-------------+------------+-----------+-------+-------+
|   ten1epg2    |   tenant1    |  extension  |       | ten1po2  |   trunk    |
false    |     13     | ten1vrf1  | 11001 |       |
|               |              |             |       | ten1po1  |
|             |            |           |       |       |
+---------------+--------------+-------------+-------+----------+------------
+-------------+------------+-----------+-------+-------+
|   ten1epg3    |   tenant1    | l3-hand-off |       | sharedPO |   trunk    |
false    |     20     | ten1vrf1  | 11001 |       |
|               |              |             |       |          |
|             |            |           |       |       |
+---------------+--------------+-------------+-------+----------+------------
+-------------+------------+-----------+-------+-------+
|   ten1epg4    |   tenant1    | l3-hand-off |       | sharedPO |   trunk    |
false    |     19     | sharedVrf | 31001 |       |
|               |              |             |       |          |
|             |            |           |       |       |
+---------------+--------------+-------------+-------+----------+------------
+-------------+------------+-----------+-------+-------+
|   ten1epg1    |   tenant1    |  extension  |       | ten1po2  |   trunk    |
false    |     11     | sharedVrf | 31001 |       |
|               |              |             |       | ten1po1  |
|             |            |           |       |       |
|               |              |             |       |          |
|             |            |           |       |       |
+---------------+--------------+-------------+-------+----------+------------
+-------------+------------+-----------+-------+-------+
|   ten2epg3    |   tenant2    | l3-hand-off |       | sharedPO |   trunk    |
false    |     30     | ten2vrf1  | 21001 |       |
|               |              |             |       |          |
|             |            |           |       |       |
+---------------+--------------+-------------+-------+----------+------------
+-------------+------------+-----------+-------+-------+
|   ten2epg1    |   tenant2    |  extension  |       | ten2po2  |   trunk    |
false    |     21     | sharedVrf | 31001 |       |
|               |              |             |       | ten2po1  |
|             |            |           |       |       |
|               |              |             |       |          |
|             |            |           |       |       |
+---------------+--------------+-------------+-------+----------+------------
+-------------+------------+-----------+-------+-------+
|   ten2epg2    |   tenant2    |  extension  |       | ten2po1  |   trunk    |
false    |     23     | ten2vrf1  | 21001 |       |
|               |              |             |       | ten2po2  |
```

```
|             |            |           |       |       |
+---------------+-------------+-------------+-------+----------+------------
+-------------+------------+-----------+-------+-------+
efa tenant service bgp peer show --detail
==========================================
Name             : sharedTenbgppeer1
Tenant           : sharedTenant
State            : bs-state-created
Description      :

Static Peer
-----------
        Device IP        : 10.20.246.15
        VRF              : sharedVrf
        AFI              : ipv4
        SAFI             : unicast
        Remote IP        : 10.0.31.3
        Remote ASN       : 50000
        Next Hop Self    : false
        Update Source IP :
        BFD Enabled      : false
        BFD Interval     : 0
        BFD Rx           : 0
        BFD Multiplier   : 0
        MD5 Password     :
        Dev State        : provisioned
        App State        : cfg-in-sync

        Device IP        : 10.20.246.16
        VRF              : sharedVrf
        AFI              : ipv4
        SAFI             : unicast
        Remote IP        : 10.0.31.3
        Remote ASN       : 50000
        Next Hop Self    : false
        Update Source IP :
        BFD Enabled      : false
        BFD Interval     : 0
        BFD Rx           : 0
        BFD Multiplier   : 0
        MD5 Password     :
        Dev State        : provisioned
        App State        : cfg-in-sync

Dynamic Peer
-----------
        0 Records
==========================================
```

Sharing Multiple VRFs with the same RT (route-target)

1. If you are running EFA EFA 2.5.0 or below, run the following command:

```
efa tenant vrf create --tenant "ten1" --name "ten1vrf1" --routing-type "distributed" --
rt-type import --rt 100:100 --rt-type export --rt 100:100 --rt-type import --rt
200:200 --rt-type export --rt 200:200 --rt-type import --rt 300:300 --rt-type export --
rt 400:400 --max-path 8 --redistribute connected

efa tenant vrf create --tenant "ten2" --name "ten2vrf1" --routing-type "distributed" --
rt-type import --rt 100:100 --rt-type export --rt 100:100 --rt-type import --rt
200:200 --rt-type export --rt 200:200 --rt-type import --rt 300:300 --rt-type export --
rt 400:400 --max-path 8 --redistribute connected
VRF Creation Failed:
Route Target must be unique. Conflicting value: 'import 100:100'
```

2. If you are running EFA EFA 2.5.0 or above, run the following command:

```
efa tenant vrf create --tenant "ten1" --name "ten1vrf1" --routing-type "distributed" --
rt-type import --rt 100:100 --rt-type export --rt 100:100 --rt-type import --rt
200:200 --rt-type export --rt 200:200 --rt-type import --rt 300:300 --rt-type export --
rt 400:400 --max-path 8 --redistribute connected

efa tenant vrf create --tenant "ten2" --name "ten2vrf1" --routing-type "distributed" --
rt-type import --rt 100:100 --rt-type export --rt 100:100 --rt-type import --rt
200:200 --rt-type export --rt 200:200 --rt-type import --rt 300:300 --rt-type export --
rt 400:400 --max-path 8 --redistribute connected
```

If you are running EFA 2.5.0 or above, verify the switch configuration on SLX.

```
L1# show running-config vrf
bvrf ten1vrf1
rd 172.31.254.69:1
evpn irb ve 8192
address-family ipv4 unicast
  route-target export 100:100 evpn
  route-target export 200:200 evpn
  route-target export 400:400 evpn
  route-target import 100:100 evpn
  route-target import 200:200 evpn
  route-target import 300:300 evpn
!
address-family ipv6 unicast
  route-target export 100:100 evpn
  route-target export 200:200 evpn
  route-target export 400:400 evpn
  route-target import 100:100 evpn
  route-target import 200:200 evpn
  route-target import 300:300 evpn
!
!
vrf ten2vrf1
rd 172.31.254.69:2
evpn irb ve 8190
address-family ipv4 unicast
  route-target export 100:100 evpn
  route-target export 200:200 evpn
  route-target export 400:400 evpn
  route-target import 100:100 evpn
  route-target import 200:200 evpn
  route-target import 300:300 evpn
!
address-family ipv6 unicast
  route-target export 100:100 evpn
  route-target export 200:200 evpn
  route-target export 400:400 evpn
  route-target import 100:100 evpn
  route-target import 200:200 evpn
  route-target import 300:300 evpn
!
!
```

```
L2# show running-config vrf
bvrf ten1vrf1
rd 172.31.254.70:1
evpn irb ve 8192
address-family ipv4 unicast
  route-target export 100:100 evpn
  route-target export 200:200 evpn
  route-target export 400:400 evpn
  route-target import 100:100 evpn
  route-target import 200:200 evpn
  route-target import 300:300 evpn
!
address-family ipv6 unicast
  route-target export 100:100 evpn
  route-target export 200:200 evpn
  route-target export 400:400 evpn
  route-target import 100:100 evpn
  route-target import 200:200 evpn
  route-target import 300:300 evpn
!
!
vrf ten2vrf1
rd 172.31.254.70:2
evpn irb ve 8190
address-family ipv4 unicast
  route-target export 100:100 evpn
  route-target export 200:200 evpn
  route-target export 400:400 evpn
  route-target import 100:100 evpn
  route-target import 200:200 evpn
  route-target import 300:300 evpn
!
address-family ipv6 unicast
  route-target export 100:100 evpn
  route-target export 200:200 evpn
  route-target export 400:400 evpn
  route-target import 100:100 evpn
  route-target import 200:200 evpn
  route-target import 300:300 evpn
!
!
```

Distributed and Centralized Routing

L3 routers can be configured in centralized mode or distributed mode.

In centralized mode, routing is configured only on the border leaf pairs. In contrast, for distributed mode, routing is configured on the corresponding leaf nodes where the endpoints reside.

During router creation, You can provide centralized mode or distributed mode as input:

```
openstack router create R1  --distributed
openstack router create R2  --centralized
```

The default option is centralized, so the following command creates centralized routing:

```
openstack router create R2
```

The L3 service plugin passes this information to EFA. Then, the EFA Tenant Service configures VRF or routing on border leafs or on leaf nodes, based on the configuration mode.

Note

- The OpenStack integration works with only one pair of border leaf devices. The centralized routing instance is created on this pair. Ensure that you add only one border leaf pair during fabric creation.
- Only centralized router is supported at this time. The creation of a distributed router is disabled.

Prepare Clos Fabric for Centralized Routing

You can use the following command to create a Clos fabric containing border-leaf devices, which can be used for centralized routing.

Tip

If any devices in a fabric are in "admin-down" state, use of the following commands in that same fabric will not add or delete devices in the fabric: **`efa fabric device add-bulk`** and **`efa fabric device remove`**.

```
efa fabric create --name <fabric-name> --type clos

efa fabric device add-bulk --name <fabric-name>
           --border-leaf <list-of-border-leaf-ip> --leaf <list-of-leaf-ip>
           --spine <list-of-spine-ip>

efa fabric configure --name <fabric-name>
```

Prepare Small Data Center Fabric for Centralized Routing

You can use the following command to create a small data center (non-Clos) fabric containing border-leaf devices, which can be used for centralized routing.

Tip

If any devices in a fabric are in "admin-down" state, use of the following commands in that same fabric will not add or delete devices in the fabric: **`efa fabric device add-bulk`** and **`efa fabric device remove`**.

```
efa fabric create --name <fabric-name> --type non-clos

efa fabric device add-bulk --name <fabric-name>
           --rack <leaf-rack-name> --ip <leaf-ip-pair>
           --border-leaf-rack <bl-rack-name> --border-leaf-ip <bl-ip-pair>

efa fabric configure --name <fabric-name>
```

Enable Centralized Routing on Tenant VRF

You need to explicitly enable centralized routing at the tenant VRF level to override the default distributed routing behavior.

Therefore, a given tenant can have multiple VRFs with some VRFs operating in distributed routing mode and some VRFs operating in centralized routing mode.

To enable centralized routing on Tenant VRF, run the following command during VRF Create:

```
efa tenant vrf create --name <vrf-name> --tenant <tenant-name>
                      --routing-type {centralized | distributed}
                      --centralized-router <list-of-border-leaf-routers>
```

Example

```
efa tenant vrf create --name VRF1 --tenant tenant1
                  --routing-type centralized --centralized-router BL1-IP,BL2-IP
```

Configure physical routers for centralized routing on Tenant VRF

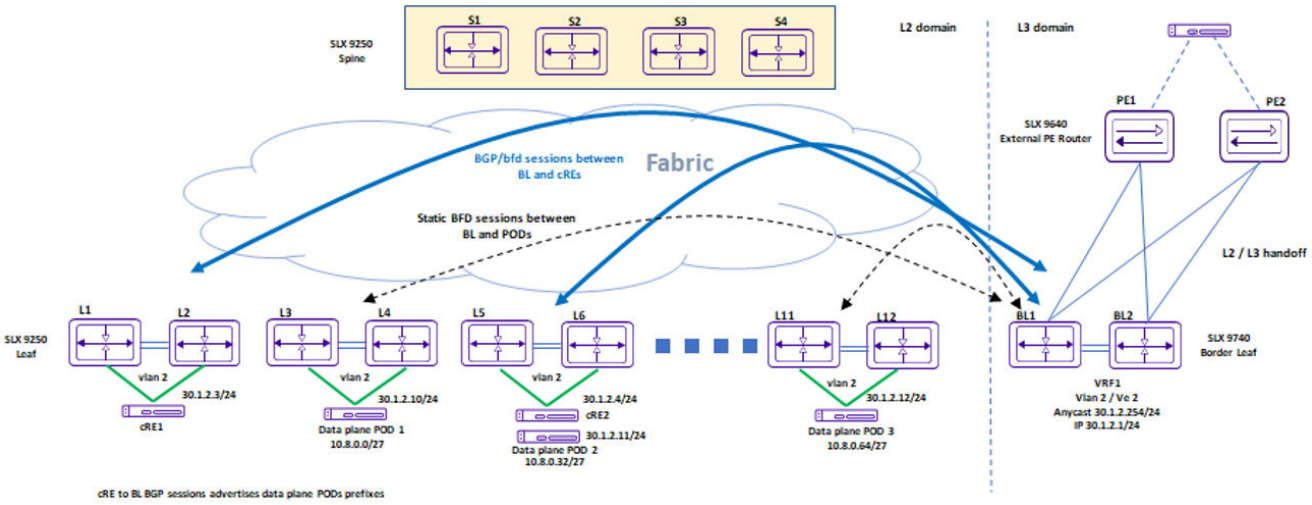
Provide a list of border-leaf IPs on which the centralized router (VRF) needs to be instantiated. If a given fabric has only one BL pair and you have not provided any BL pair as centralized router, then the only available BL pair is used as centralized router by default.

Provide only one BL pair on which the centailized router (VRF) needs to be instantiated.

VRF instantiation happens on the border-leaf devices during the EPG (endpoint group) create or update operations.

You can't provide leaf, spine, or super-spine IPs as the target device for centralized routing.

VRF (with centralized routing enabled) and its dependent L3 configuration (anycast-ip, local-ip, VRF static route, VRF static route bfd, router bgp static or dynamic peer, and router bgp peer-group) are instantiated only on the border-leaf(s) on which the parent VRF exists.



When the centralized routing is enabled for a given tenant VRF:

1. Define the target border-leaf device on which the VRF needs to be instantiated.

   The VRF instantiation happens only on those border-leaf devices and not on any other leaf or border-leaf devices.
2. You do not need to provide the target border-leaf device on which the anycast-ip needs to be configured.

The anycast-ip configuration happens *automatically* on the border-leaf devices on which the VRF is instantiated.

3. Provide the border-leaf IP (on which the VRF is instantiated) for the local-ip configuration.
4. Provide the border-leaf ip (on which the VRF is instantiated) for the VRF SR (Static Route) and VRF SR-BFD (Static Route – BFD) configuration.
5. Provide the border-leaf ip (on which the VRF is instantiated) for the BGP static and dynamic peer configuration.
6. Provide the border-leaf ip (on which the VRF is instantiated) for the BGP peer-group configuration.

1. To configure physical routers for centralized routing on Tenant VRF, run the following command during VRF Create:

```
efa tenant vrf create --name <vrf-name> --tenant <tenant-name>
                    --routing-type {centralized | distributed}
                       --centralized-router <list-of-border-leaf-routers>
```

Example

```
efa tenant vrf create --name VRF1 --tenant tenant1
                    --routing-type centralized --centralized-router BL1-IP,BL2-IP
```

2. Carve out of VRFs on the border-leaf pairs.

Instantiate VRFs on the border-leaf devices based on the L3 scale requirements.

Suppose the fabric has 100 VRFs with 4K anycast-ip, then you can instantiate all 100 VRFs on a single border-leaf pair. If the L3 scale requirements are higher than the scale supported by a single border-leaf pair, then add additional border-leaf pair.

Configure anycast-ip on Tenant Endpoint Group

Anycast IP gets automatically configured on the border-leaf (BL1 and BL2) on which the VRF is instantiated.

```
efa tenant epg create --name ten1epg1 --tenant tenant1
               --port L1-IP[0/11],L2-IP[0/11]
               --switchport-mode trunk -ctag-range 11 --vrf VRF1 --anycast-ip
11:10.10.11.1/24
```

Configure local-ip on Tenant Endpoint Group

To configure local IP on tenant endpoint group (EPG), run the following commands:

```
efa tenant epg create --name ten1epg1 --tenant tenant1 --vrf VRF1 --switchport-mode trunk
--ctag-range 11
                    --anycast-ip 11:10.10.11.1/24 --port L1-IP[0/1],L2-IP[0/1]
                  --local-ip 11,BL1-IP:11.22.33.41/24 --local-ip 11,BL2-IP:11.22.34.41/24
```

Configure static route on tenant VRF

Provide the border-leaf IP (on which the VRF is instantiated) for the VRF SR (Static Route) and VRF SR-BFD (Static Route – BFD) configuration.

1. Create static route on tenant VRF

```
efa tenant vrf create --name <vrf-name> --tenant <tenant-name>
            --ipv6-static-route-next-hop <border-leaf-ip, destination, next-hop,
distance>
            --ipv4-static-route-next-hop <border-leaf-ip, destination, next-hop,
distance>
```

2. Update static route on tenant VRF

```
efa tenant vrf update –name <vrf-name> --tenant <tenant-name>
            --operation <static-route-add|static-route-delete>
            --ipv6-static-route-next-hop <border-leaf-ip, destination, next-hop,
distance>
            --ipv4-static-route-next-hop <border-leaf-ip, destination, next-hop,
distance>
```

```
efa tenant vrf create --name VRF1 --tenant tenant1
            --ipv6-static-route-next-hop BL1-IP,2000::/64,1001::2
            --ipv6-static-route-next-hop BL1-IP,2000::/64,1002::2
            --ipv6-static-route-next-hop BL2-IP,2001::/64,1001::2,4
            --ipv6-static-route-next-hop BL2-IP,2001::/64,1002::2
            --ipv4-static-route-next-hop BL1-IP,22.0.0.0/24,13.0.0.1,2
            --ipv4-static-route-next-hop BL1-IP,22.0.0.0/24,13.0.0.2
            --ipv4-static-route-next-hop BL2-IP,23.0.0.0/24,13.0.0.1
            --ipv4-static-route-next-hop BL2-IP,23.0.0.0/24,13.0.0.2
```

Configure static route BFD on Tenant VRF

Provide the border-leaf IP (on which the VRF is instantiated) for the VRF SR-BFD (Static Route – BFD) configuration.

1. Configure static route BFD on Tenant VRF during VRF Create.

```
efa tenant vrf create --name <vrf-name> --tenant <tenant-name>
              --ipv6-static-route-bfd <border-leaf-ip, destination-ip, source-ip, bfd-
min-tx, bfd-min-rx, bfd-multiplier>
              --ipv4-static-route-bfd <border-leaf-ip, destination-ip, source-ip, bfd-
min-tx, bfd-min-rx, bfd-multiplier>
```

2. Configure static route BFD on Tenant VRF during VRF Update.

```
efa tenant vrf update –name <vrf-name> --tenant <tenant-name>
              --operation <static-route-bfd-add|static-route-bfd-delete>
              --ipv6-static-route-bfd <border-leaf-ip, destination-ip, source-ip, bfd-
min-tx, bfd-min-rx, bfd-multiplier>
              --ipv4static-route-bfd  <border-leaf-ip, destination-ip, source-ip, bfd-
min-tx, bfd-min-rx, bfd-multiplier>
```

```
efa tenant vrf create --name VRF1 --tenant tenant1
              --ipv6-static-route-bfd BL1-IP,1001::2,1001::1,100,200,5
              --ipv6-static-route-bfd BL2-IP,1011::2,1011::1,100,200,5
              --ipv6-static-route-bfd BL1-IP,1002::2, 1002::1
              --ipv6-static-route-bfd BL2-IP,1012::2, 1012::1
              --ipv4-static-route-bfd BL1-IP,13.0.0.1,13.0.0.9,200,300,6
              --ipv4-static-route-bfd BL2-IP,13.0.1.1,13.0.1.9,200,300,6
              --ipv4-static-route-bfd BL1-IP,13.0.0.2,13.0.0.10
              --ipv4-static-route-bfd BL2-IP,13.0.1.2,13.0.1.10
```

Configure peer-group on Tenant BGP

1. To configure BGP peer-group, run the following commands:

```
efa tenant service bgp peer-group create --name <peer-group-name> --tenant <tenant-
name>
      --description <description>
                --pg-name <border-leaf-ip:pg-name> --pg-asn <border-leaf-ip:pg-
name,remote-asn>
                --pg-bfd <border-leaf-ip:pg-name,bfd-enable(true/false),interval,min-
rx,multiplier>
                --pg-next-hop-self <border-leaf-ip:pg-name,next-hop-self(true/false)>
                --pg-update-source-ip <border-leaf-ip:pg-name,update-source-ip>
```

   Example

```
efa tenant service bgp peer-group create -name ten1BgpPG1 --tenant tenant1
                --pg-name BL1-IP:pg1 --pg-asn BL1-IP:pg1,6000
                --pg-bfd BL1-IP:pg1,true,100,200,5
                --pg-next-hop-self BL1-IP:pg1,true
                --pg-update-source-ip BL1-IP:pg1,10.20.30.40
```

2. To update BGP peer-group, run the following commands:

```
efa tenant service bgp peer-group update --name <peer-group-name> --tenant <tenant-
name>
                --operation <peer-group-add|peer-group-delete|peer-group-desc-update>
--description <description>
                --pg-name <border-leaf-ip:pg-name> --pg-asn <border-leaf-ip:pg-
name,remote-asn>
                --pg-bfd <border-leaf-ip:pg-name,bfd-enable(true/false),interval,min-
rx,multiplier>
                --pg-next-hop-self <border-leaf-ip:pg-name,next-hop-self(true/false)>
                --pg-update-source-ip <border-leaf-ip:pg-name,update-source-ip>
```

   Example

```
efa tenant service bgp peer-group update --name ten1BgpPG1 --tenant tenant1
                --operation peer-group-add
                --pg-name BL1-IP:pg2 -pg-asn BL1-IP:pg2,7000
                --pg-bfd BL1-IP:pg2,true,200,300,6
                --pg-next-hop-self BL1-IP:pg2,true
                --pg-update-source-ip BL1-IP:pg2,10.20.30.41
```

Configure static peer on Tenant BGP

1. To configure BGP static peer, run the following commands:

```
efa tenant service bgp peer create --name <peer-name> --tenant <tenant-
name>
            --ipv4-uc-nbr <border-leaf-ip,vrf-name:ipv4-neighbor,remote-
as>
            --ipv4-uc-nbr-bfd <border-leaf-ip,vrf-name:ipv4-neighbor,bfd-enable(true/
false),bfd-interval,bfd-rx,bfd-mult>
            --ipv4-uc-nbr-update-source-ip <border-leaf-ip,vrf-name:ipv4-
neighbor,update-source-ip>
            --ipv4-uc-nbr-next-hop-self <border-leaf-ip,vrf-name:ipv4-neighbor,next-
hop-self(true/false)>
            --ipv6-uc-nbr <border-leaf-ip,vrf-name:ipv6-neighbor,remote-as>
            --ipv6-uc-nbr-bfd <border-leaf-ip,vrf-name:ipv6-neighbor,bfd-enable(t/
f),bfd-interval,bfd-rx,bfd-mult>
            --ipv6-uc-nbr-update-source-ip <border-leaf-ip,vrf-name:ipv6-
neighbor,update-source-ip>
```

```
            --ipv6-uc-nbr-next-hop-self <border-leaf-ip,vrf-name:ipv6-neighbor,next-
hop-self(true/false)>
```

Example

```
efa tenant service bgp peer create --name bgpservice1 --tenant
tenant1
            --ipv4-uc-nbr BL1-IP,VRF1:10.20.30.40,5000
            --ipv4-uc-nbr-bfd BL1-
IP,VRF1:10.20.30.40,true,100,200,5
            --ipv4-uc-nbr-update-source-ip BL1-IP,VRF1:10.20.30.40,11.22.20.33
            --ipv4-uc-nbr-next-hop-self BL1-IP,VRF1:10.20.30.40,true
```

2. To update BGP static peer, run the following commands:

```
efa tenant service bgp peer update --name <peer-name> --tenant <tenant-name>
            --operation peer-add
            --ipv4-uc-nbr <border-leaf-ip,vrf-name:ipv4-neighbor,remote-as>
            --ipv4-uc-nbr-bfd <border-leaf-ip,vrf-name:ipv4-neighbor,bfd-enable(t/
f),bfd-interval,bfd-rx,bfd-mult>
            --ipv4-uc-nbr-update-source-ip <border-leaf-ip,vrf-name:ipv4-
neighbor,update-source-ip>
            --ipv4-uc-nbr-next-hop-self <border-leaf-ip,vrf-name:ipv4-neighbor,next-
hop-self(true/false)>
            --ipv6-uc-nbr <border-leaf-ip,vrf-name:ipv6-neighbor,remote-as>
            --ipv6-uc-nbr-bfd <border-leaf-ip,vrf-name:ipv6-neighbor,bfd-enable(t/
f),bfd-interval,bfd-rx,bfd-mult>
            --ipv6-uc-nbr-update-source-ip <border-leaf-ip,vrf-name:ipv6-
neighbor,update-source-ip>
            --ipv6-uc-nbr-next-hop-self <border-leaf-ip,vrf-name:ipv6-neighbor,next-
hop-self(true/false)>
```

Example

```
efa tenant service bgp peer update --name bgpservice1 --tenant tenant1 --operation
peer-add
            --ipv6-uc-nbr BL1-IP,VRF1:10::40,5000
            --ipv6-uc-nbr-bfd BL1-IP,VRF1:10::40,true,100,200,5
            --ipv6-uc-nbr-update-source-ip BL1-IP,VRF1:10::40,11::22
            --ipv6-uc-nbr-next-hop-self BL1-IP,VRF1:10::40,true
```

Configure Dynamic peer on Tenant BGP

1. To configure BGP dynamic peer, run the following commands:

```
efa tenant service bgp peer create --name <peer-name> --tenant <tenant-name>
                --ipv4-uc-dyn-nbr <border-leaf-ip,vrf-name:listen-range,peer-group-
name,listen-limit>
                --ipv6-uc-dyn-nbr <border-leaf-ip,vrf-name:listen-range,peer-group-
name,listen-limit>
```

Example

```
efa tenant service bgp peer create --name bgpservice1 --tenant tenant1
               --ipv4-uc-dyn-nbr BL1-IP,VRF1:11.22.33.44/30,pg1,10
```

2. To update BGP dynamic peer, run the following commands:

```
efa tenant service bgp peer update --name <peer-name> --tenant <tenant-name>
                --operation peer-add
                --ipv4-uc-dyn-nbr <border-leaf-ip,vrf-name:listen-range,peer-group-
name,listen-limit>
                --ipv6-uc-dyn-nbr <border-leaf-ip,vrf-name:listen-range,peer-group-
name,listen-limit>
```

Example

```
efa tenant service bgp peer create --name bgpservice1 --tenant
tenant1
                --operation peer-add –ipv4-uc-dyn-nbr BL1-IP,VRF1:11::22/127,pg1,20
```

Centralized Routing on Single Rack Small Data Center Leaf Pair (not Border Leaf Pair)

The following items are required before you configure centralized routing on Single Rack Small Data Center Leaf Pair.

- The device-role (leaf or border-leaf) are specified during the addition of the devices to the fabric, prior to the “fabric configure”.
- Border-Leaf pair can exist in a Clos or Small Data Center fabric irrespective of the VRFs instantiated in the fabric are distributed or centralized.
- Device role border-leaf implies the leaf pair used at the edge (border) of the fabric, and not restricted to the centralized routing.
- Tenant (PO, VRF, EPG, or BGP) provisioning happens on a configured fabric.
- Only the Border-Leaf devices can act as Centralized Routers.
- Default routing-type for a VRF is “distributed” and you need to explicitly provide the value “centralized” if needed.
- During the creation of the VRF as a CR (Centralized Router), EFA needs to instantiate the VRF on a pair of Border-Leaf devices.
- If the fabric (Clos or Small Data Center) has only one pair of Border-Leaf devices then the same pair is chosen as the designated CRs (Centralized Routers) for the VRF. Else, need to explicitly provide the Border-Leaf devices as the designated CRs (Centralized Routers) during the creation of the VRF.
- EFA is designed to expand or compress with the addition or deletion of racks (rack = MCT-pair) as per your requirement.
- EFA cannot determine “a given fabric is a single rack small data center fabric and can never be expanded beyond that”. Hence there is no specific automation for a single-rack use case.
- For CR on a single rack small data center fabric, EFA recommends that you configure the fabric with the device-role = border-leaf for both the MCT nodes.
- If you want to use Day 1 Centralized Routing provisioning on a “Day 0 Configured Single Rack Leaf Small Data Center Fabric”, then the same results in failure because CR can be instantiated only on the border-leaf pair of the small data center fabric.
- You cannot recreate the fabric with device-role = border-leaf.

Fabric Setting for a single-rack-deployment

1. A new fabric setting “single-rack-deployment” is introduced as follows:

```
efa fabric setting update --name <non-clos-fabric-name> --single-rack-deployment <Yes|
No>
```

2. The fabric setting is applicable only for the Small Data Center fabric.

3. Default value of single-rack-delployment = No.
4. After upgrade from EFA 2.4.x to EFA 2.5.0, all the EFA 2.4.x small data center fabrics have single-rack-deployment =No.
5. Single Rack Deployment
    a. When the “single-rack-deployment = Yes” and the fabric is configured,
        - You are not allowed to modify the single-rack-deployment from Yes to No.
        - The state is used as an indicator to EFA that “a given fabric is a single rack non clos fabric and will never be expanded beyond that”, so that EFA can have specific automation for the specific scenario of allowing the non-border-leaf rack act as CR (Centralized Router) for single rack small data center leaf pair deployments.
        - You cannot expand such a fabric. If you intend to expand such a fabric, then you must delete the fabric and recreate the same with “single-rack-deployment = No”.
    b. When the “single-rack-deployment = Yes” and the fabric is not configured,
        - You can modify the single-rack-deployment value from Yes to No.
6. Multi Rack Deployment
    a. When the “single-rack-deployment = No” and fabric is configured or not-configured,
        - You can modify the value of single-rack-deployment from No to Yes, provided the existing number of racks in the fabric is 1.
7. Fabric Device Add
    - Validations are done to ensure the number of racks in the given fabric adhere to the fabric settings.

Create a Tenant VRF for Single Rack Small Data Center Leaf Pair Deployment

Everything about the VRF remains as it is except for the Centralized Routing usecase on Single Rack Small Data Center Leaf Pair Deployment.

1. Create a VRF on Multi Rack Small Data Center or Multi Stage Clos Deployment.
2. Create a VRF on Single Rack Small Data Center Border-Leaf Pair Deployment.
3. Create a VRF on Single Rack Small Data Center Leaf Pair Deployment (single-rack-deployment = No).
4. Create a VRF on Single Rack Small Data Center Leaf Pair Deployment (single-rack-deployment = Yes).
    a. Create VRF on Distributed Router.
    b. Create VRF on Centralized Router.
        - During the creation of CR VRF, the single leaf rack will be considered as the designated CRs (Centralized Routers).
        - VRF will be instantiated as the CR (Centralized Router) VRF as input by the user and not DR (Distributed Router).

Configure a Single-Rack Leaf in Day 0 and Day 1 Provisioning

You can configure a single-rack leaf on Day 0 and then configure centralized and distributed routing on Day 1.

Tip
If any devices in a fabric are in "admin-down" state, use of the following commands in that same fabric will not add or delete devices in the fabric: **`efa fabric device add-bulk`** and **`efa fabric device remove`**.

1. Configure a single-rack leaf in a small data center fabric on Day 0.

```
efa fabric create -name <fabric-name> --type non-clos
efa fabric device add-bulk --name <fabric-name> --rack <rack-name> --ip <ip-pair>
--username <username> --password <password>
efa fabric configure --name <fabric-name>
```

2. Configure centralized routing on Day 1.

```
efa fabric setting update -name <fabric-name> --single-rack-deployment Yes
efa tenant vrf create -name <vrf-name> --tenant <tenant-name> --routing-type
centralized
```

3. Configure distributed routing on Day 1.

```
efa tenant vrf create -name <vrf-name> --tenant <tenant-name>
```

   There are no changes in the provisioning model.

Route-Distinguisher (RD) allocation independent of Route-Target (RT)

Provisioning in EFA pre-2.5.0

EFA auto allocates VRF RT in the format `xx:yy`.

EFA auto allocates VRF RD in the format `<router-id>:<lowest-yy-of-rt-in-a-given-vrf>`.

If users have not provided explicit RT values, then EFA uniquely allocates the RT values for each VRF. Since RD allocation is based on the RT value, EFA uniquely allocates the RD value for each VRF.

But if users provide RT values (in the format xx:yy and yy is common across multiple VRFs), then the RD which is derived based on the yy (<router-id>:<lowest-yy-of-rt>), cannot be unique across the VRFs and hence results in the failure indicating multiple VRFs cannot have common RD.

```
efa tenant vrf create --tenant "ten1" --name "ten1vrf1" --routing-type "distributed" --rt-
type export --rt 65010:1 --rt-type import --rt 65010:2 --max-path 8 --redistribute
connected

efa tenant vrf create --tenant "ten1" --name "ten1vrf2" --routing-type "distributed" --rt-
type export --rt 65010:2 --rt-type import --rt 65010:1 --max-path 8 --redistribute
connected

efa tenant epg create --tenant "ten1" --name "ten1epg1" --type extension --switchport-
mode trunk --single-homed-bfd-session-type auto --po ten1po1 --vrf ten1vrf1 --ctag-range
25 --l3-vni 32821 --anycast-ip 25:10.0.21.1/24 --ctag-description "25:Tenant L3 Extended
VLAN" --l2-vni 25:32770 --suppress-arp 25:true --suppress-nd 25:false

efa tenant epg create --tenant "ten1" --name "ten1epg2" --type extension --switchport-
mode trunk --single-homed-bfd-session-type auto --po ten1po1 --vrf ten1vrf2 --ctag-range
```

```
26 --l3-vni 32823 --anycast-ip 26:11.0.21.1/24 --ctag-description "26:Tenant L3 Extended
VLAN" --l2-vni 26:32771 --suppress-arp 26:true --suppress-nd 26:false

Device: 10.20.246.15
Network Policy Error: Vrf ten1vrf2 creation failed due to netconf rpc [error] '% Error:
This RD is already configured for another VRF',
Device: 10.20.246.16
Network Policy Error: Vrf ten1vrf2  creation failed due to netconf rpc [error] '% Error:
This RD is already configured for another VRF',
```

Provisioning in EFA 2.5.0 or above

EFA auto allocates VRF RT in the format xx:**yy**

EFA auto allocates VRF RD in the format <router-id>:<**unique-number-per-vrf-independent-of-rt**>

Since RD value has no relation to yy value of RT anymore, and since EFA auto allocates the unique-number-per-vrf which is appended to the router-id, the uniqueness of RD is always guaranteed.

```
efa tenant vrf create --tenant "ten1" --name "ten1vrf1" --routing-type "distributed" --rt-
type export --rt 65010:1 --rt-type import --rt 65010:2 --max-path 8 --redistribute
connected

efa tenant vrf create --tenant "ten1" --name "ten1vrf2" --routing-type "distributed" --rt-
type export --rt 65010:2 --rt-type import --rt 65010:1 --max-path 8 --redistribute
connected

efa tenant epg create --tenant "ten1" --name "ten1epg1" --type extension --switchport-
mode trunk --single-homed-bfd-session-type auto --po ten1po1 --vrf ten1vrf1 --ctag-range
25 --l3-vni 32821 --anycast-ip 25:10.0.21.1/24 --ctag-description "25:Tenant L3 Extended
VLAN" --l2-vni 25:32770 --suppress-arp 25:true --suppress-nd 25:false

efa tenant epg create --tenant "ten1" --name "ten1epg2" --type extension --switchport-
mode trunk --single-homed-bfd-session-type auto --po ten1po1 --vrf ten1vrf2 --ctag-range
26 --l3-vni 32823 --anycast-ip 26:11.0.21.1/24 --ctag-description "26:Tenant L3 Extended
VLAN" --l2-vni 26:32771 --suppress-arp 26:true --suppress-nd 26:false
```

Switch Config in EFA 2.5.0 or above

```
Rack1-Device1# show running-config
vrf
vrf ten1vrf1
rd 172.31.254.40:1
evpn irb ve 8192
address-family ipv4 unicast
  route-target export 65010:1 evpn
  route-target import 65010:2 evpn
!
address-family ipv6 unicast
  route-target export 65010:1 evpn
  route-target import 65010:2 evpn
!
!
vrf ten1vrf2
rd 172.31.254.40:2
evpn irb ve 8190
address-family ipv4 unicast
  route-target export 65010:2 evpn
  route-target import 65010:1 evpn
!
address-family ipv6 unicast
  route-target export 65010:2 evpn
  route-target import 65010:1 evpn
!
```

```
Rack1-Device2# show running-config
vrf
vrf ten1vrf1
rd 172.31.254.209:1
evpn irb ve 8192
address-family ipv4 unicast
  route-target export 65010:1 evpn
  route-target import 65010:2 evpn
!
address-family ipv6 unicast
  route-target export 65010:1 evpn
  route-target import 65010:2 evpn
!
!
vrf ten1vrf2
rd 172.31.254.209:2
evpn irb ve 8190
address-family ipv4 unicast
  route-target export 65010:2 evpn
  route-target import 65010:1 evpn
!
address-family ipv6 unicast
  route-target export 65010:2 evpn
  route-target import 65010:1 evpn
!
```

BFD timers for router-bgp BFD and static-route BFD sessions

- On the SLX, you can provide the Bidirectional Forwarding Detection (BFD) timer configurations per static-route, per BGP peer and per BGP peer-group.

  In SLX versions prior to 20.3.2:

  1. The timer values provided per static-route, per BGP peer, and per BGP peer-group are not effective for the single-hop static route BFD sessions and single-hop router BGP BFD sessions.
  2. For the timer values to be effective for the single-hop sessions, you must additionally configure the BFD timers also on the source interface (the interface acting as the BFD source).

  But this restriction is removed from 20.3.2 onwards.
- On the SLX, you can provide the BFD timer values directly at the “router bgp” level.

  1. In SLX versions prior to 20.3.2:

     The timer values provided directly at the “router bgp” level are effective only for the multi-hop BFD sessions of both default and non-default VRF BGP peers.
  2. In SLX versions 20.3.2 or above:

The timer values provided directly at the “router bgp” level are effective for both multi-hop and single-hop BFD sessions of both default and non-default VRF BGP peers.

Note

- If EFA gets upgraded first to v2.5.0 and later SLX-OS gets upgraded to v20.3.2 or above, drift and reconcile operation automatically gets triggered and reconciles the expected BFD configuration.

  Prior to upgrade of SLX-OS, ensure that you enable the maintenance mode on reboot using the **`efa inventory device setting update --maint-mode-enable-on-reboot Yes –ip <IP address of SLX>`** command. If you have upgraded SLX-OS without enabling the maintenance mode, trigger a manual drift and reconcile (DRC) using the command **`efa inventory drift-reconcile execute --ip <IP address of SLX>" –reconcile`**.
- If SLX-OS is already upgraded to v20.3.2 or above, and then EFA is upgraded to v2.5.0, trigger a manual drift and reconcile using the command **`efa inventory drift-reconcile execute --ip <IP address of SLX>" –reconcile`**.

  If you have not triggered the manual DRC, the **`efa fabric show --name <fabric name>`** command shows devices in `cfg-refreshed` state, and the intended BFD configuration does not get configured on the devices.
- For the static-route and router-bgp BFD enhancement to be effective, you must upgrade SLX to 20.3.2 or later.

Administered Partial Success

For a two-leaf MCT pair, you can enable configurations to succeed when only one node of the pair is reachable.

Overview

By default, when a REST operation succeeds on one device but fails on another, configuration changes are rolled back for both devices. For more information, see Rollback Scenarios for Data Consistency on page 50.

However, for a two-leaf MCT pair, you can administratively change the process to permit configuration to succeed even when one device is down. This process, called an administered partial success, is as follows.

- You use the **`efa inventory admin-state`** command to change the state of the unreachable device to "admin down." The device then goes into maintenance mode. For more information about changing a device state, see Administratively Manage a Device State on page 194.
- EFA filters out configurations destined for MCT pair as follows.
  - Create-related and delete-related configurations destined for the "admin up" device succeed.
  - Create-related configurations are not attempted for the "admin down" device, but the configurations are considered a success. These configurations are marked as pending, to be pushed to the device when it comes back up.

- Delete-related configurations (de-configurations) are not attempted for the "admin down" device and the operation fails with an error in the REST response. You can retry these de-configurations after the device transitions to “admin up” state.

  EFA does not want to leave stale configurations on the devices because if stale configurations are left on the devices, then bringing the devices (with stale configurations) back into EFA are erroneous considering the full brownfield support is missing in EFA.

- When the device is again reachable, you change the state of the device to "admin-up."
- EFA pushes the pending configurations to the device, and the drift and reconcile process ensures that the configurations in EFA and the device are synchronized. For more information, see Drift and Reconcile on page 39.
- The device comes out of maintenance mode.

Tips and considerations

- You can use Switch Health Management to verify the reachability of a device. Use the **`--health-check-interval`** and **`--health-check-heartbeat-miss-threshold`** settings of the **`efa inventory device setting update`** command. For more information, see Monitor Device Health on page 261.
- You can retry the same CLI or REST operation after the “admin down” devices transition to “admin up” state so that the deconfiguration is attempted on all the devices. You can use the “force” option available in the REST API to forcefully delete the entities from EFA even in case of partial success topology.
- You can use the **`efa tenant debug device drift`** command to determine any drift between the intended EFA configuration and the device configuration. These commands also identify the app state and the dev state: **`efa tenant epg show`** and **`efa tenant po show`**.
- EFA blocks the tenant reconciliation API, and rest of the tenant APIs support partial success behavior.
- If a high-availability failover or restart occurs while a device is in "admin down" or "admin up" state, you must reapply the state.
- If an operation such as drift and reconcile or a firmware download is in progress when you submit the command to change the state, the command is blocked until the operation is complete.
- This feature is supported only for devices in an MCT pair. Standalone devices are not supported.
- You can change the status of only one device in an MCT pair to "admin down" to benefit from administered partial success.
  - When both devices are in "admin down" state, the topology is considered a complete failure. Configuration attempts on these devices are rejected and error messages are returned in the REST responses. Administered partial success is not applicable.
  - When both devices are in "admin up" state, the topology is considered a complete success. Configuration attempts on these devices are accepted. Administered partial success is not applicable.

Behavior changes during "admin down" state

After a device state changes to "admin down," the following behavior changes occur.

- Switch Health Management does not trigger the drift and reconcile process.

- A device going into maintenance mode does not trigger the drift and reconcile process.
- The following commands are blocked from affecting the device.

Table 17: Blocked commands

Command type and name
Inventory commands
efa inventory device compare --ip
efa inventory drift-reconcile --ip
efa inventory device setting update --ip
efa inventory rma --ip
efa inventory config-backup execute --ip
efa inventory config-replay execute --ip
efa inventory device update --fabric
efa inventory device firmware-download prepare add --ip
efa inventory device update --ip
efa inventory device interface set-speed --ip
efa inventory device interface set-breakout --ip
efa inventory device interface unset-breakout --ip
efa inventory device interface set-mtu --ip
efa inventory device interface set-admin-state --ip
efa inventory device running-config persist --ip
Fabric commands
efa fabric configure --name
efa fabric device remove --ip <> --name <> Allowed with the --no-device-cleanup option.
efa fabric show-config --name
efa fabric topology show underlay --name
efa fabric topology show overlay --name
efa fabric topology show physical --name

Behavior changes during "admin up" state

After a device is returned to "admin up" state and after the drift and reconcile process is complete (which the state change triggers), Switch Health Management and drift and reconcile resume normal behavior. Also, the blocked commands are unblocked.

Administratively Manage a Device State

You can administratively manage the state of an SLX device using the EFA command line.

You can change a state to up or down, delete a state from the history, and view the state history and the current state. For details about the command and its parameters, see the *Extreme Fabric Automation Command Reference, 2.7.0*.

1. To change a device to the up state, run the following command.

```
$ efa inventory admin-state up --ip <device IP>
AdminStateUp [success]
Admin State Up execution UUID: 8d9fa0cf-dc76-42cc-ac7a-57902a47c1b2
```

   This example changes the state for the specified IP address and generates a UUID, which you can use in the **`efa inventory admin-state detail`** version of the command.

2. To change a device to the down state, run the following command.

```
$ efa inventory admin-state down --ip <device IP>
AdminStateDown [success]
Admin State Down execution UUID: 28eb0845-7a7a-4851-b453-b3020c6900f2
```

   This example changes the state for the specified IP address and generates a UUID, which you can use in the **`efa inventory admin-state detail`** version of the command.

3. To review the details of a state change, run the following command.

```
$ efa inventory admin-state detail --uuid 28eb0845-7a7a-4851-b453-b3020c6900f2
```

4. To view the history of the admin changes for a specified device, run the following command.

```
# efa inventory admin-state history --ip <device IP>
```

5. To display the admin state and the health check state of a device, run the following command.

```
$ efa inventory admin-state show --ip <device IP>
```

6. To delete the instance of the admin state change for a device, run the following command.

```
$ efa inventory admin-state delete --key <device IP or UUID>
```

APS Behavior of Tenant Configuration

Existing behavior in EFA 2.3.0

Configuration or Deconfiguration is never attempted on admin down switching devices.

Target Devices

Devices on which the configuration is intended to be pushed.

Complete Failure Topology

- Topology having atleast one single-homed device in admin down state or atleast one dual-homed device pair with both the devices in admin down state is a “complete failure topology”.

- Any Tenant CLI or REST API of create or delete nature attempted on the target devices having “complete failure topology” is rejected with an appropriate error and result in a complete failure. EFA does not have any configuration recipe prepared for this REST API or CLI as the entire request is rejected. For example, an EPG (endpoint group) create attempted on a single-homed device which is admin down state.

Complete Success Topology

- Topology with all the single-homed devices and all the dual-homed device pair in admin up state is a “complete success topology”.
- Any Tenant CLI or REST API of create or delete nature attempted on the target devices having “complete success topology” results in the configuration recipe preparation for all the target devices and the configuration is attempted on all the target devices as all the target devices are in “admin up” state.

Partial Success Topology

- Topology which is not “complete failure topology” and having at least one dual-homed device pair with one of the device in admin down state and the other device in admin up state is a “partial success topology”.
- Any Tenant CLI or REST API of create nature attempted on the target devices having “partial success topology” will result in the configuration being attempted onto the “admin up” devices and configuration not being attempted onto the “admin down” devices. Even though configuration is not attempted for “admin down” devices, the configuration is treated as success for the “admin down” devices.

  Configuration recipe is prepared and persisted in EFA for all the devices and the configuration is auto reconciled with the devices when the “admin down” devices transition to “admin up”.

  When the devices are in the “admin up” state, the EFA intended configuration synchronizes with the device configuration.
- Any Tenant CLI or REST API of delete nature attempted on the target devices having “partial success topology” results in deconfiguration attempted on the “admin up” devices, and deconfiguration not attempted on the “admin down” devices. The CLI or REST API operation fails with an appropriate error indicating that the deconfiguration not being attempted on the “admin down” devices.

  The reason being EFA does not want to leave stale configurations on the devices because if the stale configurations are left on the devices, then bringing the devices (having stale configurations) back into EFA is erroneous considering the full brownfield support is missing in EFA. You can retry the same CLI or REST API operation after the “admin down” devices transition to “admin up” state so that the deconfiguration is attempted on all the devices. You can always use “force” option available in REST API to forcefully delete the entities from EFA even in case of partial success topology.

- Drift between the EFA intended configuration and device config is shown in the **`efa tenant debug device drift`** CLI or REST API output and in the corresponding entity GET or SHOW output (for example, **`efa tenant epg show`** and **`efa tenant po show`** in the form of "app-state" and "dev-state".
- EFA blocks the tenant reconciliation API and rest of the tenant APIs support partial success behavior.

APS: Pre-provisioning Support by Modifying the Target Device List to Include the MCT Neighbor

Scenario	2.3.2 Target Device List	2.4.0 Target Device List	EFA 2.3.2 Resultant Topology	EFA 2.4.0 Resultant Topology
Single Homed PO Create with PO member on Rack1Device2	Rack1Device2	Rack1Device1 Rack1Device2	Complete Failure	Partial Success
Endpoint group (EPG) create with cluster edge port (CEP) member on Rack1Device2	Rack1Device2 Rack1Device1	Rack1Device1 Rack1Device2	Partial Success	Partial Success
BGP Peer Group create with the peer-group residing on Rack1Device2	Rack1Device2	Rack1Device1 Rack1Device2	Complete Failure	Partial Success
BGP Peer Create with the static or dynamic BGP peers residing on Rack1Device2	Rack1Device2	Rack1Device1 Rack1Device2	Complete Failure	Partial Success
VRF update with SR or SR-BFD residing on Rack1Device2	Rack1Device2	Rack1Device1 Rack1Device2	Complete Failure	Partial Success

```
efa tenant show
+------+-------------+-------------+------------+-----------+-----------+---------
+----------------------+
| Name | L2VNI-Range | L3VNI-Range | VLAN-Range | VRF-Count | Enable-BD |  Type
|        Ports         |
+------+-------------+-------------+------------+-----------+-----------+---------
+----------------------+
| ten1 |             |             | 11-20      | 10        | False     | private |
10.20.246.15[0/1-10] |
|      |             |             |            |           |           |         |
10.20.246.16[0/1-10] |
+------+-------------+-------------+------------+-----------+-----------+---------
+----------------------+
Tenant Details

efa inventory admin-state down --ip 10.20.246.15
AdminStateDown [success]
Admin State Down execution UUID: 6eaa1ebe-40fe-4628-8d5c-df11ffc4521e
execute the CLI to get details : efa inventory admin-state detail --uuid
6eaa1ebe-40fe-4628-8d5c-df11ffc4521e
```

```
efa inventory admin-state detail --uuid 6eaa1ebe-40fe-4628-8d5c-df11ffc4521e
+--------------------------------+----------------------------------------+
|              NAME              |                VALUE                   |
+--------------------------------+----------------------------------------+
| UUID                           | 6eaa1ebe-40fe-4628-8d5c-df11ffc4521e |
+--------------------------------+----------------------------------------+
| Device IP                      | 10.20.246.15                           |
+--------------------------------+----------------------------------------+
| Admin State Action             | down                                   |
+--------------------------------+----------------------------------------+
| Status                         | success                                |
+--------------------------------+----------------------------------------+
| Fabric Status                  | success                                |
+--------------------------------+----------------------------------------+
| Tenant Status                  | success                                |
+--------------------------------+----------------------------------------+
| Maintenance Mode Enable Status | success                                |
+--------------------------------+----------------------------------------+
| Start Time                     | 2021-02-06 21:18:53 -0800 PST          |
+--------------------------------+----------------------------------------+
| Last Modified                  | 2021-02-06 21:19:59 -0800 PST          |
+--------------------------------+----------------------------------------+
| Duration                       | 1m5.517263907s                         |
+--------------------------------+----------------------------------------+
```

Behavior in 2.3.2

```
efa tenant po create --name ten1po1 --tenant ten1 --port 10.20.246.15[0/1-2] --speed
10Gbps --negotiation active
    PortChannel creation failed:
            Error: Device 10.20.246.15 is administratively down

efa tenant po create --name ten1po1 --tenant ten1 --port 10.20.246.16[0/1-2] --speed
10Gbps --negotiation active
    PortChannel created successfully.

efa tenant epg create --name ten1epg1 --tenant ten1 --po ten1po1 --switchport-mode trunk
--ctag-range 11-12 --anycast-ip 11:10.0.11.1/24 --anycast-ip 12:10.0.12.1/24 --vrf
ten1vrf1 --l2-vni 11:11 --l2-vni 12:12 --l3-vni 8192
    EndpointGroup created successfully.

efa tenant service bgp peer create --name ten1bgppeer1 --tenant ten1 --ipv4-uc-nbr
10.20.246.15,ten1vrf1:10.0.0.0,65001
    BgpService creation Failed:
            Error: Devices [10.20.246.15] are administratively down

efa tenant service bgp peer-group create --name ten1bgppeergroup1 --tenant ten1 --pg-name
10.20.246.15:pg1 --pg-asn 10.20.246.15,pg1:65010
    BgpService creation Failed:
            Error: Devices [10.20.246.15] are administratively down
```

Behavior in 2.4.0

```
efa tenant po create --name ten1po1 --tenant ten1 --port 10.20.246.15[0/1-2] --speed
10Gbps --negotiation active

efa tenant epg create --name ten1epg1 --tenant ten1 --po ten1po1 --switchport-mode trunk
--ctag-range 11-12 --anycast-ip 11:10.0.11.1/24 --anycast-ip 12:10.0.12.1/24 --vrf
ten1vrf1

efa tenant service bgp peer create --name ten1bgppeer1 --tenant ten1 --ipv4-uc-nbr
10.20.246.15,ten1vrf1:10.0.0.0,65001

efa tenant service bgp peer-group create --name ten1bgppeergroup1 --tenant ten1 --pg-name
```

```
10.20.246.15:pg1 --pg-asn 10.20.246.15,pg1:65010

efa tenant po show --name ten1po1 --tenant ten1
+---------+--------+----+--------+-------------+----------+---------+--------------------
+------------+-----------------+-----------+
|  Name   | Tenant | ID | Speed  | Negotiation | Min Link |  Lacp   |        Ports
|   State    |    Dev State    | App State |
|         |        |    |        |             |  Count   | Timeout |
|           |                 |           |
+---------+--------+----+--------+-------------+----------+---------+--------------------
+------------+-----------------+-----------+
| ten1po1 |  ten1  | 1  | 10Gbps |   active    |    1     |  long   | 10.20.246.15[0/1-2]
| po-created | not-provisioned | cfg-ready |
+---------+--------+----+--------+-------------+----------+---------+--------------------
+------------+-----------------+-----------+

efa tenant vrf show --name ten1vrf1 --tenant ten1
(efa:root)root@node-2:~# efa tenant vrf show --name ten1vrf1 --tenant ten1
+----------+--------+-------------+--------------------+-----------------+----------
+-----------+-----------+------------+-----------------+-----------+
|   Name   | Tenant | Routing Type | Centralized Routers |   Redistribute   | Max Path |
Local Asn | Enable GR |   State    |    Dev State    | App State |
+----------+--------+-------------+--------------------+-----------------+----------
+-----------+-----------+------------+-----------------+-----------+
| ten1vrf1 |  ten1  | distributed  |                    | connected,static |    50
|   65002   |   false   | vrf-create | not-provisioned | cfg-ready |
+----------+--------+-------------+--------------------+-----------------+----------
+-----------+-----------+------------+-----------------+-----------+


efa tenant epg show --detail
================================================================================================
=============================================================================
Name          : ten1epg1
Tenant        : ten1
Description   :
Type          : extension
Ports         :
POs           :
              : unstable            : ten1po1
Port Property : switchport mode     : trunk
              : native-vlan-tagging : false
NW Policy     : ctag-range          : 11-12
              : vrf                 : ten1vrf1 [unstable]
              : l3-vni              : 8192
Network Property [Flags : * - Native Vlan]
+------+--------+--------------+--------------+---------+--------------------------------
+-------------------------+-------------+-----------------------+---------------------
+-----------------+-----------+
| Ctag | L2-Vni | Anycast-IPv4 | Anycast-IPv6 | BD-name | Local IP (Device-IP->Local-IP)
|    Ctag-Description     | Mtu-IPv6-ND | ManagedConfig-IPv6-ND | OtherConfig-IPv6-ND
|    Dev-state    | App-state |
+------+--------+--------------+--------------+---------+--------------------------------
+-------------------------+-------------+-----------------------+---------------------
+-----------------+-----------+
| 11   | 11     | 10.0.11.1/24 |              |         |
| Tenant L3 Extended VLAN |             | False                 | False               |
not-provisioned | cfg-ready |
+------+--------+--------------+--------------+---------+--------------------------------
+-------------------------+-------------+-----------------------+---------------------
+-----------------+-----------+
| 12   | 12     | 10.0.12.1/24 |              |         |
| Tenant L3 Extended VLAN |             | False                 | False               |
not-provisioned | cfg-ready |
```

```
+------+--------+-------------+-------------+---------+-------------------------------
+-------------------------+-------------+----------------------+--------------------
+-----------------+-----------+

efa tenant service bgp peer-group show
================================================================================================
=================================================================
Name        : ten1bgppeergroup1
Tenant      : ten1
State       : bgp-pg-state-created
Description :
+--------------+-----------------+------------+------------+--------------+--------
+----------------+---------------+-----------------+-----------------+-----------+
|  Device IP   | PEER-GROUP-NAME | REMOTE ASN | BFD Enabled | BFD Interval | BFD Rx | BFD
Multiplier | Next-Hop-Self | Update-Source-IP |    Dev-state    | App-state |
+--------------+-----------------+------------+-------------+--------------+--------
+----------------+---------------+-----------------+-----------------+-----------+
| 10.20.246.15 | pg1             | 65010      | false       | 0            | 0      |
0              | false         |                 | not-provisioned | cfg-ready |
+--------------+-----------------+------------+-------------+--------------+--------
+----------------+---------------+-----------------+-----------------+-----------+

efa tenant service bgp peer show
================================================================================================
================================================================================================
========
Name        : ten1bgppeer1
Tenant      : ten1
State       : bs-state-created
Description :
Static Peer:
+--------------+----------+------+---------+-----------+------------+-------------
+--------------+--------+----------------+---------------+------------------
+-----------------+-----------+
|  Device IP   |   VRF    | AFI  |  SAFI   | REMOTE IP | REMOTE ASN | BFD Enabled | BFD
Interval | BFD Rx | BFD Multiplier | Next Hop Self | Update Source IP |    Dev-state    |
App-state |
+--------------+----------+------+---------+-----------+------------+-------------
+--------------+--------+----------------+---------------+------------------
+------------------+-----------+
| 10.20.246.15 | ten1vrf1 | ipv4 | unicast | 10.0.0.0  | 65001      | false       |
0            | 0      | 0              | false         |                  | not-
provisioned | cfg-ready |
+--------------+----------+------+---------+-----------+------------+-------------
+--------------+--------+----------------+---------------+------------------
+------------------+-----------+
Dynamic Peer:
+-----------+-----+-----+------+--------------+------------+--------------+-----------
+-----------+
| Device IP | VRF | AFI | SAFI | Listen Range | Peer Group | Listen Limit | Dev-state |
App-state |
+-----------+-----+-----+------+--------------+------------+--------------+-----------
+-----------+
================================================================================================
================================================================================================
========

efa inventory admin-state up --ip 10.20.246.15
AdminStateUp [success]
Admin State Up started execution UUID: 0ae0db00-c0de-4d55-8410-6d67bca4ad65


efa inventory admin-state detail --uuid 0ae0db00-c0de-4d55-8410-6d67bca4ad65
+------------------------------+----------------------------------------+
```

```
|            NAME             |                VALUE                 |
+-----------------------------+--------------------------------------+
| UUID                        | 0ae0db00-c0de-4d55-8410-6d67bca4ad65 |
+-----------------------------+--------------------------------------+
| Device IP                   | 10.20.246.15                         |
+-----------------------------+--------------------------------------+
| Admin State Action          | up                                   |
+-----------------------------+--------------------------------------+
| Status                      | success                              |
+-----------------------------+--------------------------------------+
| Fabric Status               | success                              |
+-----------------------------+--------------------------------------+
| Tenant Status               | success                              |
+-----------------------------+--------------------------------------+
| Drift and Reconcile id      | a3c987ea-f709-4f7c-8ae4-bc5c9481cc55 |
+-----------------------------+--------------------------------------+
| Drift and Reconcile Status  | DR Completed                         |
+-----------------------------+--------------------------------------+
| Start Time                  | 2021-02-06 21:39:09 -0800 PST        |
+-----------------------------+--------------------------------------+
| Last Modified               | 2021-02-06 21:47:09 -0800 PST        |
+-----------------------------+--------------------------------------+
| Duration                    | 8m0.126957723s                       |
+-----------------------------+--------------------------------------+
--- Time Elapsed: 47.334754ms ---
(efa:root)root@node-2:~#

efa tenant vrf show --name ten1vrf1 --tenant ten1
+----------+--------+-------------+---------------------+------------------+----------
+-----------+-----------+--------------------+-------------+-------------+
|   Name   | Tenant | Routing Type | Centralized Routers |   Redistribute   | Max Path |
Local Asn | Enable GR |       State        |  Dev State  |  App State  |
+----------+--------+-------------+---------------------+------------------+----------
+-----------+-----------+--------------------+-------------+-------------+
| ten1vrf1 |  ten1  | distributed  |                     | connected,static |    50
|   65002   |   false   | vrf-device-created | provisioned | cfg-in-sync |
+----------+--------+-------------+---------------------+------------------+----------
+-----------+-----------+--------------------+-------------+-------------+

efa tenant po show
+---------+--------+----+-------------------------+--------+-------------+--------------
+----------------------+-------------+------------+-------------+--------------+
|  Name   | Tenant | ID |       Description       | Speed  | Negotiation | MinLinkCount
|        Ports        | LacpTimeout |   State    |  Dev-State  |   App-State  |
+---------+--------+----+-------------------------+--------+-------------+--------------
+----------------------+-------------+------------+-------------+--------------+
| ten1po1 | ten1   | 1  | EFA Port-channel ten1po1 | 10Gbps | active      | 1
| 10.20.246.15[0/1-2] | long        | po-created | provisioned |  cfg-in-sync |
+---------+--------+----+-------------------------+--------+-------------+--------------
+----------------------+-------------+------------+-------------+--------------+

efa tenant epg show
================================================================================================
=========================================================================
Name          : ten1epg1
Tenant        : ten1
Description   :
Type          : extension
Ports         :
POs           :
              : unstable            : ten1po1
Port Property : switchport mode     : trunk
              : native-vlan-tagging : false
NW Policy     : ctag-range          : 11-12
```

```
              : vrf                : ten1vrf1
              : l3-vni             : 8192
Network Property [Flags : * - Native Vlan]
+------+--------+--------------+--------------+---------+-------------------------------
+-------------------------+-------------+----------------------+--------------------
+-------------+--------------+
| Ctag | L2-Vni | Anycast-IPv4 | Anycast-IPv6 | BD-name | Local IP (Device-IP->Local-IP)
|    Ctag-Description     | Mtu-IPv6-ND | ManagedConfig-IPv6-ND | OtherConfig-IPv6-ND |
Dev-state  |   App-state  |
+------+--------+--------------+--------------+---------+-------------------------------
+-------------------------+-------------+----------------------+--------------------
+-------------+--------------+
| 11   | 11     | 10.0.11.1/24 |              |         |
| Tenant L3 Extended VLAN |             | False                | False               |
provisioned |  cfg-in-sync |
+------+--------+--------------+--------------+---------+-------------------------------
+-------------------------+-------------+----------------------+--------------------
+-------------+--------------+
| 12   | 12     | 10.0.12.1/24 |              |         |
| Tenant L3 Extended VLAN |             | False                | False               |
provisioned |  cfg-in-sync |
+------+--------+--------------+--------------+---------+-------------------------------
+-------------------------+-------------+----------------------+--------------------
+-------------+--------------+
For 'unstable' entities, run 'efa tenant po/vrf show' for details
================================================================================================
====================================================================


efa tenant service bgp peer-group show
================================================================================================
========================================================================
Name        : ten1bgppeergroup1
Tenant      : ten1
State       : bgp-pg-state-created
Description :
+--------------+----------------+------------+-------------+--------------+--------
+----------------+---------------+-----------------+-------------+-------------+
|  Device IP   | PEER-GROUP-NAME | REMOTE ASN | BFD Enabled | BFD Interval | BFD Rx | BFD
Multiplier | Next-Hop-Self | Update-Source-IP |  Dev-state  |  App-state  |
+--------------+----------------+------------+-------------+--------------+--------
+----------------+---------------+-----------------+-------------+-------------+
| 10.20.246.15 | pg1            | 65010      | false       | 0            | 0      |
0              | false         |                 | provisioned | cfg-in-sync |
+--------------+----------------+------------+-------------+--------------+--------
+----------------+---------------+-----------------+-------------+-------------+
================================================================================================
====================================================================

efa tenant service bgp peer show
================================================================================================
================================================================================================
========
Name        : ten1bgppeer1
Tenant      : ten1
State       : bs-state-created
Description :
Static Peer:
+--------------+----------+------+---------+-----------+------------+-------------
+--------------+--------+---------------+---------------+-----------------+-------------
+-------------+
|  Device IP   |   VRF    | AFI  |  SAFI   | REMOTE IP | REMOTE ASN | BFD Enabled | BFD
Interval | BFD Rx | BFD Multiplier | Next Hop Self | Update Source IP |  Dev-state  |
App-state  |
```

```
+--------------+----------+------+---------+-----------+------------+-------------
+--------------+--------+----------------+---------------+------------------+-------------
+-------------+
| 10.20.246.15 | ten1vrf1 | ipv4 | unicast | 10.0.0.0  | 65001      | false       |
0            | 0      | 0              | false         |                  | provisioned |
cfg-in-sync |
+--------------+----------+------+---------+-----------+------------+-------------
+--------------+--------+----------------+---------------+------------------+-------------
+-------------+
Dynamic Peer:
+-----------+-----+-----+------+-------------+------------+--------------+-----------
+-----------+
| Device IP | VRF | AFI | SAFI | Listen Range | Peer Group | Listen Limit | Dev-state |
App-state |
+-----------+-----+-----+------+-------------+------------+--------------+-----------
+-----------+
================================================================================================
================================================================================================
========
```

APS: Deletion Support for Pre-provisioned Configurations

Problem in EFA 2.3.2

Creation of EFA entities (PO and EPG) on an admin down device followed by the Deletion of the same EFA entities (PO and EPG) on the same admin down device used to fail even though the configuration was never pushed to the devices.

Solution in EFA 2.4.0

Succeed the deletion of the EFA entities (PO and EPG) if the resultant configuration to be deleted has never been pushed to the devices.

Pre-provisioned config

The pre-provisioned config is present in EFA DB and not present on the SLX.

```
efa inventory admin-state  show --ip 10.20.246.15
+----------------------+--------------+
|         NAME         |VALUE         |
+----------------------+--------------+
| Device IP            |10.20.246.15  |
+----------------------+--------------+
| Admin State          |down          |
+----------------------+--------------+
| Health Check Status  |Disable       |
+----------------------+--------------+
```

```
efa inventory admin-state  show --ip 10.20.246.16
+----------------------+--------------+
|         NAME         |VALUE         |
+----------------------+--------------+
| Device IP            |10.20.246.16  |
+----------------------+--------------+
| Admin State          |up            |
+----------------------+--------------+
| Health Check Status  |Disable       |
+----------------------+--------------+
```

```
efa tenant po create --name ten1po1 --tenant ten1 --port
10.20.246.15[0/1],10.20.246.16[0/1] --speed 10Gbps --negotiation active
efa tenant vrf create --name ten1vrf1 --tenant ten1
efa tenant epg create --name ten1epg1 --tenant ten1 --po ten1po1 --switchport-mode trunk
--ctag-range 11-12 --anycast-ip 11:10.0.11.1/24 --anycast-ip 12:10.0.12.1/24 --vrf
ten1vrf1
efa tenant service bgp peer create --name ten1bgppeer1 --tenant ten1 --ipv4-uc-nbr
10.20.246.15,ten1vrf1:10.0.0.0,65001 --ipv4-uc-nbr 10.20.246.16,ten1vrf1:10.1.0.0,65001
efa tenant service bgp peer-group create --name ten1bgppeergroup1 --tenant ten1 --pg-name
10.20.246.15:pg1 --pg-asn 10.20.246.15,pg1:65010  --pg-name 10.20.246.16:pg1 --pg-asn
10.20.246.16,pg1:65010

efa tenant po show
+---------+--------+----+-------------------------+--------+-------------+--------------
+-------------------+-------------+------------+-----------------+-----------+
|  Name   | Tenant | ID |       Description       | Speed  | Negotiation | MinLinkCount
|       Ports       | LacpTimeout |   State    |    Dev-State    | App-State |
+---------+--------+----+-------------------------+--------+-------------+--------------
+-------------------+-------------+------------+-----------------+-----------+
| ten1po1 | ten1   | 1  | EFA Port-channel ten1po1 | 10Gbps | active      | 1
| 10.20.246.15[0/1] | long        | po-created | not-provisioned | cfg-ready |
|         |        |    |                          |        |             |
| 10.20.246.16[0/1] |             |            |                 |           |
+---------+--------+----+-------------------------+--------+-------------+--------------
+-------------------+-------------+------------+-----------------+-----------+

efa tenant service bgp peer-group show
================================================================================================
=================================================================
Name        : ten1bgppeergroup1
Tenant      : ten1
State       : bgp-pg-state-created
Description :
+--------------+----------------+------------+-------------+--------------+--------
+---------------+---------------+-----------------+-----------------+-------------+
|  Device IP   | PEER-GROUP-NAME | REMOTE ASN | BFD Enabled | BFD Interval | BFD Rx | BFD
Multiplier | Next-Hop-Self | Update-Source-IP |    Dev-state    |  App-state  |
```

```
+--------------+-----------------+-----------+-------------+--------------+--------
+----------------+---------------+-----------------+-----------------+-------------+
| 10.20.246.16 | pg1             | 65010     | false       | 0            | 0      |
0              | false         |                 | provisioned     | cfg-in-sync |
+--------------+-----------------+-----------+-------------+--------------+--------
+----------------+---------------+-----------------+-----------------+-------------+
| 10.20.246.15 | pg1             | 65010     | false       | 0            | 0      |
0              | false         |                 | not-provisioned | cfg-ready   |
+--------------+-----------------+-----------+-------------+--------------+--------
+----------------+---------------+-----------------+-----------------+-------------+
================================================================================================
=====================================================================

efa tenant vrf show --name ten1vrf1 --tenant ten1
================================================================================================
=====================================================================
Name                    : ten1vrf1
Vrf State               : vrf-device-created
Vrf Device State        : not-provisioned
Vrf App State           : cfg-ready
Tenant Name             : ten1
Routing Type            : distributed
L3 VNI                  : 8191
IRB BD                  : 4095
IRB VE                  : 8191
BR BD                   :
BR VE                   :
BR VNI                  : 4096
RH max path             :
RH ecmp enable          :
Graceful restart enable :
Route Target            : import 101:101
                        : export 101:101
Static Route            : Switch-IP->Network,Nexthop-IP[Route-Distance], ...
                        :
Local Asn               :
Static Route BFD        : Switch-IP->[DestIP,SourceIP][Interval,Min-Rx,Multiplier], ...
                        :
Max Path                : 8
Redistribute            : connected
================================================================================================
=====================================================================

efa tenant epg show
================================================================================================
=====================================================================
Name          : ten1epg1
Tenant        : ten1
Description   :
Type          : extension
Ports         :
POs           :
              : unstable            : ten1po1
Port Property : switchport mode     : trunk
              : native-vlan-tagging : false
NW Policy     : ctag-range          : 11-12
              : vrf                 : ten1vrf1 [unstable]
              : l3-vni              : 8191
Network Property [Flags : * - Native Vlan]
+------+--------+-------------+-------------+---------+-------------------------------
+-------------------------+-------------+----------------------+---------------------
+-----------------+------------+
| Ctag | L2-Vni | Anycast-IPv4 | Anycast-IPv6 | BD-name | Local IP (Device-IP->Local-IP)
|    Ctag-Description     | Mtu-IPv6-ND | ManagedConfig-IPv6-ND | OtherConfig-IPv6-ND
```

```
|    Dev-state    | App-state |
+------+--------+--------------+--------------+---------+-------------------------------
+-------------------------+-------------+-----------------------+---------------------
+-----------------+-----------+
| 11   | 11     | 10.0.11.1/24 |              |         |
| Tenant L3 Extended VLAN |              | False                 | False               |
not-provisioned | cfg-ready |
+------+--------+--------------+--------------+---------+-------------------------------
+-------------------------+-------------+-----------------------+---------------------
+-----------------+-----------+
| 12   | 12     | 10.0.12.1/24 |              |         |
| Tenant L3 Extended VLAN |              | False                 | False               |
not-provisioned | cfg-ready |
+------+--------+--------------+--------------+---------+-------------------------------
+-------------------------+-------------+-----------------------+---------------------
+-----------------+-----------+
For 'unstable' entities, run 'efa tenant po/vrf show' for details
================================================================================================
=====================================================================

efa tenant service bgp peer show
================================================================================================
================================================================================================
========
Name        : ten1bgppeer1
Tenant      : ten1
State       : bs-state-created
Description :
Static Peer:
+--------------+----------+------+---------+-----------+------------+-------------
+--------------+--------+----------------+---------------+------------------
+-----------------+-------------+
|  Device IP   |   VRF    | AFI  |  SAFI   | REMOTE IP | REMOTE ASN | BFD Enabled | BFD
Interval | BFD Rx | BFD Multiplier | Next Hop Self | Update Source IP |    Dev-state
|  App-state  |
+--------------+----------+------+---------+-----------+------------+-------------
+--------------+--------+----------------+---------------+------------------
+-----------------+-------------+
| 10.20.246.16 | ten1vrf1 | ipv4 | unicast | 10.1.0.0  | 65001      | false       |
0            | 0      | 0              | false         |                  |
provisioned     | cfg-in-sync |
+--------------+----------+------+---------+-----------+------------+-------------
+--------------+--------+----------------+---------------+------------------
+-----------------+-------------+
| 10.20.246.15 | ten1vrf1 | ipv4 | unicast | 10.0.0.0  | 65001      | false       |
0            | 0      | 0              | false         |                  | not-
provisioned | cfg-ready   |
+--------------+----------+------+---------+-----------+------------+-------------
+--------------+--------+----------------+---------------+------------------
+-----------------+-------------+
Dynamic Peer:
+-----------+-----+-----+------+--------------+------------+--------------+-----------
+-----------+
| Device IP | VRF | AFI | SAFI | Listen Range | Peer Group | Listen Limit | Dev-state |
App-state |
+-----------+-----+-----+------+--------------+------------+--------------+-----------
+-----------+
================================================================================================
```

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

Scenario (Deletion of pre-provisioned config)	EFA 2.3.2	EFA 2.4.0
`efa tenant service bgp peer delete --name ten1bgppeer1 --tenant ten1`	`BgpService Deletion Failed: Device: 10.20.246.15 Bgp Peer Error: Device is administratively down`	`BgpService deleted successfully`
`efa tenant service bgp peer-group delete --name ten1bgppeergroup1 --tenant ten1`	`BgpService deletion Failed: Device: 10.20.246.15 Bgp Peer Error: Device is administratively down`	`BgpService deleted successfully`
`efa tenant epg delete --name ten1epg1 --tenant ten1`	`EndpointGroup deletion failed: Error: [10.20.246.15] device(s) are administratively down`	`EndpointGroup: ten1epg1 deleted successfully`
`efa tenant po delete --name ten1po1 --tenant ten1`	`Portchannel deletion failed: Error: [10.20.246.15] device(s) are administratively down`	`PortChannel: ten1po1 deleted successfully`

In-flight Transaction Recovery

EFA can recover in-flight (in-progress) transactions after a service restart or high-availability failover.

Overview

After a service restart or high-availability failover, EFA can recover in-flight transactions by rolling them back or rolling them forward. In-flight transactions are those that are outstanding in the execution log after a restart or a failover.

- When transactions are rolled back, the requested action is undone.
- When transactions are rolled forward, the requested action is completed.

By default, this feature enables the automatic recovery of Day-1 through Dan-N operations for tenant-related configurations. You can use the **`efa system feature update --inflight-`**

transaction-auto-recovery disable command to disable the feature. When the feature is enabled, the recovery strategy is as follows.

Table 18: Recovery strategy

Operation type	Commands	Strategy
Create operations	• **efa tenant create** • **efa tenant epg create** • **efa tenant po create** • **efa tenant service bgp peer create** • **efa tenant service bgp peer-group create** • **efa tenant vrf create**	Roll back
Delete operations	• **efa tenant delete** • **efa tenant epg delete** • **efa tenant po delete** • **efa tenant service bgp peer delete** • **efa tenant service bgp peer-group delete** • **efa tenant vrf delete**	Roll forward
Update with add operations, such as port-add, ctag-range-add, and vrf-add	• **efa tenant update** • **efa tenant epg update** • **efa tenant po update** • **efa tenant service bgp peer update** • **efa tenant service bgp peer-group update** • **efa tenant vrf update**	Roll back
Update with delete operations, such as port-delete, vrf-delete, and ctag-range-delete	• **efa tenant update** • **efa tenant epg update** • **efa tenant po update** • **efa tenant service bgp peer update** • **efa tenant service bgp peer-group update** • **efa tenant vrf update**	Roll forward

Consider the following expected behaviors for this feature:

- During operations that take a long time, such as drift and reconcile and firmware downloads, tenant operations and recovery operations are blocked.
- When multiple transactions are pending in the execution log after a restart or a failover, recovery occurs in the order in which the operations appear in the execution log.
- If a service restart or high availability failover occurs during transaction recovery, then the status of those recovery operations is changed to a normal status. For example, if a restart occurs during the

rollback of an endpoint group (EPG), the status changes to delete-pending. There is no automatic recovery of interrupted recovery transactions. You must manually verify and address the status of such operations.

Important

Day-0 and administrative operations (those for the Inventory Service and Fabric Service) are not recovered automatically. If these operations are interrupted by a service restart or a failover, you must manually redo the operations.

Examples

This example enables automatic in-flight transaction recovery.

```
efa system feature update --inflight-transaction-auto-recovery enable
Feature Setting Updated Successful
--- Time Elapsed: 634.557118ms ---
```

This example disables automatic in-flight transaction recovery.

```
efa system feature update --inflight-transaction-auto-recovery disable
Feature Setting Updated Successful
--- Time Elapsed: 634.557125ms ---
```

Traffic Mirroring

EFA supports traffic monitoring on both Clos and small data center (non-Clos) fabrics, for troubleshooting issues with applications and fabrics. EFA performs traffic monitoring by means of packet mirroring in a cloud native infrastructure solution and network functions virtualization in infrastructure deployments.

Following are the ports from which you can mirror the ingress and egress traffic:

- Leaf ports connecting to the compute devices
- Leaf ports connecting to the neighboring MCT leaf device (ICL ports)
- Border leaf ports connecting to the external gateway
- Border leaf ports connecting to the neighboring MCT border leaf device (ICL ports)

There are two types of traffic mirroring:

1. In-band traffic mirroring
2. Out-of-band traffic mirroring

Following table describes the comparison between In-band and Out-of-band traffic mirroring solution:

In-band Mirroring	Out-of-band Mirroring
No additional hardware or ports	One additional switch, one reserved port on all leaf and border leaf switches
All configuration by EFA, no separate devices to be managed	Separate configuration on mirror switch through OOB mechanisms

In-band Mirroring	Out-of-band Mirroring
All ingress information, including test access point (TAP) and VLAN, can be retained and used for classification	Ingress port information and possibly VLAN information, is not retained
Fabric needs to be measured for expected extra mirror traffic	Mirroring traffic has minimal impact on normal traffic and fabric capacity, no extra measurement needed
All functionality needs to be present in ingress leaf top of rack (ToR) switch	Minimal configuration needed on EFA, and dataplane support needed in the fabric
Extra tunnel configuration in fabric underlay	Fabric underlay is unmodified
Configuration of underlay tunnels to sink app breaks underlay/overlay separation	Tunnels to sink apps are outside the domain of fabric, and do not overlap
Cannot be applied for control port mirroring	Partial reuse possible for a common mirroring solution also on control network
Fabric has to be programmed for creating additional headers and remote destination reachability, underlay or overlay separation is lost	No fabric dependency on final encapsulation and forwarding toward sink
Egress ACL rule support minimal	Two level filtering possible, once in ingress switch, and once in the dedicated mirror switch, More complicated mirror rules can be cascaded.
QoS support needed on tenant and mirrored traffic streams because they share the same fabric links	No QoS support needed, because links are separate
Cannot be leveraged for troubleshooting fabric issues, due to reliance on fabric	Can be leveraged for troubleshooting fabric issues
Fabric admin needs to do all configuration because underlay routing modifications are needed	EFA tenant admin can create TAP sessions on the fabric switches, with pre-provisioning and custom provisioning of the configuration on mirror switch by fabric admin

Note

For information about commands and supported parameters to configure traffic mirroring, see *Extreme Fabric Automation Command Reference, 2.7.0*.

In-band Traffic Mirroring

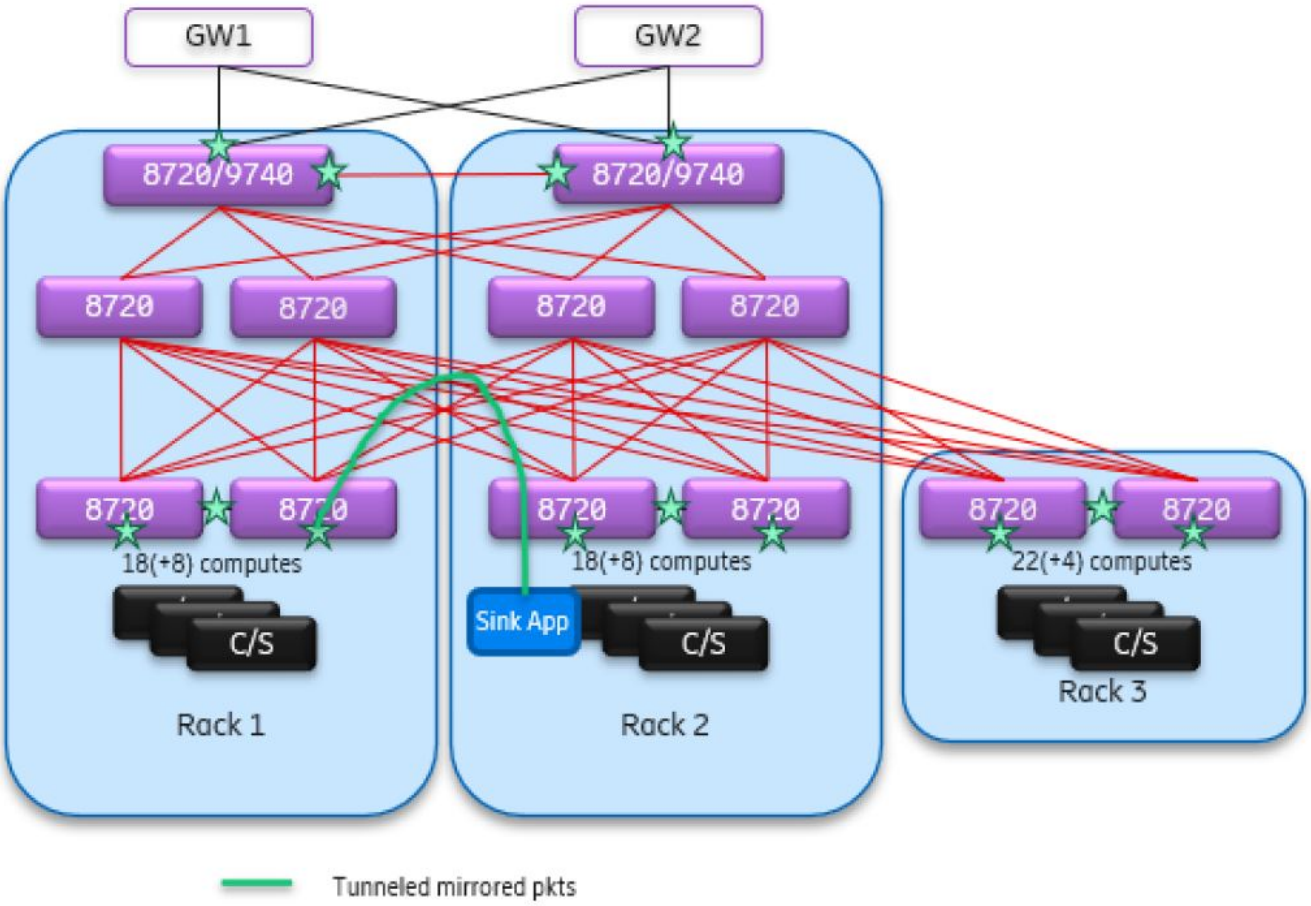


Figure 21: In-band traffic mirroring topology

- Fabric links are used for carrying mirrored and tenant traffic. Mirrored traffic needs to be encapsulated in ERSPAN headers to support multiple sessions.
- Separate tunnels are created over fabric links between the ingress leaf switch and either egress leaf switch or directly to the sink.
- Sink can be deployed in computes, in separate stand-alone servers, or outside the datacenter.
- No separate ports or devices are required for mirroring.

Out-of-band Traffic Mirroring

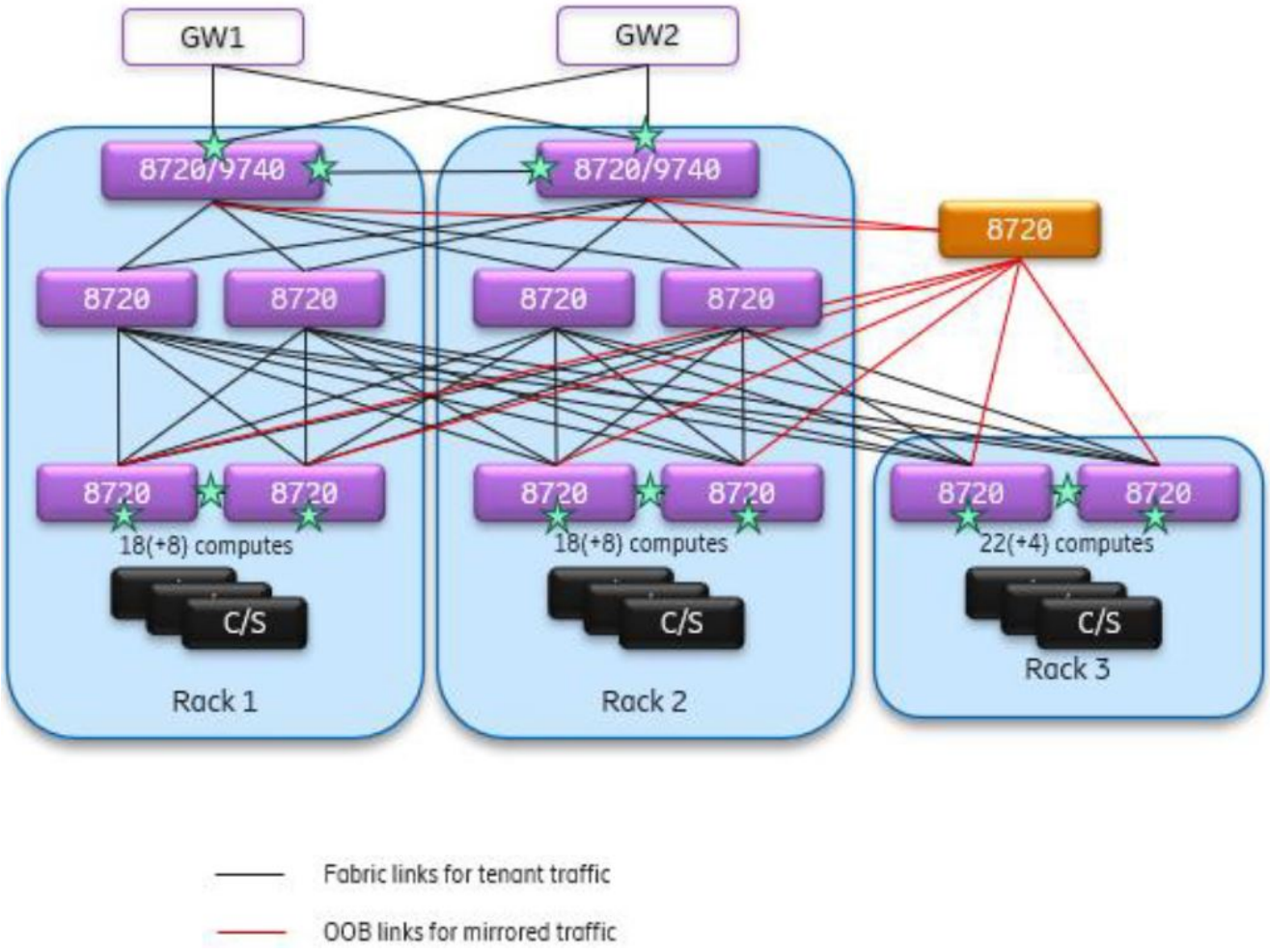


Figure 22: Out-of-band traffic mirroring topology

- Mirrored traffic is captured on the ingress or egress leaf switch and carried on a separate set of links to a separate add-on mirror switch.
- One port is reserved on each fabric leaf and border leaf switch, and connected to the mirror switch through separate OOB cabling.
- EFA configures basic mirroring sessions and actions on fabric switches.
- Advanced configuration on the mirror switch is handled separately, not by EFA.
- Connectivity between the TAP (traffic access point) sink (a class or function designed to receive incoming events from another object or function.) and the mirror switch can be configured and customized separately through OOB mechanisms.
- Perform demultiplexing on sessions involving traffic access (TAP) points, using filtering on packet header fields in the sink application
- 100Gbps links are required between the mirror switch and each fabric switch.
- The mirror switch may be an 8720 or a specialized packet broker with advanced functions.

There are three types of out-of-band traffic mirroring:

- Port-based mirroring
- Flow-based traffic mirroring
- VLAN-based traffic mirroring
- ICL port mirroring

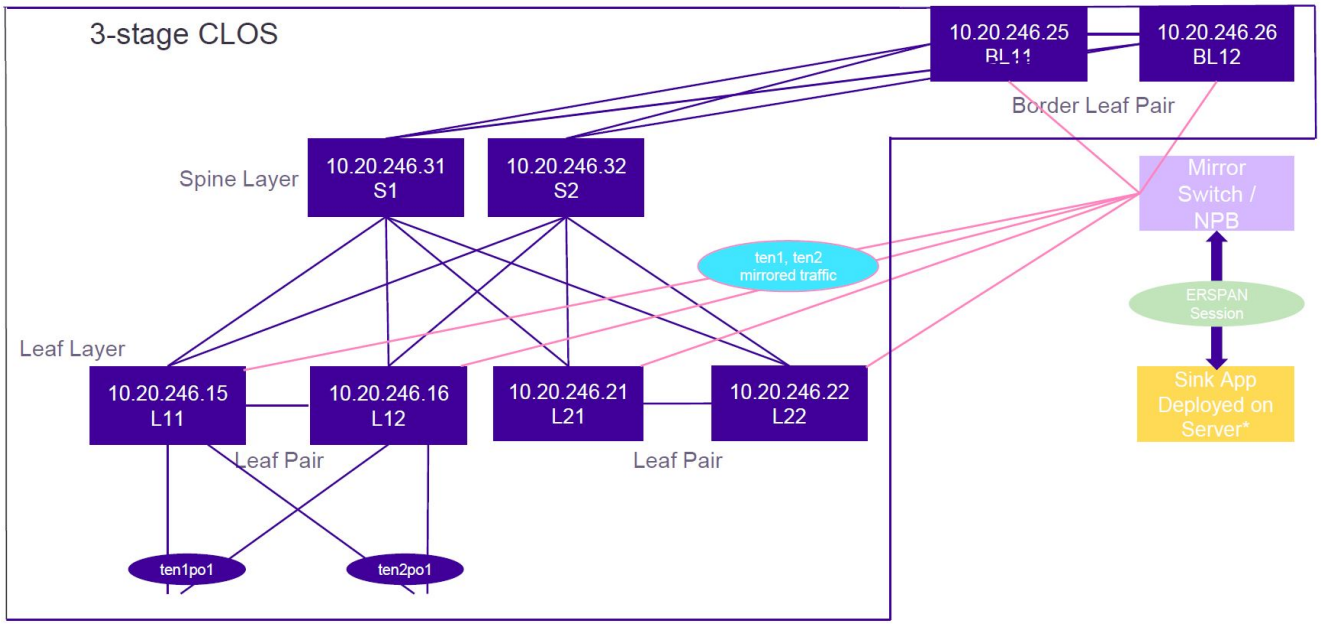


Figure 23: Port-based traffic mirroring topology

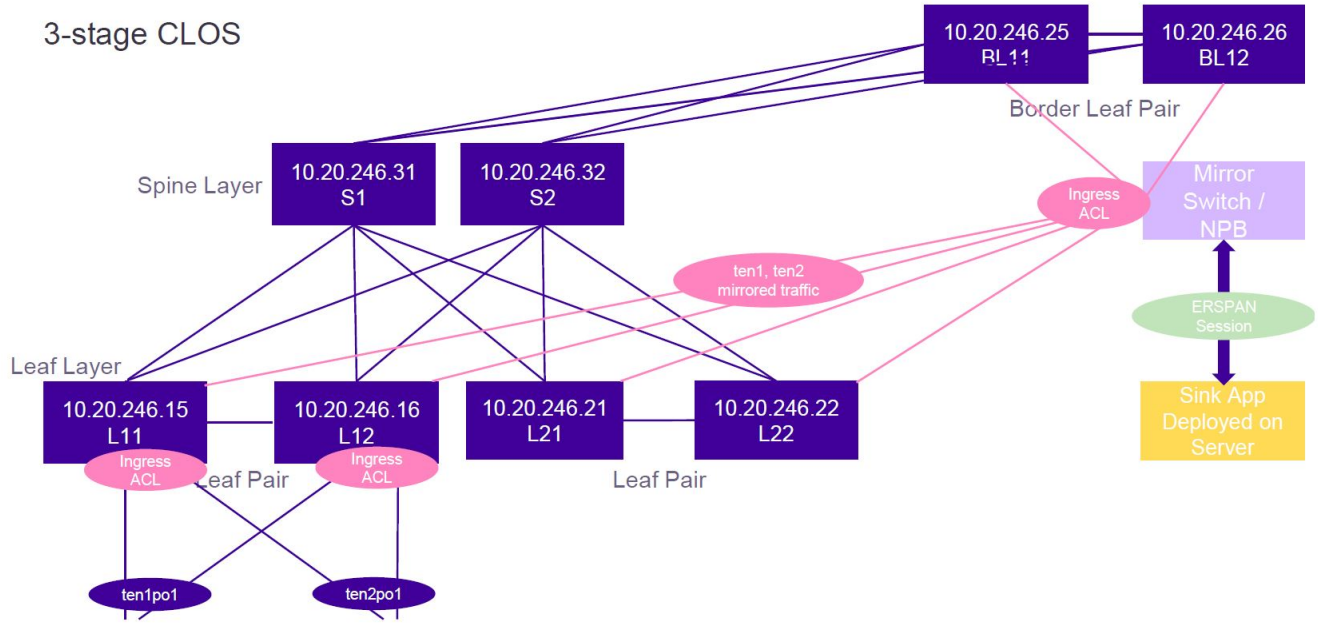


Figure 24: Flow-based traffic mirroring topology

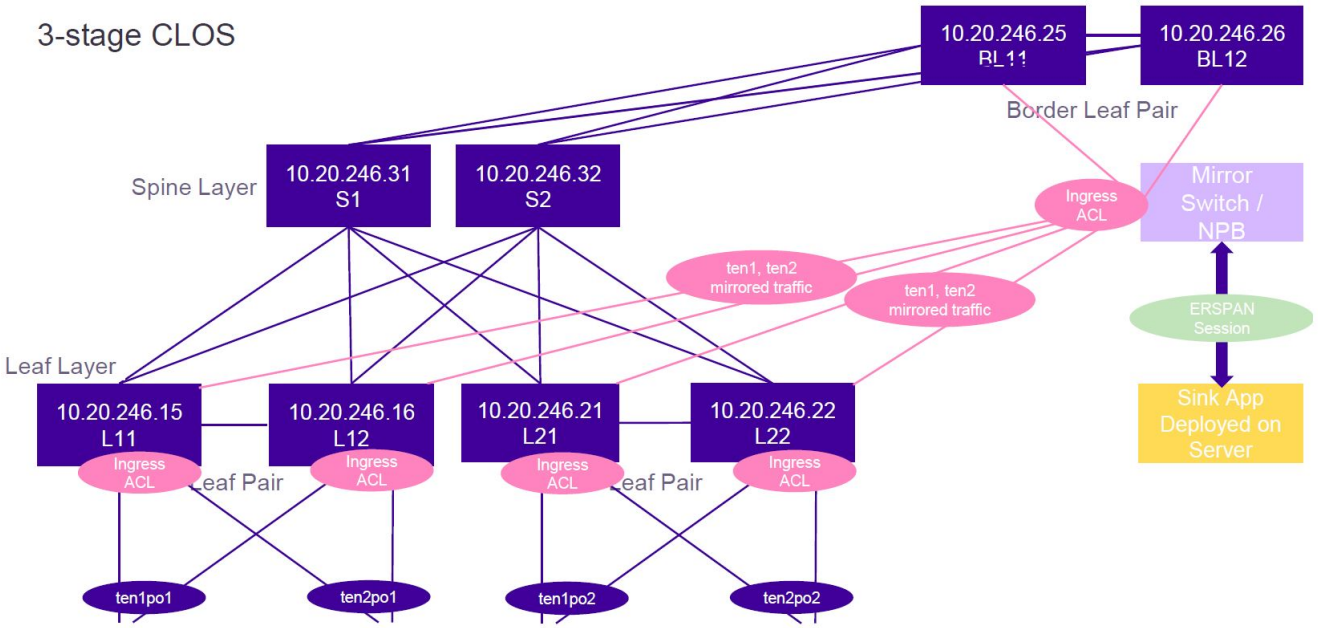


Figure 25: VLAN-based traffic mirroring topology

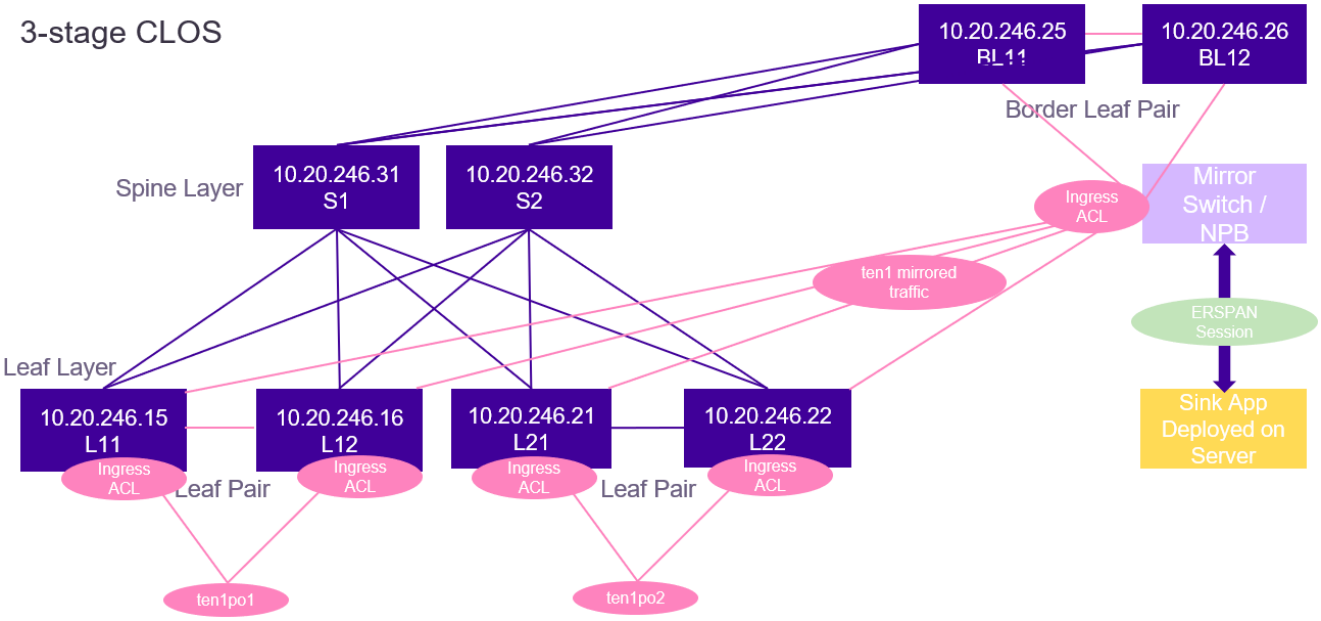


Figure 26: ICL port traffic mirroring topology

Support Matrix

Mirror Type	Mirror Destination Type	Mirror Source	Mirror Destination
Port-Based	Span	Ethernet	Ethernet
Port-Based	Span	Port-Channel	Ethernet

Mirror Type	Mirror Destination Type	Mirror Source	Mirror Destination
Flow-Based	Span	Ethernet	Ethernet
Flow-Based	Span	Port-Channel	Ethernet
Flow-Based	Span	Local VLAN (with device-ip)	Ethernet
Flow-Based	Span	Global VLAN (without device-ip)	Ethernet
Flow-Based	Span	Local VLAN Range (Supported Range format a-b)	Ethernet
Flow-Based	Span	Global VLAN Range (Supported Range format a-b)	Ethernet

	np-mac-acl-in	np-mac-acl-out	np-ip-acl-in	np-ip-acl-out	np-ipv6-acl-in
L2 VLAN EPG	Supported	Supported	Not Supported	Not Supported	Not Supported
L3 VLAN EPG	Not Supported	Not Supported	Supported	Supported	Supported
L2 BD EPG	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported
L3 BD EPG	Not Supported	Not Supported	Supported	Supported	Supported

	pp-mac-acl-in	pp-mac-acl-out	pp-ip-acl-in	pp-ip-acl-out	pp-ipv6-acl-in
Ethernet	Supported	Supported	Supported	Supported	Supported
Port-Channel	Supported	Not supported, except 9740	Supported	Supported	Not supported, except 9740

Note

- Only SPAN is supported as Destination Type and Mirror Direction support is platform dependent
- Only Ethernet interface is supported as Mirror Destination
- Mirror Destination port value is a mandatory for a mirror source during mirror session create, except for a Global VLAN SPAN

Configure Port-Based Mirroring in a Multi-Tenant Architecture

1. Run the following commands to configure access control list applications on Ethernet or port channel and VLAN or virtual Ethernet:

```
efa tenant service mirror session create --name <session-name> --tenant <tenant-name>

           --source {<device-ip>,<eth | po | vlan>,<if-name>}
           --type {<source-device-ip>,<eth | po | vlan>,<source-if-name>:<port-based | flow-based>}

           --destination {<source-device-ip>,<eth | po | vlan>,<source-if-name> :
                         <destination-device-ip>,<eth | po | vlan>,<destination-if-name}
           --destination-type {<source-device-ip>,< eth | po | vlan>,<source-if-name>:<span>}


           --direction {<source-device-ip>,< eth | po | vlan>,<source-if-name> : <tx | rx | both>}
```

```
(efa:root)root@node-2:~# efa tenant show
+--------------+---------+------------+-------------+-------------+-----------+-----------
+-----------------------+---------------------+
|     Name     |  Type   | VLAN Range | L2VNI Range | L3VNI Range | VRF Count | Enable BD |
Ports         |  Mirroring Ports    |
+--------------+---------+------------+-------------+-------------+-----------+-----------
+-----------------------+---------------------+
| sharedTenant | shared  |            |             |             |     0     |   false
|                       | 10.20.246.15[0/31]  |
|              |         |            |             |             |           |
|                       | 10.20.246.16[0/31]  |
|              |         |            |             |             |           |
|                       | 10.20.246.21[0/31]  |
|              |         |            |             |             |           |
|                       | 10.20.246.22[0/31]  |
|              |         |            |             |             |           |
|                       | 10.20.246.25[0/31]  |
|              |         |            |             |             |           |
|                       | 10.20.246.26[0/31]  |
+--------------+---------+------------+-------------+-------------+-----------+-----------
+-----------------------+---------------------+
|     ten1     | private |   11-20    |             |             |    10     |   false   |
10.20.246.15[0/1-10]  |                     |
|              |         |            |             |             |           |           |
10.20.246.16[0/1-10]  |                     |
+--------------+---------+------------+-------------+-------------+-----------+-----------
+-----------------------+---------------------+
|     ten2     | private |   21-30    |             |             |    10     |   false   |
10.20.246.15[0/11-20] |                     |
|              |         |            |             |             |           |           |
10.20.246.16[0/11-20] |                     |
+--------------+---------+------------+-------------+-------------+-----------+-----------
+-----------------------+---------------------+


(efa:root)root@node 2:~# efa tenant po show
+---------+--------+--------------+-----+-------------+----------+---------+-------------------
+------------+-------------+-------------+
|   Name  | Tenant | ID |  Speed  | MTU | Negotiation | Min Link |  Lacp   |        Ports      |
State   |  Dev State  |  App State  |
|         |        |    |         |     |             |   Count  | Timeout |
|           |             |             |
+---------+--------+------+--------+-----+-------------+----------+---------+-------------------
+------------+-------------+-------------+
| ten1po1 | ten1   |  2 |  10Gbps |     |    active   |    1     |   long  | 10.20.246.15[0/1] | po-
created | provisioned | cfg-in-sync |
|         |        |    |         |     |             |          |         | 10.20.246.16[0/1]
|            |             |             |
+---------+--------+------+--------+-----+-------------+----------+---------+-------------------
+------------+-------------+-------------+
| ten2po1 | ten2   |  3 |  10Gbps |     |    active   |    1     |   long  | 10.20.246.15[0/11]| po-
created | provisioned | cfg-in-sync |
|         |        |    |         |     |             |          |         |
10.20.246.16[0/11]|           |             |           |
+---------+--------+----+---------+-----+-------------+----------+---------+-------------------
+------------+-------------+-------------+
```

Example:

10.20.246.15	10.20.246.16
efa tenant service mirror session create --name ten1mirrorsession1 --tenant ten1 --source 10.20.246.15,po,ten1po1 **--type 10.20.246.15,po,ten1po1:port-based** --destination 10.20.246.15,po,ten1po1:10.20.246.15,eth,0/31 **--destination-type 10.20.246.15,po,ten1po1:span** --direction 10.20.246.15,po,ten1po1:both	efa tenant service mirror session create –name ten1mirrorsession2 --tenant ten1 --source 10.20.246.16,po,ten1po1 **--type 10.20.246.16,po,ten1po1:port-based** --destination 10.20.246.16,po,ten1po1:10.20.246.16,eth,0/31 **--destination-type 10.20.246.16,po,ten1po1:span** --direction 10.20.246.16,po,ten1po1:both
10.20.246.15	10.20.246.16
efa tenant service mirror session create –name ten2mirrorsession1 --tenant ten2 --source 10.20.246.15,po,ten2po1 **--type 10.20.246.15,po,ten2po1:port-based** --destination 10.20.246.15,po,ten2po1:10.20.246.15,eth,0/31 **--destination-type 10.20.246.15,po,ten2po1:span** --direction 10.20.246.15,po,ten2po1:both	efa tenant service mirror session create –name ten2mirrorsession2 --tenant ten2 --source 10.20.246.16,po,ten2po1 **--type 10.20.246.16,po,ten2po1:port-based** --destination 10.20.246.16,po,ten2po1:10.20.246.16,eth,0/31 **--destination-type 10.20.246.16,po,ten2po1:span** --direction 10.20.246.16,po,ten2po1:both

2. Verify the switch configuration on the SLX device.

10.20.246.15

```
SLX# show running-config monitor
session
monitor session 1
 source port-channel 2 destination
ethernet 0/31 direction both
!monitor session 2
 source port-channel 3 destination
ethernet 0/31 direction both
!
SLX# show monitor session 1
Session                  : 1
Type                     : SPAN
Description              : [None]
State                    : Enabled
Source Interface         : Po 2
(Down)
Destination Interface    : Eth 0/31
(Down)
Direction                : Both
Type                     : port-
based

SLX# show monitor session 2
Session                  : 2
Type                     : SPAN
Description              : [None]
State                    : Enabled
Source Interface         : Po 3
(Down)
Destination Interface    : Eth 0/31
(Down)
Direction                : Both
Type                     : port-
based
```

10.20.246.16

```
SLX# show running-config monitor
session
monitor session 1
 source port-channel 2 destination
ethernet 0/31 direction both
!monitor session 2
 source port-channel 3 destination
ethernet 0/31 direction both
!
SLX# show monitor session 1
Session                  : 1
Type                     : SPAN
Description              : [None]
State                    : Enabled
Source Interface         : Po 2
(Down)
Destination Interface    : Eth 0/31
(Down)
Direction                : Both
Type                     : port-
based

SLX# show monitor session 2
Session                  : 2
Type                     : SPAN
Description              : [None]
State                    : Enabled
Source Interface         : Po 3
(Down)
Destination Interface    : Eth 0/31
(Down)
Direction                : Both
Type                     : port-
based
```

Configure Flow-Based Mirroring in a Multi-Tenant Architecture

1. Run the following commands to configure access control list applications on ethernet or port channel and VLAN or virtual ethernet:

```
efa tenant epg create --name <epg-name> --tenant <tenant-name>
                --switchport --switchport-mode trunk -ctag-range <ctag-range>
                --port <mirror-source-port-list> --po <mirror-source-po-list>

                --pp-mac-acl-in <acl-name> --pp-mac-acl-out <acl-name>
                --pp-acl-in <acl-name> --pp-ip-acl-out <acl-name>

                --np-mac-acl-in <ctag:acl-name> --np-mac-acl-out <ctag:acl-name>
                --np-ip-acl-in <ctag:acl-name> --np-ip-acl-out <ctag:acl-name>
```

2. Run the following commands to configure a mirror session:

```
efa tenant service mirror session create -name? <session-name> --tenant <tenant-name>
                   --source {<device-ip>,<eth | po | vlan>,<if-name>}
                   --type {<source-device-ip>,<eth | po | vlan>,<source-if-
name>:<port-based | flow-based>}

                --destination {<source-device-ip>,<eth | po | vlan>,<source-if-
name> :
                   <destination-device-ip>,<eth | po | vlan>,<destination-if-name}
```

```
                    --destination-type {<source-device-ip>,< eth | po | vlan>,<source-
if-name>:<span>}

                    --direction {<source-device-ip>,< eth | po | vlan>,<source-if-
name> : <tx | rx | both>}

(efa:root)root@node-2:~# efa tenant show
+--------------+---------+------------+-------------+-------------+-----------
+-----------+-----------------------+---------------------+
|     Name     |  Type   | VLAN Range | L2VNI Range | L3VNI Range | VRF Count | Enable
BD |         Ports         |  Mirroring Ports    |
+--------------+---------+------------+-------------+-------------+-----------
+-----------+-----------------------+---------------------+
| sharedTenant | shared  |            |             |             |     0     |
false   |                       | 10.20.246.15[0/31]  |
|              |         |            |             |             |
|           |                         | 10.20.246.16[0/31]  |
|              |         |            |             |             |
|           |                         | 10.20.246.21[0/31]  |
|              |         |            |             |             |
|           |                         | 10.20.246.22[0/31]  |
|              |         |            |             |             |
|           |                         | 10.20.246.25[0/31]  |
|              |         |            |             |             |
|           |                         | 10.20.246.26[0/31]  |
+--------------+---------+------------+-------------+-------------+-----------
+-----------+-----------------------+---------------------+
|     ten1     | private |   11-20    |             |             |    10     |
false   | 10.20.246.15[0/1-10]  |                     |
|              |         |            |             |             |
|           | 10.20.246.16[0/1-10]  |                     |
+--------------+---------+------------+-------------+-------------+-----------
+-----------+-----------------------+---------------------+
|     ten2     | private |   21-30    |             |             |    10     |
false   | 10.20.246.15[0/11-20] |                     |
|              |         |            |             |             |
|           | 10.20.246.16[0/11-20] |                     |
+--------------+---------+------------+-------------+-------------+-----------
+-----------+-----------------------+---------------------+


(efa:root)root@node 2:~# efa tenant po show
+---------+--------+--------------+-----+-------------+----------+---------
+--------------------+------------+-------------+-------------+
|   Name  | Tenant | ID |  Speed  | MTU | Negotiation | Min Link |  Lacp   |
Ports      |    State   |  Dev State  |  App State  |
|         |        |    |         |     |             |   Count  | Timeout
|                    |            |             |             |
+---------+--------+--------------+-----+-------------+----------+---------
+--------------------+------------+-------------+-------------+
| ten1po1 | ten1   |  2 |  10Gbps |     |    active   |    1     |   long  |
10.20.246.15[0/1] | po-created | provisioned | cfg-in-sync |
|         |        |    |         |     |             |          |         |
10.20.246.16[0/1] |            |             |             |
+---------+--------+--------------+-----+-------------+----------+---------
+--------------------+------------+-------------+-------------+
| ten2po1 | ten2   |  3 |  10Gbps |     |    active   |    1     |   long  |
10.20.246.15[0/11]| po-created | provisioned | cfg-in-sync |
|         |        |    |         |     |             |          |         |
10.20.246.16[0/11]|            |             |             |
+---------+--------+----+---------+-----+-------------+----------+---------
+--------------------+------------+-------------+-------------+
```

Example

```
efa tenant epg create –name
ten1epg1 –tenant ten1
            --switchport-mode
trunk --po ten1po1 --ctag-range 11
            --pp-ip-acl-in ext-ip-
permit-any-mirror-acl
            --pp-ip-acl-out  ext-
ip-permit-any-mirror-acl

efa tenant service mirror session
create –name ten1mirrorsession1 --
tenant ten1
            --source
10.20.246.15,po,ten1po1
            --type
10.20.246.15,po,ten1po1:flow-based

            --destination
10.20.246.15,po,ten1po1:10.20.246.1
5,eth,0/31
            --destination-type
10.20.246.15,po,ten1po1:span

            --direction
10.20.246.15,po,ten1po1:both

efa tenant service mirror session
create –name ten2mirrorsession1 --
tenant ten2
            --source
10.20.246.15,po,ten2po1
            --type
10.20.246.15,po,ten2po1:flow-based

            --destination
10.20.246.15,po,ten2po1:10.20.246.1
5,eth,0/31
            --destination-type
10.20.246.15,po,ten2po1:span

            --direction
10.20.246.15,po,ten2po1:both
```

```
efa tenant epg create –name
ten2epg1 –tenant ten2
            --switchport-mode
trunk --po ten2po1 --ctag-range 21
            --pp-ip-acl-in ext-ip-
permit-any-mirror-acl
            --pp-ip-acl-out ext-ip-
permit-any-mirror-acl



efa tenant service mirror session
create –name ten1mirrorsession2 --
tenant ten1
            --source
10.20.246.16,po,ten1po1
            --type
10.20.246.16,po,ten1po1:flow-based

            --destination
10.20.246.16,po,ten1po1:10.20.246.1
6,eth,0/31
            --destination-type
10.20.246.16,po,ten1po1:span

            --direction
10.20.246.16,po,ten1po1:both

efa tenant service mirror session
create –name ten2mirrorsession2 --
tenant ten2
            --source
10.20.246.16,po,ten2po1
            --type
10.20.246.16,po,ten2po1:flow-
based

            --destination
10.20.246.16,po,ten2po1:10.20.246.1
6,eth,0/31
            --destination-type
10.20.246.16,po,ten2po1:span

            --direction
10.20.246.16,po,ten2po1:both
```

3. Verify the switch configuration on the SLX device.

10.20.246.15

```
SLX# show running-config ip access-list
ip access-list extended ext-ip-permit-any-mirror-acl
 seq 10 permit ip any any mirror
!
SLX# show running-config interface Port-channel 2,3
interface Port-channel 2
 description EFA Port-channel ten1po1
 cluster-client auto
 switchport
 switchport mode trunk
 switchport trunk allowed vlan add 11
 no switchport trunk tag native-vlan
 ip access-group ext-ip-permit-any-mirror-acl in
 ip access-group ext-ip-permit-any-mirror-acl out
 no shutdown
!
interface Port-channel 3
 description EFA Port-channel ten2po1
 cluster-client auto
 switchport
 switchport mode trunk
 switchport trunk allowed vlan add 21
 no switchport trunk tag native-vlan
 ip access-group ext-ip-permit-any-mirror-acl in
 ip access-group ext-ip-permit-any-mirror-acl out
 no shutdown
!
SLX#
```

10.20.246.16

```
SLX# show running-config ip access-list
ip access-list extended ext-ip-permit-any-mirror-acl
 seq 10 permit ip any any mirror
!
SLX# show running-config interface Port-channel 2,3
interface Port-channel 2
 description EFA Port-channel ten1po1
 cluster-client auto
 switchport
 switchport mode trunk
 switchport trunk allowed vlan add 11
 no switchport trunk tag native-vlan
 ip access-group ext-ip-permit-any-mirror-acl in
 ip access-group ext-ip-permit-any-mirror-acl out
 no shutdown
!
interface Port-channel 3
 description EFA Port-channel ten2po1
 cluster-client auto
 switchport
 switchport mode trunk
 switchport trunk allowed vlan add 21
 no switchport trunk tag native-vlan
 ip access-group ext-ip-permit-any-mirror-acl in
 ip access-group ext-ip-permit-any-mirror-acl out
 no shutdown
!
SLX#
```

10.20.246.15

```
SLX# show running-config monitor session
monitor session 1
 source port-channel 2 destination ethernet 0/31 direction both
!monitor session 2
 source port-channel 3 destination ethernet 0/31 direction both
!
SLX# show monitor session 1
Session                 : 1
Type                    : SPAN
Description             : [None]
State                   : Enabled
Source Interface        : Po 2 (Down)
Destination Interface   : Eth 0/31 (Down)
```

10.20.246.16

```
SLX# show running-config monitor session
monitor session 1
 source port-channel 2 destination ethernet 0/31 direction both
!monitor session 2
 source port-channel 3 destination ethernet 0/31 direction both
!
SLX# show monitor session 1
Session                 : 1
Type                    : SPAN
Description             : [None]
State                   : Enabled
Source Interface        : Po 2 (Down)
Destination Interface   : Eth 0/31 (Down)
```

```
Direction                : Both
Type                     : flow-based

SLX# show monitor session 2
Session                  : 2
Type                     : SPAN
Description              : [None]
State                    : Enabled
Source Interface         : Po 3
(Down)
Destination Interface    : Eth 0/31
(Down)
Direction                : Both
Type                     : flow-based
```

```
Direction                : Both
Type                     : flow-based

SLX# show monitor session 2
Session                  : 2
Type                     : SPAN
Description              : [None]
State                    : Enabled
Source Interface         : Po 3
(Down)
Destination Interface    : Eth 0/31
(Down)
Direction                : Both
Type                     : flow-based
```

Access Control List and Data Consistency Support

EFA provisions the Access Control List (ACL) internally because there is no ACL CRUD support available from the EFA side.

Full-fledged data consistency support (any drift in the ACL rules is identified and reconciled) is available when the ACL CRUD is supported via EFA.

Configure VLAN-Based Mirroring in a Multi-Tenant Architecture

VLAN based mirroring applies only to VLAN-based tenants and not to BD (bridge domain)-based tenants.

1. Run the following commands to configure access control list applications on Ethernet or port channel and VLAN or virtual Ethernet:

```
efa tenant epg create --name <epg-name> --tenant <tenant-name>

            --switchport --switchport-mode trunk -ctag-range <ctag-range>
            --port <mirror-source-port-list> --po <mirror-source-po-list>

            --pp-mac-acl-in <acl-name> --pp-mac-acl-out <acl-name>
            --pp-ip-acl-in <acl-name> --pp-ip-acl-out <acl-name>

            --np-mac-acl-in <ctag:acl-name> --np-mac-acl-out <ctag:acl-name>
            --np-ip-acl-in <ctag:acl-name> --np-ip-acl-out <ctag:acl-name>
```

2. Run the following commands to configure a mirror session:

```
efa tenant service mirror session create -name <session-name> --tenant <tenant-name>

            --source {<device-ip>,<eth | po | vlan>,<if-name>}
            --type {<source-device-ip>,<eth | po | vlan>,<source-if-name>:<port-based
| flow-based>}

            --destination {<source-device-ip>,<eth | po | vlan>,<source-if-name> :
                        <destination-device-ip>,<eth | po | vlan>,<destination-if-name}
            --destination-type {<source-device-ip>,< eth | po | vlan>,<source-if-
name>:<span>}

            --direction {<source-device-ip>,< eth | po | vlan>,<source-if-name> : <tx
| rx | both>}


(efa:root)root@node-2:~# efa tenant show
+--------+---------+-------+-------+-------+-------+--------+----------------------
```

```
+--------------------+
| Name   |  Type   | VLAN  | L2VNI | L3VNI | VRF   | Enable |         Ports        |
Mirroring Ports  |
|        |         | Range | Range | Range | Count | BD     |
|                    |
+--------+---------+-------+-------+-------+-------+--------+----------------------
+--------------------+
| shared | Shared  |       |       |       |   0   | false  |                      |
10.20.246.16[0/31] |
| Tenant |         |       |       |       |       |        |                      |
10.20.246.21[0/31] |
|        |         |       |       |       |       |        |                      |
10.20.246.22[0/31] |
|        |         |       |       |       |       |        |                      |
10.20.246.25[0/31] |
|        |         |       |       |       |       |        |                      |
10.20.246.26[0/31] |
+--------+---------+-------+-------+-------+-------+--------+----------------------
+--------------------+
|  ten1  | private | 11-20 |       |       |   10  | false  | 10.20.246.15[0/1-10]
|                    |
|        |         |       |       |       |       |        | 10.20.246.16[0/1-10]
|                    |
|        |         |       |       |       |       |        | 10.20.246.21[0/1-10]
|                    |
|        |         |       |       |       |       |        | 10.20.246.22[0/1-10]
|                    |
+--------+---------+-------+-------+-------+-------+--------+----------------------
+--------------------+
|  ten2  | private | 21-30 |       |       |   10  | false  |
10.20.246.15[0/11-20]|                   |
|        |         |       |       |       |       |        |
10.20.246.16[0/11-20]|                   |
|        |         |       |       |       |       |        |
10.20.246.21[0/11-20]|                   |
|        |         |       |       |       |       |        |
10.20.246.22[0/11-20]|                   |
+--------+---------+-------+-------+-------+-------+--------+----------------------
+--------------------+


(efa:root)root@node 2:~# efa tenant po show
+---------+--------+--------------+-----+-------------+----------+---------
+--------------------+------------+-------------+-------------+
|   Name  | Tenant | ID |  Speed  | MTU | Negotiation | Min Link |  Lacp   |
Ports      |    State   |  Dev State  |  App State  |
|         |        |    |         |     |             |   Count  | Timeout
|                    |            |             |             |
+---------+--------+--------------+-----+-------------+----------+---------
+--------------------+------------+-------------+-------------+
| ten1po1 | ten1   |  2 |  10Gbps |     |    active   |    1     |   long  |
10.20.246.15[0/1] | po-created | provisioned | cfg-in-sync |
|         |        |    |         |     |             |          |         |
10.20.246.16[0/1] |            |             |             |
+---------+--------+--------------+-----+-------------+----------+---------
+--------------------+------------+-------------+-------------+
| ten2po1 | ten2   |  3 |  10Gbps |     |    active   |    1     |   long  |
10.20.246.15[0/11]| po-created | provisioned | cfg-in-sync |
|         |        |    |         |     |             |          |         |
10.20.246.16[0/11]|            |             |             |
+---------+--------+--------------+-----+-------------+----------+---------
+--------------------+------------+-------------+-------------+
| ten1po2 | ten1   |  2 |  10Gbps |     |    active   |    1     |   long  |
10.20.246.21[0/1] | po-created | provisioned | cfg-in-sync |
```

```
|         |        |    |         |     |             |          |         |
10.20.246.22[0/1] |            |            |             |
+---------+--------+------------+-----+-------------+----------+---------
+-------------------+------------+-------------+-------------+
| ten2po2 | ten2    |  3 |  10Gbps |     |    active   |    1     |   long  |
10.20.246.21[0/11]| po-created | provisioned | cfg-in-sync |
|         |        |    |         |     |             |          |         |
10.20.246.22[0/11]|            |             |             |
+---------+--------+------------+-----+-------------+----------+---------
+-------------------+------------+-------------+-------------+
```

Example

```
efa tenant epg create -name
ten1epg1 -tenant ten1
            --switchport-mode
trunk --po ten1po1,ten1po2  --ctag-
range 11
            --np-mac-acl-in 11:ext-
mac-permit-any-mirror-acl
            --np-mac-acl-out
11:ext-mac-permit-any-mirror-acl


efa tenant service mirror session
create -name ten1mirrorsession1 --
tenant ten1
            --source vlan,11
            --type vlan,11:flow-
based
            --destination-type
vlan,11:span
            --destination
vlan,11:10.20.246.15,eth,0/31
            --direction
vlan,11:both
```

```
efa tenant epg create -name
ten2epg1 -tenant ten2
            --switchport-mode
trunk --po ten2po1,ten2po2 --ctag-
range 21
            --np-mac-acl-in 21:ext-
mac-permit-any-mirror-acl
            --np-mac-acl-out
21:ext-mac-permit-any-mirror-acl


efa tenant service mirror session
create -name ten2mirrorsession1 --
tenant ten2
            --source vlan,21
            --type vlan,21:flow-
based
            --destination-type
vlan,21:span
            --destination
vlan,21:10.20.246.16,eth,0/31
            --direction
vlan,21:both
```

3. Verify the switch configuration on the SLX device.

10.20.246.15

```
SLX# show running-config mac access-list
mac access-list extended ext-mac-permit-any-mirror-acl
 seq 10 permit any any mirror
!
SLX#

SLX# show running-config vlan 11,21
vlan 11
 description Tenant L2 Extended VLAN
 mac access-group ext-mac-permit-any-mirror-acl in
 mac access-group ext-mac-permit-any-mirror-acl out
!
vlan 21
 description Tenant L2 Extended VLAN
 mac access-group ext-mac-permit-any-mirror-acl in
 mac access-group ext-mac-permit-any-mirror-acl out
!
SLX#
```

10.20.246.16

```
SLX# show running-config mac access-list
mac access-list extended ext-mac-permit-any-mirror-acl
 seq 10 permit any any mirror
!
SLX#

SLX# show running-config vlan 11,21
vlan 11
 description Tenant L2 Extended VLAN
 mac access-group ext-mac-permit-any-mirror-acl in
 mac access-group ext-mac-permit-any-mirror-acl out
!
vlan 21
 description Tenant L2 Extended VLAN
 mac access-group ext-mac-permit-any-mirror-acl in
 mac access-group ext-mac-permit-any-mirror-acl out
!
SLX#
```

10.20.246.21

```
SLX# show running-config mac access-list
mac access-list extended ext-mac-permit-any-mirror-acl
 seq 10 permit any any mirror
!
SLX#

SLX# show running-config vlan 11,21
vlan 11
 description Tenant L2 Extended VLAN
 mac access-group ext-mac-permit-any-mirror-acl in
 mac access-group ext-mac-permit-any-mirror-acl out
!
vlan 21
 description Tenant L2 Extended VLAN
 mac access-group ext-mac-permit-any-mirror-acl in
 mac access-group ext-mac-permit-any-mirror-acl out
!
SLX#
```

10.20.246.22

```
SLX# show running-config mac access-list
mac access-list extended ext-mac-permit-any-mirror-acl
 seq 10 permit any any mirror
!
SLX#

SLX# show running-config vlan 11,21
vlan 11
 description Tenant L2 Extended VLAN
 mac access-group ext-mac-permit-any-mirror-acl in
 mac access-group ext-mac-permit-any-mirror-acl out
!
vlan 21
 description Tenant L2 Extended VLAN
 mac access-group ext-mac-permit-any-mirror-acl in
 mac access-group ext-mac-permit-any-mirror-acl out
!
SLX#
```

10.20.246.15-16

```
SLX# show running-config monitor session
monitor session 1
 source vlan 11 destination ethernet 0/31 direction both flow-based
!
monitor session 2
 source vlan 21 destination ethernet 0/31 direction both flow-based
!SLX#

SLX# show monitor session 1
Session                 : 1
Type                    : SPAN
Description             : [None]
```

10.20.246.21-22

```
SLX# show running-config monitor session
monitor session 1
 source vlan 11 destination ethernet 0/31 direction both flow-based
!
monitor session 2
 source vlan 21 destination ethernet 0/31 direction both flow-based
!SLX

#SLX# show monitor session 1
Session                 : 1
Type                    : SPAN
Description             : [None]
```

```
State                       : Enabled
Source Interface            : Vlan 11
Destination Interface       : Eth 0/31
(Down)
Direction                   : Both
Type                        : flow-based

SLX# show monitor session 2
Session                     : 2
Type                        : SPAN
Description                 : [None]
State                       : Enabled
Source Interface            : Vlan 21
Destination Interface       : Eth 0/31
(Down)
Direction                   : Both
Type                        : flow-based
SLX#
```

```
State                       : Enabled
Source Interface            : Vlan 11
Destination Interface       : Eth 0/31
(Down)
Direction                   : Both
Type                        : flow-based

SLX# show monitor session 2
Session                     : 2
Type                        : SPAN
Description                 : [None]
State                       : Enabled
Source Interface            : Vlan 21
Destination Interface       : Eth 0/31
(Down)
Direction                   : Both
Type                        : flow-based
SLX#
```

Configure ICL Port Mirroring in a Multi-Tenant Architecture

1. Run the following commands to configure access control list applications on Ethernet or port channel and VLAN or virtual Ethernet:

```
efa tenant epg create --name <epg-name> --tenant <tenant-name>

        --type port-profile
        --po <mirror-source-po-list>

        --pp-ipv6-acl-in <acl-name>
        --pp-ip-acl-in <acl-name> --pp-ip-acl-out <acl-name>
```

2. Run the following commands to configure a mirror session:

```
efa tenant service mirror session create -name <session-name> --tenant <tenant-name>
    --source {<device-ip>,<eth | po | vlan>,<if-name>}
    --type {<source-device-ip>,<eth | po | vlan>,<source-if-name>:<port-based | flow-
based>}

    --destination-type {<source-device-ip>,< eth | po | vlan>,<source-if-name>:<span>}
    --destination {<source-device-ip>,<eth | po | vlan>,<source-if-name> :
            <destination-device-ip>,<eth | po | vlan>,<destination-if-name}
    --direction {<source-device-ip>,< eth | po | vlan>,<source-if-name> : <tx | rx |
both>}


(efa:root)root@node-2:~# efa tenant show
+--------+--------+-------+-------+-------+-------+--------
+-----------------------------+--------------------+
| Name   |  Type  | VLAN  | L2VNI | L3VNI | VRF   | Enable |
Ports             |   Mirroring Ports  |
|        |        | Range | Range | Range | Count | BD
|                              |                    |
+--------+--------+-------+-------+-------+-------+--------
+-----------------------------+--------------------+
| shared | Shared |       |       |       |   0   | false  |
10.20.246.15[0/46-47]       | 10.20.246.15[0/31] |
| Tenant |        |       |       |       |       |        |
10.20.246.16[0/46-47]       | 10.20.246.16[0/31] |
|        |        |       |       |       |       |        |
10.20.246.21[0/9-10,0/46-48]| 10.20.246.21[0/31] |
|        |        |       |       |       |       |        |
10.20.246.22[0/9-10,0/46-48]| 10.20.246.22[0/31] |
```

```
|         |         |       |       |       |       |
|                             | 10.20.246.25[0/31] |
|         |         |       |       |       |       |
|                             | 10.20.246.26[0/31] |
+--------+--------+-------+-------+-------+-------+--------
+-----------------------------+--------------------+


(efa:root)root@node 2:~# efa tenant po show
+---------+--------+----+--------+-----+-------------+----------+---------
+-----------------------------+------------+-------------+-------------+
|   Name  | Tenant | ID | Speed  | MTU | Negotiation | Min Link |  Lacp   |
Ports               |    State   |  Dev State  |  App State  |
|         |        |    |        |     |             |   Count  | Timeout
|                             |            |             |             |
+---------+--------+----+--------+-----+-------------+----------+---------
+-----------------------------+------------+-------------+-------------+
| ten1po1 | ten1   | 64 | 10Gbps |     |    active   |    1     |   long  |
10.20.246.15[0/46-47]       | po-created | provisioned | cfg-in-sync |
|         |        |    |        |     |             |          |         |
10.20.246.16[0/46-47]       |            |             |             |
+---------+--------+----+--------+-----+-------------+----------+---------
+-----------------------------+------------+-------------+-------------+
| ten2po1 | ten2   | 64 | 10Gbps |     |    active   |    1     |   long  |
10.20.246.21[0/9-10,0/46-48]| po-created | provisioned | cfg-in-sync |
|         |        |    |        |     |             |          |         |
10.20.246.22[0/9-10,0/46-48]|            |             |             |
+---------+--------+----+--------+-----+-------------+----------+---------
+-----------------------------+------------+-------------+-------------+
```

Example

```
efa tenant epg create –name
ten1epg1 –tenant ten1 --type port-
profile
         --po ten1po1
         --pp-ipv6-acl-in ext-ipv6-
permit-any-mirror-acl
efa tenant service mirror session
create –name mirrorsession1 --
tenant ten1
              --source
10.20.246.15,po,ten1po1
              --type
10.20.246.15,po,ten1po1:port-based
              --destination
10.20.246.15,po,ten1po1:10.20.246.1
5,eth,0/31
              --destination-type
10.20.246.15,po,ten1po1:span
              --direction
10.20.246.15,po,ten1po1:tx
efa tenant service mirror session
create –name mirrorsession2 --
tenant ten1
              --source
10.20.246.15,po,ten1po1
              --type
10.20.246.15,po,ten1po1:flow-based
              --destination
10.20.246.15,po,ten1po1:10.20.246.1
5,eth,0/31
              --destination-type
10.20.246.15,po,ten1po1:span
              --direction
10.20.246.15,po,ten1po1:rx
```

```
efa tenant epg create –name
ten1epg2 –tenant ten1 --type port-
profile
              --po ten1po2
              --pp-ipv6-acl-in ext-
ipv6-permit-any-mirror-acl
efa tenant service mirror session
create –name mirrorsession3 --
tenant ten1
               --source
10.20.246.21,po,ten1po2
               --type
10.20.246.21,po,ten1po2:port-based
               --destination
10.20.246.21,po,ten1po2:10.20.246.2
1,eth,0/31
               --destination-type
10.20.246.21,po,ten1po2:span
               --direction
10.20.246.21,po,ten1po2:tx
efa tenant service mirror session
create –name mirrorsession4 --
tenant ten1
              --source
10.20.246.21,po,ten1po2
              --type
10.20.246.21,po,ten1po2:flow-based
              --destination
10.20.246.21,po,ten1po2:10.20.246.2
1,eth,0/31
              --destination-type
10.20.246.21,po,ten1po2:span
              --direction
10.20.246.21,po,ten1po2:rx
```

3. Verify the switch configuration on the SLX device.

10.20.246.15

```
SLX# show running-config ipv6
access-list
ipv6 access-list extended ext-ipv6-
permit-any-mirror-acl
 seq 10 permit ipv6 any any mirror
!
SLX#

SLX# show running-config int po 64
interface Port-channel 64
 mtu 9216
 description MCTPeerInterface
 ip address 10.20.20.3/31
 ipv6 access-group ext-ipv6-permit-
any-mirror-acl in
 no shutdown
!
SLX#
```

10.20.246.16

```
SLX# show running-config ipv6
access-list
ipv6 access-list extended ext-ipv6-
permit-any-mirror-acl
 seq 10 permit ipv6 any any mirror
!
SLX#

SLX# show running-config int po 64
interface Port-channel 64
 mtu 9216
 description MCTPeerInterface
 ip address 10.20.20.2/31
 ipv6 access-group ext-ipv6-permit-
any-mirror-acl in
 no shutdown
!
SLX#
```

10.20.246.21

```
SLX# show running-config ipv6
access-list
ipv6 access-list extended ext-ipv6-
permit-any-mirror-acl
 seq 10 permit ipv6 any any mirror
!
SLX#

SLX# show running-config int po 64
interface Port-channel 64
 mtu 9216
 description MCTPeerInterface
 ip address 10.20.20.3/31
 ipv6 access-group ext-ipv6-permit-
any-mirror-acl in
 no shutdown
!
SLX#
```

10.20.246.22

```
SLX# show running-config ipv6
access-list
ipv6 access-list extended ext-ipv6-
permit-any-mirror-acl
 seq 10 permit ipv6 any any mirror
!
SLX#

SLX# show running-config int po 64
interface Port-channel 64
 mtu 9216
 description MCTPeerInterface
 ip address 10.20.20.2/31
 ipv6 access-group ext-ipv6-permit-
any-mirror-acl in
 no shutdown
!
SLX#
```

10.20.246.15

```
SLX# show running-config monitor
session
monitor session 1
 source port-channel 64
destination ethernet 0/31
direction tx
!
monitor session 2
 source port-channel 64
destination ethernet 0/31
direction rx flow-based
!
SLX# show monitor session 1
Session                 : 1
Type                    : SPAN
Description             : [None]
State                   : Enabled
Source Interface        : Po 64 (Up)
Destination Interface   : Eth 0/31
(Down)
```

10.20.246.21

```
SLX# show running-config monitor
session
monitor session 1
 source port-channel 64
destination ethernet 0/31
direction tx
!
monitor session 2
 source port-channel 64
destination ethernet 0/31
direction rx flow-based
!
SLX# show monitor session 1
Session                 : 1
Type                    : SPAN
Description             : [None]
State                   : Enabled
Source Interface        : Po 64 (Up)
Destination Interface   : Eth 0/31
(Down)
```

Direction : Tx **Type : port-based** SLX# show monitor session 2 Session : 2 Type : SPAN Description : [None] State : Enabled Source Interface : Po 64 (Up) Destination Interface : **Eth 0/31 (Down)** Direction : Rx Type : **flow-based**	Direction : Tx **Type : port-based** SLX# show monitor session 2 Session : 2 Type : SPAN Description : [None] State : Enabled Source Interface : Po 64 (Up) **Destination Interface : Eth 0/31 (Down)** Direction : Rx **Type : flow-based**

Scalability

Scaled REST Request Timeout

When the scaled EFA tenant REST request (which takes more than 15 minutes) is executed, the REST request fails with an error `service is not available or internal server error has occurred, please try again later`.

Run the **`show`** command to verify the successful completion of failed REST request.

Scaled DRC Timeout

When a lot of tenant configuration is drifted in a scaled setup, the tenant configuration Drift and Reconciliation (DRC) will take more than 10 mins (timeout value). Therefore, the **`efa inventory drift-reconcile execute --ip <device-ip>`** command fails with the status `tenant-dr-timeout` as the time taken to reconcile the scale drifted tenant configuration is beyond the timeout (10 mins).

Run the **`efa inventory drift-reconcile execute --ip <device-ip>`** command for the same device once again to check the correct status of the DRC.

Policy Services Provisioning

Policy Services Provisioning Overview on page 230
Prefix List on page 230
Configure IP Prefix List on Devices on page 231
Route Map on page 233
Configure Route Map on devices on page 233
Event handling for IP prefix list and route map on page 237

Policy Services Provisioning Overview

Policy Service in EFA manages and configures IP prefix lists and route maps on fabric devices.

Note

Brownfield deployment does not support IP prefix lists and route maps.

Database, REST API and inter-service communication

Policy Service has its own database and provides REST APIs for its clients to configure and manage entities. It also registers with RabbitMQ to receive or publish messages.

Inventory Service interactions

Policy Service subscribes to the Inventory Service to receive events including device registration, device deletion, and changes to previously identified IP prefix lists and route maps.

During initialization or startup of the Policy Service, it fetches the essential entities, like device info, using REST APIs to populate its database.

Policy Service supports Drift and Reconcile (DRC) to receive and process the DRC events.

Prefix List

A prefix list allows routing systems to determine which routes must be accepted when they peer with their neighbors. A prefix list includes IP prefixes with a match criteria that allows or denies route redistribution. Prefix lists may contain one or more ordered entries which are processed sequentially.

EFA enables you to create, delete, and list IPv4 prefix list on a set of fabric devices. If you have not specified the ge or le value, then entry is matched with an exact prefix.

Configure IP Prefix List on Devices

1. Run the following command to configure the IPv4 prefix list:

```
efa policy prefix-list create ?
Flags:
      --type string          Type of prefix-list. Valid types are ipv4
      --name string          Name of Prefix list
      --rule stringArray     Rule in format seq[seq-num],action[permit/
deny],prefix[IPv4 prefix],ge[prefix-len],le[prefix-len]. Example:
seq[5],action[permit],prefix[10.0.0.0/8],ge[10],le[24]
```

Note

Use the ge and le keywords to specify the range of the prefix length to be matched. Exact match is assumed when neither ge nor le is specified.

Following is an example of IPv4 prefix list create:

```
efa policy prefix-list create --name prefix_v4 --type ipv4 --rule
seq[5],action[permit],prefix[10.0.0.0/8],ge[16]
```

2. Run the following command to configure or delete prefix-list configuration on devices:

   You can also use this command to add or remove rules.

```
efa policy prefix-list update ?
Flags:
      --type string      Type of prefix-list. Valid types are ipv4
      --name string      Name of Prefix list
      --operation string Valid options are add-rule, remove-rule, add-device, remove-
device
      --rule stringArray      Rule in format seq[seq-num],action[permit/
deny],prefix[IPv4 prefix],ge[prefix-len],le[prefix-len]. Example:
seq[5],action[permit],prefix[10.0.0.0/8],ge[10],le[24]
      --ip string        Comma separated range of device IP addresses. Example:
1.1.1.1-3,1.1.1.2,2.2.2.2
```

Note

- The add-device and remove-device operations configure or remove a prefix list rules on the specified devices.
- The add-rule and remove-rule operations add or remove a prefix list rule on the specified devices. If the prefix list is configured on the device, the rule is added or removed from the device.

Following is an example of IPv4 prefix list update:

Add device: This example configures prefix list on the devices.

```
efa policy prefix-list update --name prefix_v4 --type ipv4 --operation add-device --ip
10.20.246.10-11
```

Delete device: This example removes prefix list from the devices.

```
efa policy prefix-list update --name prefix_v4 --type ipv4 --operation remove-device --
ip 10.20.246.10-11
```

Add rule: This example adds rule to the already created prefix list.

```
efa policy prefix-list update --name prefix_v4 --type ipv4 --operation add-rule --rule
seq[5],action[permit],prefix[10.0.0.0/8],ge[16]
```

Delete rule: This example removes rule from the existing prefix list.

```
efa policy prefix-list update --name prefix_v4 --type ipv4 --operation remove-rule --
rule seq[5],action[permit],prefix[10.0.0.0/8],ge[16]
```

a. Verify the switch configuration on the SLX device.

```
SLX# show running-config ip prefix-list
ip prefix-list prefix_v4 seq 5 permit 10.0.0.0/8 ge 16
```

3. Run the following command to show the IPv4 prefix list on a list of devices:

```
efa policy prefix-list list ?
Flags:
       --type string      Type of prefix-list. Valid types are ipv4
       --ip string        Comma separated range of device IP addresses. Example:
1.1.1.1-3,1.1.1.2,2.2.2.2
```

Following is an example of IPv4 prefix list show:

IPv4 prefix list show

```
efa policy prefix-list list --type ipv4 --ip 10.20.246.10-11

Name: prefix_v4
+------+---------+--------+------------+----+----+--------------+-------------+
| Type | Seq num | Action |   Prefix   | Ge | Le | DeviceIP     | AppState    |
+------+---------+--------+------------+----+----+--------------+-------------+
| ipv4 |    5    | permit | 10.0.0.0/8 | 16 |    | 10.20.246.10 | cfg-in-sync |
+------+---------+--------+------------+----+----+--------------+-------------+
| ipv4 |    5    | permit | 10.0.0.0/8 | 16 |    | 10.20.246.11 | cfg-in-sync |
+------+---------+--------+------------+----+----+--------------+-------------+
```

4. Run the following command to delete the IPv4 prefix list on all devices:

This step deletes the prefix list on all devices and EFA.

```
efa policy prefix-list delete ?
Flags:
      --type string   Type of prefix-list. Valid types are ipv4
      --name string      Name of Prefix list
```

Following example deletes an IPv4 prefix list with name prefix_v4:

```
efa policy prefix-list delete --type ipv4 --name prefix_v4
```

Note

For more information about commands and supported parameters, see *Extreme Fabric Automation Command Reference, 2.7.0*.

Drift and Reconcile (DRC) and Idempotency for IP prefix list configuration

The following table captures the various attributes of IP prefix list for which DRC and idempotency is supported.

- A drift is identified if any of the fields are modified through SLX, CLI or other management tools.

- A reconcile operation pushes the intended configuration to SLX, so bringing the SLX configuration in sync with EFA.

Field	Identify Drift	Reconcile config	Idempotency	Comments
IPv4 prefix list rule is deleted.	Yes	Yes	No	Deleted rule will be reconciled.
IPv4 prefix list is deleted.	Yes	Yes	No	Deleted prefix list along with all rules associated with it will be reconciled.
IPv4 prefix list rule created OOB. Different rules exist with same prefix list name in EFA.	No	No	NA	
IPv4 prefix-list rule created OOB. Different rules exist with same prefix-list name and same sequence number in EFA.	Yes	Yes	NA	Prefix list rule will be reconciled to be in sync with EFA.
Create a IPv4 prefix list OOB with a prefix list name not matching any of EFA created entries.	No	No	NA	These are treated as out of band entries and EFA will not perform DRC.

Route Map

Route map is a route policy. It can use prefix list, access list, as-path, and community list to create an effective route policy. A route map consists of series of statements that check if a route matches the policy to permit or deny a route.

EFA 2.6.0 enables you to create, delete, and update the route maps on a set of devices in fabric. Note that the IPv4 prefix list is the ONLY supported match criterion, and no other match criteria is supported. Also, no set criteria is supported in this release.

Configure Route Map on devices

1. Run the following command to configure the route map with one or more rules:

```
efa policy route-map create ?
Flags:
      --name string           Name of route-map
      --rule stringArray      Rule in format seq[seq-num],action[permit/deny]
```

Following is an example of creating a route map rmap_1 with two rules:

```
efa policy route-map create --name rmap_1 --rule seq[5],action[permit] --rule
seq[10],action[permit]
```

2. Run the following command to update the route map configuration on a list of devices:

   The update command configures the route-map on device, removes configuration from a device or updates action of route-maps.

```
efa policy route-map update ?
Flags:
      --name string      Name of route-map
      --rule string      Rule in format seq[seq-num],action[permit/deny]
      --operation string Valid options are add-device, remove-device, update-action
      --ip string        Comma separated range of device IP addresses. Example:
1.1.1.1-3,1.1.1.2,2.2.2.2
```

   You can associate a route map with multiple rules.

   The add-device operation adds all the rules of the route map on the specified devices.

   The delete-device operation deletes all the rules of the route map on the specified devices.

   .

   Note
   The delete-device operation fails if the route map is bound to any BGP neighbor.

   For an update-action operation, specify the route map name and the rule. You can modify the action to permit or deny for a specific rule. You can provide only one rule at a time.

   Following are the examples of route map configuration update:

   a. Add device: Configures a route map rule on devices 10.20.246.10 and 10.20.246.11. Assume there are two route map rules for a map named rmap_1 that already exists in EFA:
      - rmap_1 seq 5 action permit
      - rmap_1 seq 10 action permit

```
efa policy route-map update --name rmap_1 --operation add-device --ip
10.20.246.10-11
```

   b. Delete device: Removes route map from the specified devices:

```
efa policy route-map update --name rmap_1 --operation delete-device --ip
10.20.246.10-11
```

   c. Update action: Changes the action from permit to deny for the specified rule

```
efa policy route-map update --name rmap_1 --rule seq[5],action[deny] --operation
update-action
```

   a. Verify the switch configuration on the SLX device.

Example 1

```
SLX# show running-config route-map
route-map rmap_1 permit 5
```

Example 2

```
SLX# show running-config route-map
route-map rmap_1 permit 5
route-map rmap_1 permit 10
```

Example 3

```
SLX# show running-config route-map
route-map rmap_1 deny 5
route-map rmap_1 permit 10
```

3. Run the following command to create route map match criteria:

   The IPv4 prefix list is the only match supported.

```
efa policy route-map-match create ?
Flags:
      --name string              Name of route-map
      --rule string              Rule in format seq[seq-num],action[permit/deny]
      --match-ipv4-prefix string IPv4 prefix-list name
```

   Following is an example of route map match create in IPv4:

```
efa policy route-map-match create --name rmap_1 --rule seq[5],action[permit] --match-
ipv4-prefix prefix_1
```

   a. Verify the switch configuration on the SLX device.

```
SLX# show running-config route-map
route-map rmap_1 permit 5
match ip address prefix-list prefix_1
```

4. Run the following command to remove the route map match criteria:

   The IPv4 prefix list is the only match supported.

```
efa policy route-map-match delete ?
Flags:
      --name string              Name of route-map
      --rule string              Rule in format seq[seq-num],action[permit/deny]
      --match-ipv4-prefix string IPv4 prefix-list name
```

   Following is an example of route map match delete in IPv4:

```
efa policy route-map-match delete --name rmap_1 --rule seq[5],action[permit]
```

   a. Verify the switch configuration on the SLX device.

```
SLX# show running-config route-map
route-map rmap_1 permit 5
```

5. Run the following command to display the route map for a list of devices:

   In the command output, the App State column reflects the state of configuration on the specified device. When there is drift in a rule, the App State is shown as `cfg-refreshed`.

```
efa policy route-map list ?
Flags:
      --ip string        Comma separated range of device IP addresses. Example:
1.1.1.1-3,1.1.1.2,2.2.2.2
```

   Example:

```
efa policy route-map list --ip 10.20.246.10-11

Route-map details:

Name: rmap_1
Seq: 5
Action: permit
Match-ipv4-prefixlist:
    Prefix-list: prefix_1

Name: rmap_2
Seq: 5
Action: permit
Match-ipv4-prefixlist:
    Prefix-list: prefix_1
```

```
IP Addresses:
+----------------+-----+---------------+-----------------+
|     Name       | Seq |  IP Address   |     App State   |
+----------------+-----+---------------+-----------------+
| rmap_1         |  5  | 10.20.246.10  |    cfg-in-sync  |
+----------------+-----+---------------+-----------------+
| rmap_1         |  5  | 10.20.246.11  |    cfg-in-sync  |
+----------------+-----+---------------+-----------------+
| rmap_2         |  5  | 10.20.246.10  |    cfg-in-sync  |
+----------------+-----+---------------+-----------------+
| rmap_2         |  5  | 10.20.246.11  |    cfg-in-sync  |
+----------------+-----+---------------+-----------------+
```

6. Run the following command to delete a route map and the associated rules on the devices:

```
efa policy route-map delete ?
Flags:
       --name stringArray      Name of route-map
       --seq string            Sequence numbers. For example 5,10,20, or all
```

The command removes the route map rule from the EFA database and from the associated devices.

You can delete a specific rule of a route map by specifying the route map name and the sequence number of the rule.

You can delete all the route map rules for a specific route map name by specifying the sequence number as "all".

The result of this command depends on whether the route map is bound with a BGP neighbor.

- If the route map is bound to BGP peer, you cannot delete the last route map rule.
- If the route map has no bindings, the command deletes the configuration on all devices associated with the route map.

Following example deletes two rules with sequence numbers 5 and 10 from a route map (rmap_1) that has three rules:

- rmap_1 seq 5 action permit
- rmap_1 seq 10 action permit
- rmap_1 seq 20 action permit

```
efa policy route-map delete --name rmap_1 --seq 5,10
```

a. Verify the switch configuration on the SLX device.

```
SLX1# show running-config route-map rmap_1
route-map rmap_1 permit 20
```

Drift and Reconcile (DRC) and Idempotency for route map configuration

The following table captures the various attributes of route map for which DRC and idempotency is supported.

- A drift is identified if any of the fields are modified through SLX, CLI, or other management tools.
- A reconcile operation pushes the intended configuration to SLX, so bringing the SLX configuration in sync with EFA.

Table 19: IP prefix list attributes supporting DRC and idempotency

Field	Identify Drift	Reconcile config	Idempotency	Comments
Route map deleted	Yes	Yes	No	Recreate the route map along with the match criteria during reconcile
Route map rule action updated	Yes	Yes	No	Reconcile the route map action (permit/deny) for that rule
Update IPv4 prefix list name in match criteria	Yes	Yes	No	Reconcile the IPv4 prefix list name
IPv4 prefix list match criteria deleted	Yes	Yes	NA	Reconcile the match criteria for IPv4 prefix list
A different match criteria NOT supported by EFA is added through OOB	No	No	NA	
A set criteria NOT supported by EFA is added through OOB	No	No	NA	
Route map is created through OOB and this is not present/created by EFA	No	No	NA	

Event handling for IP prefix list and route map

Inventory service

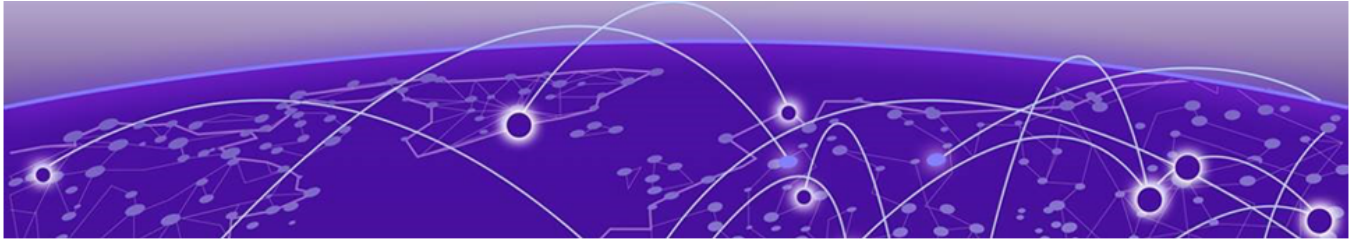
Inventory service maintains the IPv4 prefix list and route map information in its DB. As part of device update, diff is computed for these entities, and if there is a diff, an event is published.

Policy service

Policy service subscribes to the events from inventory service-related IP prefix list and route map and updates the app state of these entities. The entities for which the app state is in `cfg-refreshed` or `cfg-entry-deleted`, is reconciled as part of DRC.

To handle attribute level drift, DB maintains a bitmap to show exactly which attribute has drifted as part of DRC show output.

Policy service publishes events when you create, update or delete prefix lists and route maps.



EFA Device Management

Device Image Management on page 238
Device Health Management on page 260
Device Configuration Backup and Replay on page 261
Return Material Authorization on page 263
SLX Device Configuration on page 265

Device Image Management

Using maintenance mode, you can download firmware on one or more devices in the IP fabric with minimal disruption to data path traffic. Both Clos and small data center (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 verification
  - 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 the `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 with the `show` option, to display a table of devices in the fabric and their corresponding status

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.

- You can use this feature on a device where EFA TPVM is deployed, as long as you follow the instructions in *Extreme Fabric Automation Deployment Guide, 2.7.0*.
- 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
- SLX 9740

Hitless Firmware Upgrade

A hitless firmware upgrade uses the maintenance mode feature of the SLX device to gracefully divert traffic away from the device to alternate paths. The device can be put into maintenance mode and a firmware upgrade can be performed. The device can safely be rebooted and the new firmware activated without traffic loss. When the device is taken out of maintenance mode, traffic is allowed on the newly upgraded device.

Upgrading the Super-Spine Firmware in Clos

1. The firmware on the first super-spine is downloaded.
2. Enabling maintenance mode on a super-spine involves the Border Gateway Protocol (BGP). The `graceful_shutdown` parameter is sent to all the super-spine's underlay neighbors (all connected spines). Each neighbor processes the `graceful_shutdown` and refreshes their routes to use the alternate path. Maintenance mode is enabled on the first super-spine and traffic is diverted to the second super-spine.
3. The `running-configuration` is saved on the first super-spine to preserve all current configurations including the maintenance mode enable configuration.
4. The device is rebooted for firmware activation without traffic loss.
5. 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.
6. 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.

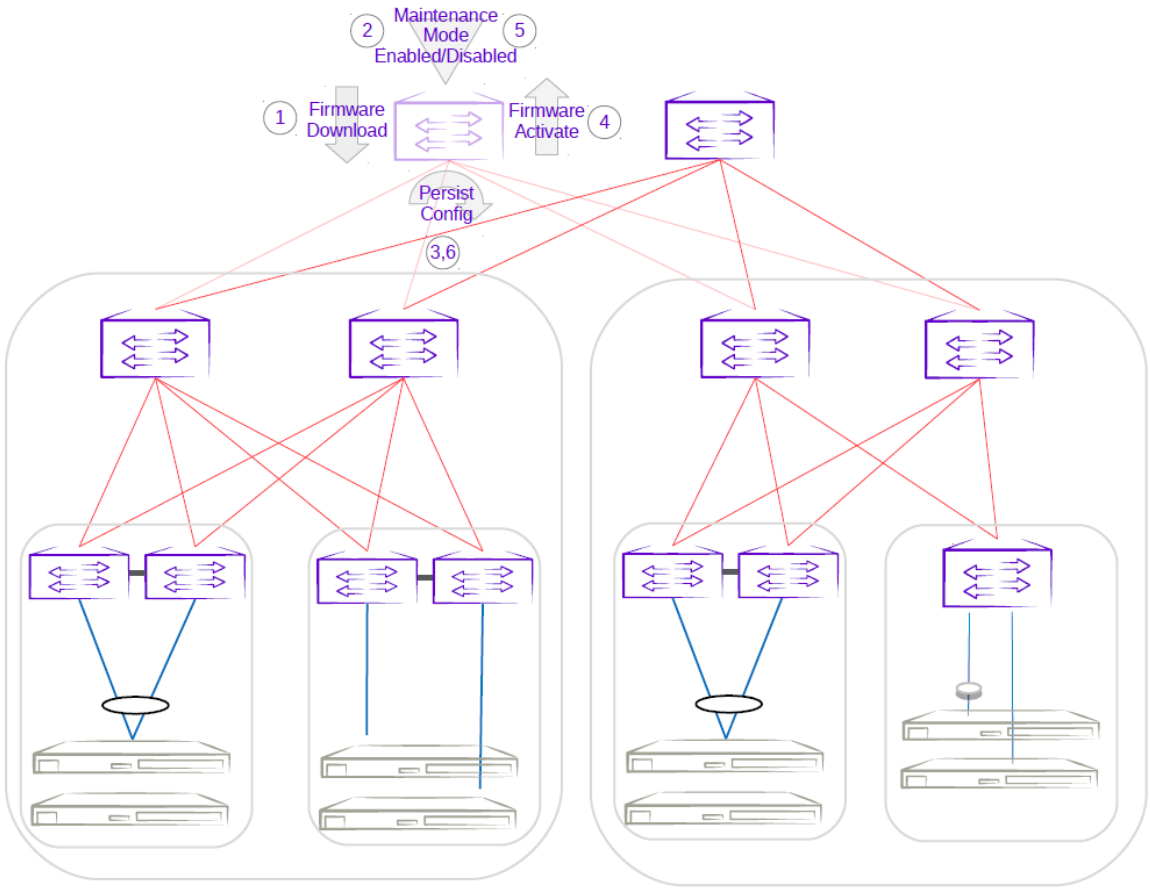


Figure 27: First super-spine firmware upgrade with maintenance mode

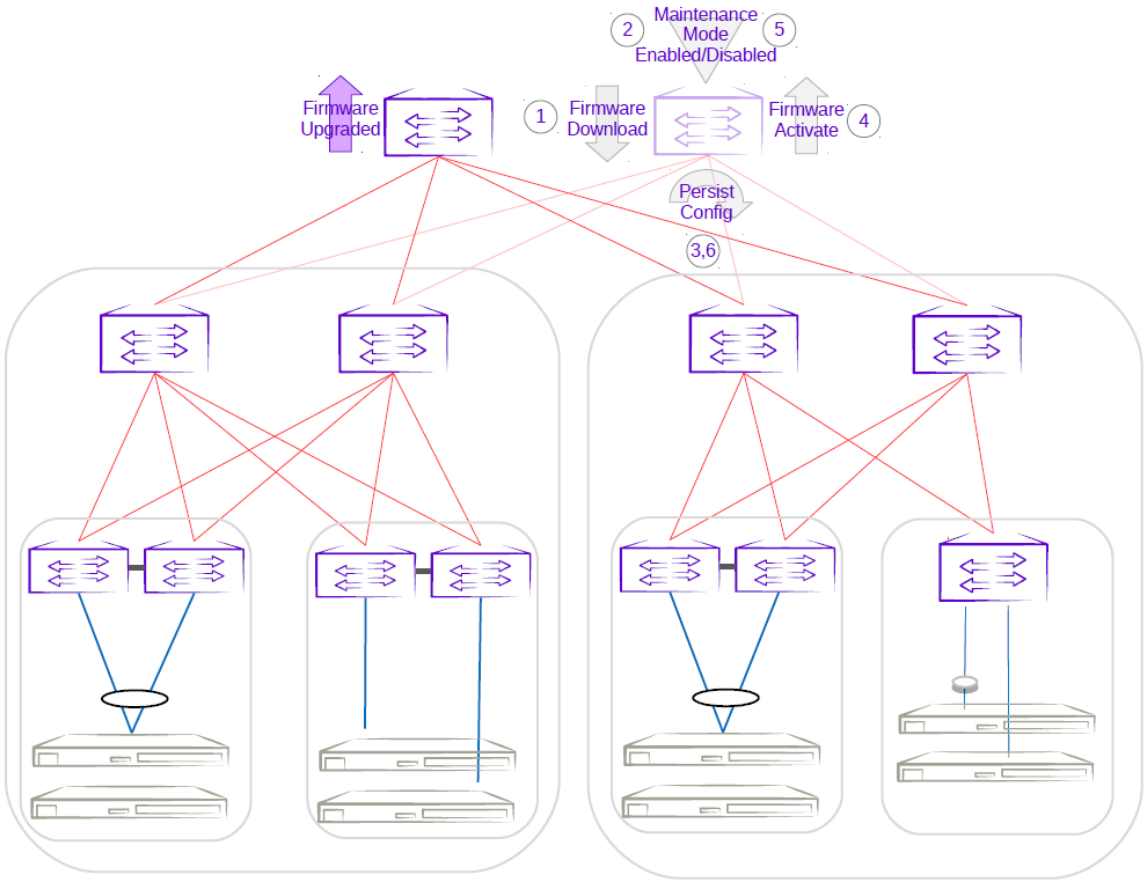


Figure 28: Second super-spine firmware upgrade with maintenance mode

Upgrading the Spine Firmware in Clos

1. The firmware on the first spine is downloaded.
2. Enabling maintenance mode on a spine also involves the Border Gateway Protocol (BGP). The `graceful_shutdown` parameter is sent to all the spine's underlay neighbors (all leafs in the pod and super-spines). The neighbors no longer send traffic to the first spine going into maintenance mode and redirect traffic to an alternate path.
3. The `running-configuration` is saved on the first spine to preserve all current configurations including the maintenance mode enable configuration.
4. The device is rebooted for firmware activation without traffic loss.
5. Once the new firmware is activated, maintenance mode is disabled to allow traffic again through the upgraded spine.
6. 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.

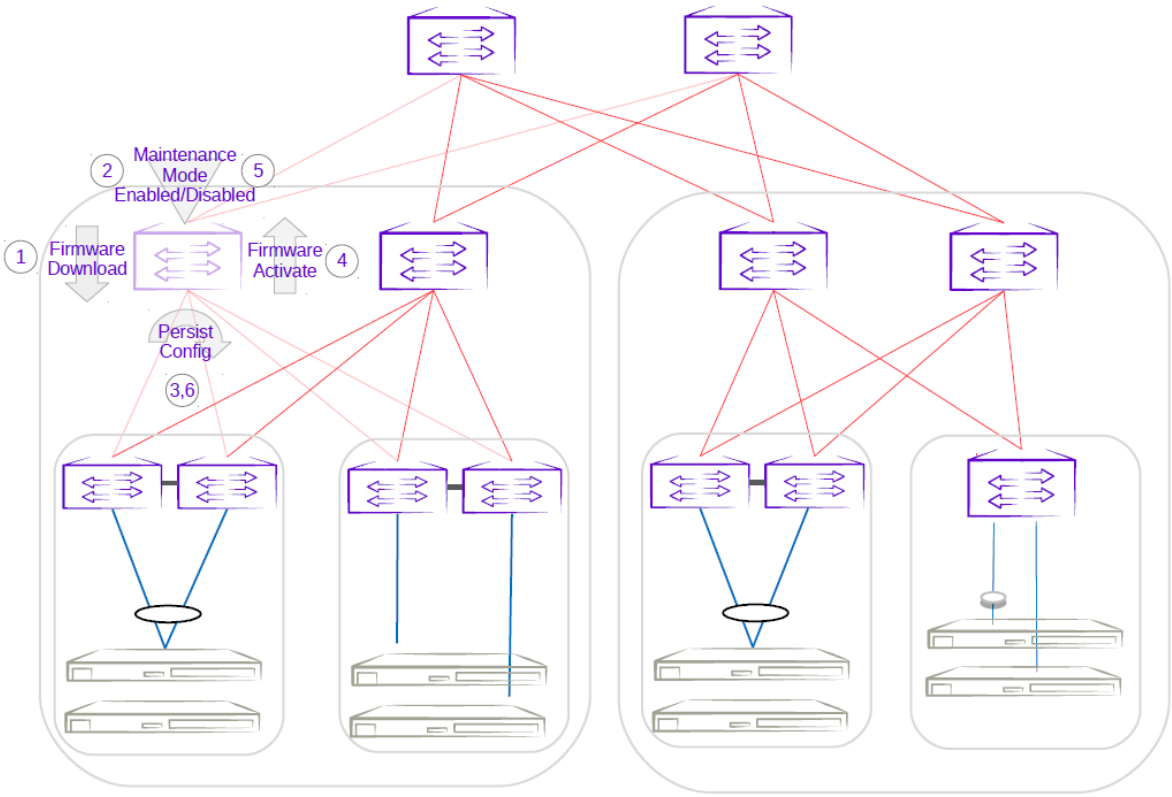


Figure 29: First spine firmware upgrade with maintenance mode

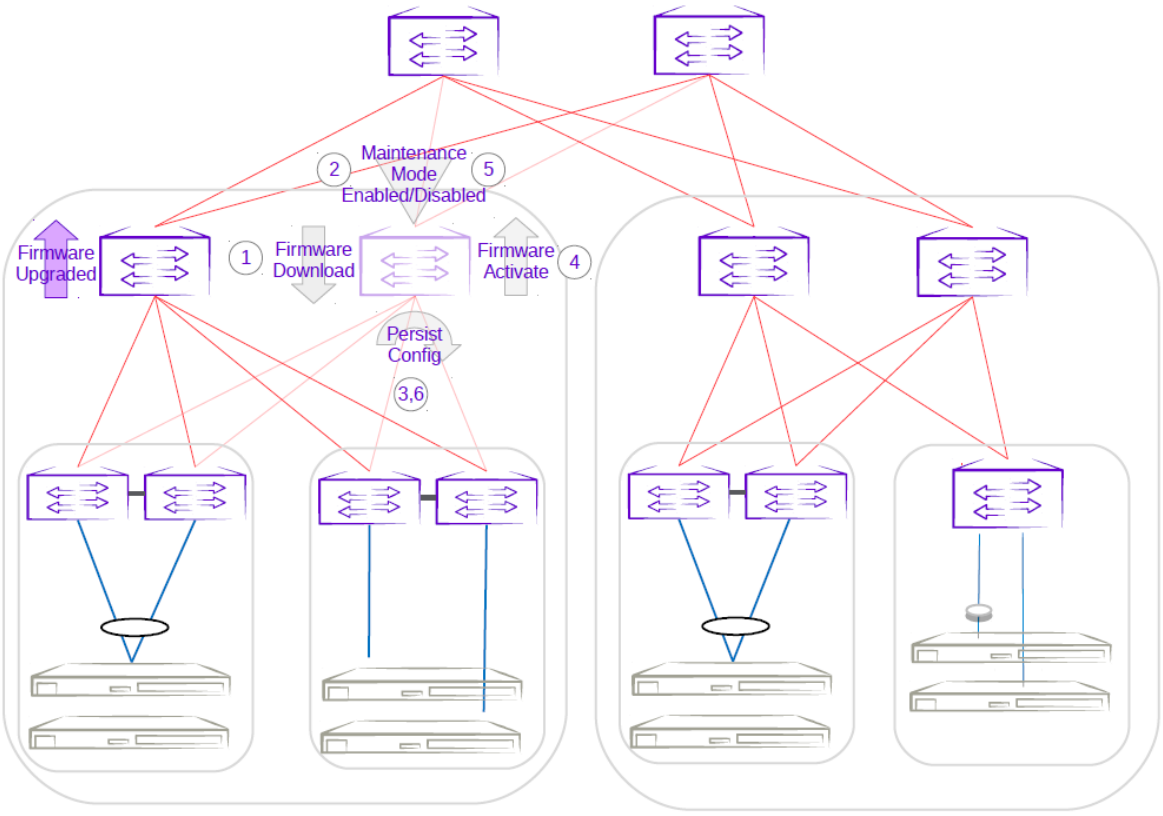


Figure 30: Second spine firmware upgrade with maintenance mode

Upgrading the Firmware of an MCT Leaf Pair with Dual-Homed Servers in Clos

1. The firmware on the MCT leaf is downloaded.
2. Enabling maintenance mode on an MCT leaf involves the Border Gateway Protocol (BGP) and MCT or NSM. The `graceful_shutdown` parameter is sent to all the leaf's underlay neighbors (all spines in the pod). The neighbors no longer send traffic to the MCT leaf going into maintenance mode and redirect traffic from spines to the peer MCT leaf. MCT instructs the peer leaf to become the designated forwarder, ICL is shut down, and CCE ports for clients are also shut down. Traffic from dual-homed servers is redirected to the peer leaf. With maintenance mode enabled, traffic is completely redirected to the peer leaf.
3. The `running-configuration` is saved on the first MCT leaf to preserve all current configurations including the maintenance mode enable configuration.
4. The device is rebooted for firmware activation without traffic loss.
5. After the firmware is upgraded, the maintenance mode is disabled to allow traffic again through the upgraded MCT leaf.
6. 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.

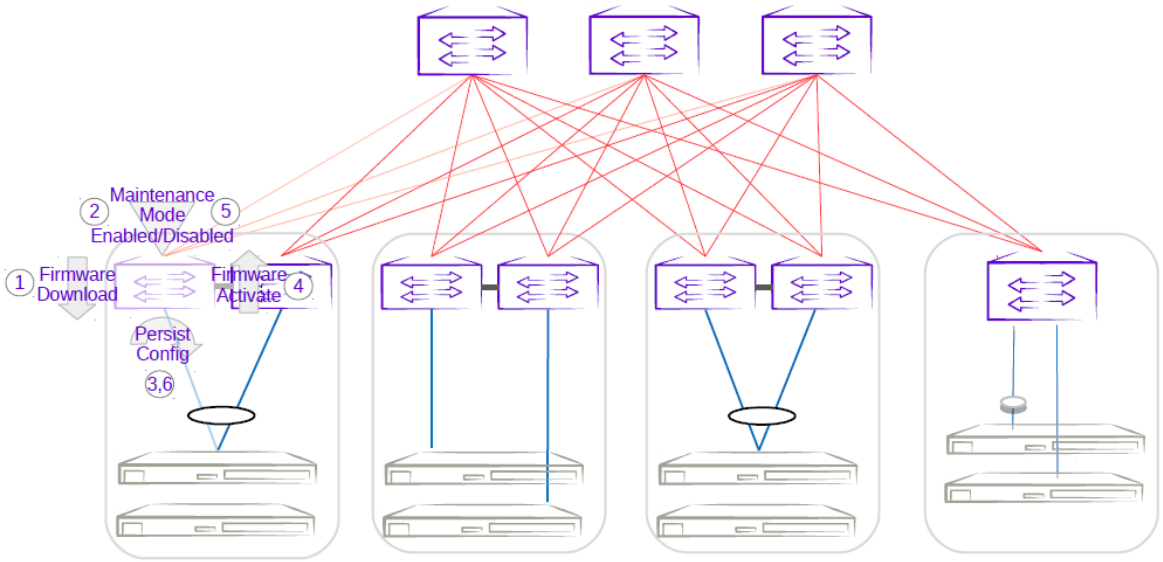


Figure 31: First MCT leaf firmware upgrade with maintenance mode

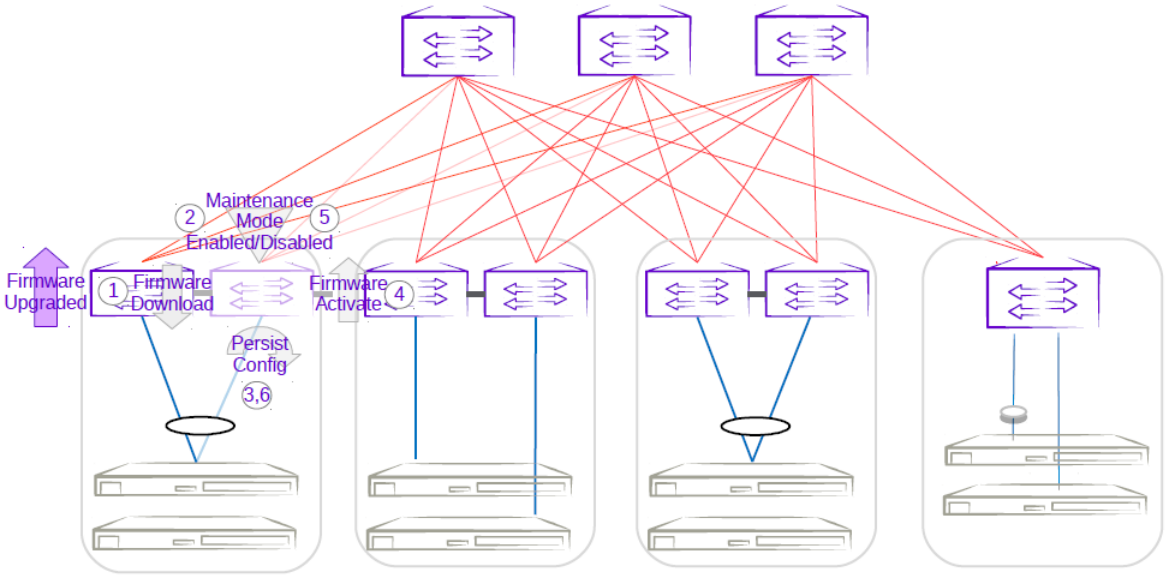


Figure 32: Second MCT leaf firmware upgrade with maintenance mode

Upgrading the Firmware of a Three-Rack Centralized MCT Pair in Small Data Center

1. The firmware on the MCT leaf is downloaded.
2. Enabling maintenance mode on one of the leafs in the centralized MCT leaf pair follows the same behavior as the MCT leaf pair in a Clos topology. The only difference is the iBGP Layer 3 backup link between MCT leaf pairs. Maintenance mode results in the traffic being redirected to the peer leaf in the centralized MCT leaf pairs.
3. The `running-configuration` is saved on the first MCT leaf to preserve all current configurations including the maintenance mode enable configuration.

4. The device is rebooted for firmware activation without traffic loss.
5. After the firmware is upgraded, the maintenance mode is disabled to allow traffic again through the upgraded MCT leaf.
6. 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.

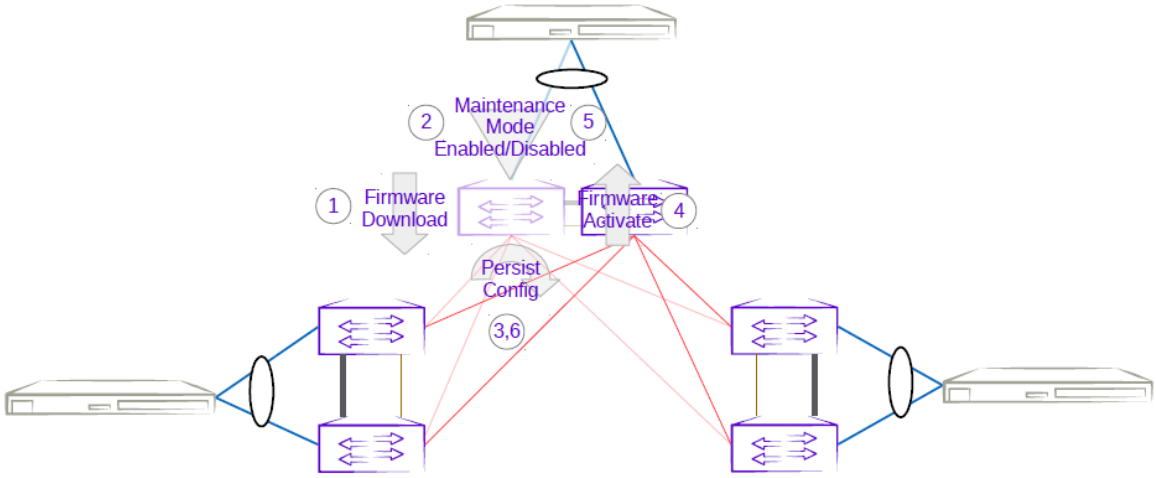


Figure 33: Three-rack centralized first MCT leaf firmware upgrade with maintenance mode

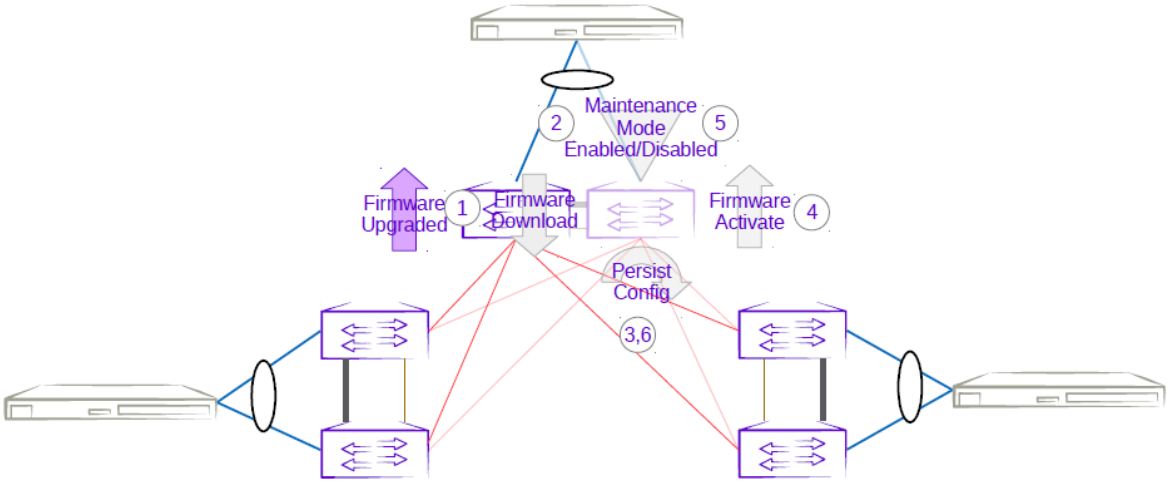


Figure 34: Three-rack centralized second MCT leaf firmware upgrade with maintenance mode

Upgrading the Firmware of a Three-Rack Ring MCT Pair in Small Data Center

1. The firmware on the MCT leaf is downloaded.
2. 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 a Clos topology. The only difference is the iBGP Layer 3 backup link between MCT leaf pairs. Maintenance mode results in the traffic being redirected to the peer MCT leaf.

3. The `running-configuration` is saved on the first MCT leaf to preserve all current configurations including the maintenance mode enable configuration.
4. The device is rebooted for firmware activation without traffic loss.
5. After the firmware is upgraded, the maintenance mode is disabled to allow traffic again through the upgraded MCT leaf.
6. 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.

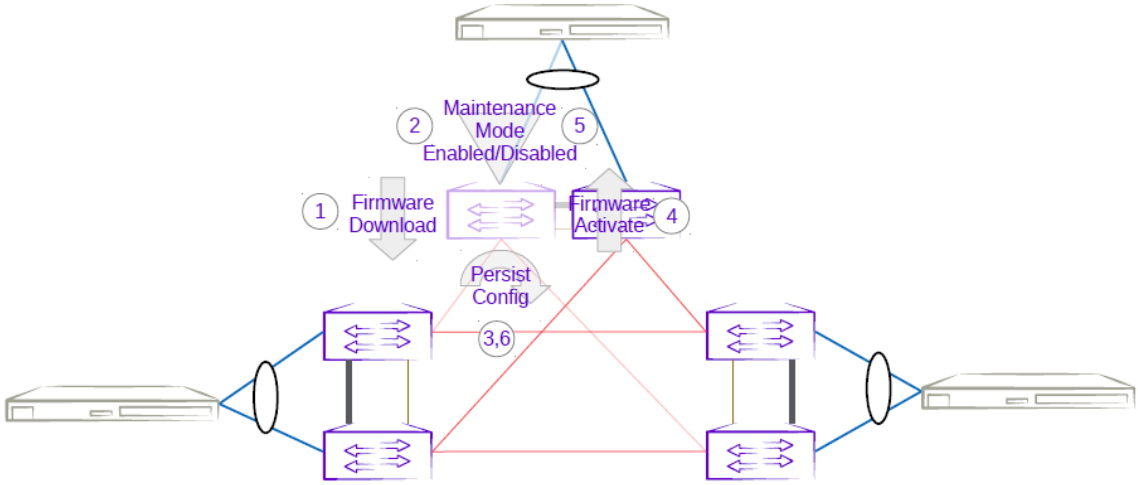


Figure 35: Three-rack ring first MCT leaf firmware upgrade with maintenance mode

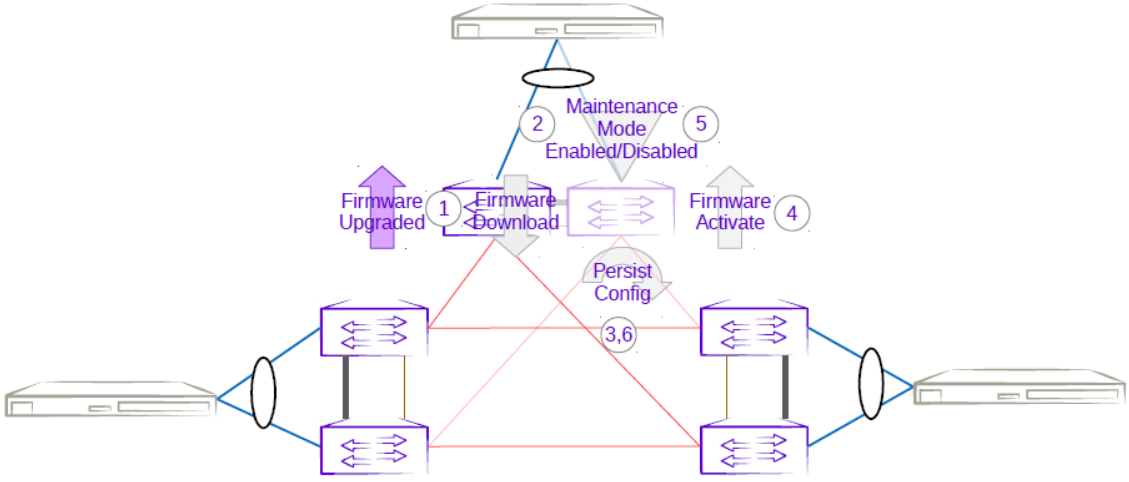


Figure 36: Three-rack ring second MCT leaf firmware upgrade with maintenance mode

Firmware Download

Use this topic to complete the firmware download and upgrade on fabric devices.

For more information about commands and supported parameters, see *Extreme Fabric Automation Command Reference, 2.7.0*

Firmware Upgrade with Minimal Traffic Loss

When updating firmware on a device, you typically begin by putting the device into maintenance mode so that traffic is diverted away from the device onto alternate paths. After performing the update, reboot the device, activate the new firmware, and take the device out of maintenance mode.

Alternatively, if it is not necessary to divert traffic away from the device, you can leave the device in active mode while updating the firmware. This enables the firmware to download faster.

Firmware Download Restart on HA Failover or Inventory Service Restart

Starting with EFA version 2.5.0, an in-progress firmware download restarts automatically if a high availability (HA) failover occurs, or if the inventory service restarts. This simplifies the process of preparing for the firmware download.

Firmware Download Implicit Fullinstall Support

The firmware download process automatically detects which devices require a firmware download fullinstall. You are warned that fullinstall is going to begin, because the fullinstall takes more time to complete. You do not need to provide any extra input outside of the normal prepare command.

EFA firmware download implicitly uses the “no reboot” option when issuing the firmware download command to the device. The SLX firmware download fullinstall command supports the “no reboot” option starting with SLXOS 20.2.3ea. This is the minimum required SLX version which should already be installed on the device for EFA firmware download to perform fullinstalls to a later SLX version on the device.

SLX firmware download does not support nocommit and fullinstall options specified together, so EFA firmware-download reports an error if there are devices requiring a fullinstall and the `--noAutoCommit` flag has been specified.

1. The **`firmware download prepare`** command reports a warning when EFA detects the device requires a full installation. The prepare is still successful.

   Example: [Supposing **`firmware download prepare`** detects a fullinstall is required for 10.20.246.4]

```
efa inventory device firmware-download prepare add --ip 10.20.246.4 --firmware-host
10.20.241.101 --firmware-directory /team/ztp/build/slx/slxos20.1.2e

+-------------+-----------+-------+--------------+-----+------
+---------------------------------+---------------+-----------------------------------
+-----------------+--------------------------------+
| IP Address  | Host Name | Model | Chassis Name | ASN | Role |        Current
Firmware        | Firmware Host |        Firmware Directory        | Target Firmware
|       Last Update Time        |
+-------------+-----------+-------+--------------+-----+------
+---------------------------------+---------------+-----------------------------------
+-----------------+--------------------------------+
| 10.20.246.4 | NH-Leaf2  | 3012  | SLX9250-32C  | 0   | Leaf |
20.2.3slxos20.2.3b_210309_0732 | 10.20.241.101 | /team/ztp/build/slx/slxos20.1.2e |
```

```
20.1.2e         | 2021-03-30 22:12:52 +0000 UTC |
+-------------+-----------+-------+--------------+-----+------
+---------------------------------+---------------+-----------------------------------
+-----------------+-------------------------------+
Firmware Download Prepare Add Details
Validate Firmware Download Prepare Add [success]

Warning(s)
    10.20.246.4: Full installation is required to firmware download 20.1.2e from
existing version. Firmware download will take longer to complete on device 10.20.246.4
```

2. The **firmware download execute** command reports an error when --noAutoCommit is specified and one or more devices require a full installation.

   Example: [Supposing **firmware download execute --noAutoCommit** is issued and fullinstall is required for 10.20.246.4]

```
efa inventory device firmware-download execute --fabric non_clos --noAutoCommit
Firmware Download Execute [failed]
    10.20.246.4: Device 10.20.246.4 cannot perform firmware download with noAutoCommit
and fullinstall requirement
```

EFA Command Blocking During Firmware Download

Before starting the firmware download, EFA verifies that all system services are not currently busy. If the verification does not complete within 2 minutes and 30 seconds, EFA displays an error. Retry the firmware download later.

Inventory Command Blocking

The following inventory commands are blocked when a firmware download is in progress for a specific device:

- Network Essentials (Native CLIs)
- Device Execute CLI
- Device Delete

Fabric Command Blocking

- The following fabric commands are blocked when at least one of the device is in `fwdl-in-progress` state:
  - **Fabric device add**

    Example: [Supposing firmware download is in progress for 10.20.246.1]

```
efa fabric device add --ip 10.20.246.2 --rack rack1 --name non-clos --username
admin --password password

Error:  Device(s) 10.20.246.1 are going through firmware download
```

  - **Fabric device remove**

    Example: [Supposing firmware download is in progress for 10.20.246.1]

```
efa fabric device remove --name non-clos --ip 10.20.246.1,10.20.246.2
         Remove Device(s) [Failed]
         Removal of  device with ip-address =  [Failed]

Reason: Device(s) 10.20.246.1 are going through firmware download
```

- **`Fabric configure`**

  Example: [Supposing firmware download is in progress for 10.20.246.1]

  ```
  efa fabric configure --name non-clos

  Error :  Device(s) 10.20.246.1 are going through firmware download
  ```

- The following fabric commands are allowed even when the devices are in `fwdl-in-progress` state:
  - Fabric device remove with “no-device-cleanup" flag
  - Fabric delete with “force” option
  - Fabric topology show physical/underlay/overlay to display the output by excluding the devices in fwdl-in-progress-state

Tenant Command Blocking

- If the target device list for a particular tenant operation has at least one device in `fwdl-in-progress` state, then the entire operation is rejected with an error to the user.

  Example: [Supposing firmware download is in progress for 10.24.80.158]

  ```
  efa tenant po create --name po1 --tenant t1 --speed 10Gbps --negotiation active --port
  10.24.80.158[0/3], 10.24.80.159[0/3]

  PortChannel creation failed:
        Error: Firmware download is in progress for the devices [10.24.80.158]
  ```

- If the target device list for a particular tenant operation has no device in `fwdl-in-progress` state, then the operation proceeds as usual.

  Example: [Supposing firmware download is in progress for 10.24.80.158]

  ```
  efa tenant po create --name po1 --tenant t1 --speed 10Gbps --negotiation active --port
  10.24.80.159[0/3]

  PortChannel creation succeeded
  ```

Failures During Group-based Firmware Download Execution

Failures can occur during the firmware download process such as network connectivity issues. Any such failures for a device during the group-based firmware download execution remain in the error state, and the execution proceeds to the next group by default. The **`firmware download execute`** command contains a new `--group-execution` parameter to control this behavior. The two policies are:

- continue-on-error (default): If any device results in an error during the firmware download process, firmware download execution will continue and process all the remaining groups. The overall status is failed.
- stop-on-error: If any device results in an error during the firmware download process, the firmware download execution will not proceed to the next group. Any devices in the remaining groups are left in a prepared state. The overall status is failed.

In the event of an error, you can restart the firmware download operation.. The execution automatically restarts on the firmware download process on failed device(s) per group

Group-based Firmware Download Restore

When the `--noAutoCommit` option is used with the firmware download execution, the device retains the previous firmware image. It enables you to test out the new firmware and decide to keep it by issuing a firmware download commit or go back to the previous firmware image by issuing a firmware download restore.

You can also apply the firmware download commit to all the devices in the fabric simultaneously. However, the firmware download restore implicitly reloads the device, so that the firmware download restore must be staged to prevent and minimize traffic loss.

You can invoke the firmware download restore for a fabric or for a set of IP addresses. In both the cases, the restore is applied in the same group order defined by the prepared list.

IP addresses based restore must be called for devices in the same fabric.

Group execution policy is inherited from what was specified in the firmware download execute CLI.

Fabric-wide Firmware Download

Use this procedure to upgrade the firmware of SLX devices in a Clos fabric.

This is the recommended method for upgrading the firmware of devices in a Clos fabric. It describes how to upgrade the device of standby EFA node and MCT leaf pairs, force a failover to change the active node to standby, and then upgrade the SLX of new standby node and remaining MCT leaf pairs.

To upgrade firmware in a small data center configuration, see *Extreme Fabric Automation Deployment Guide, 2.7.0*.

1. Prepare and run the firmware download on the devices in the fabric, in batches. In batch-1, add the device that hosts the standby node and devices on right side of the fabric. The diagram that follows illustrates the right and left devices in the batches of a fabric.

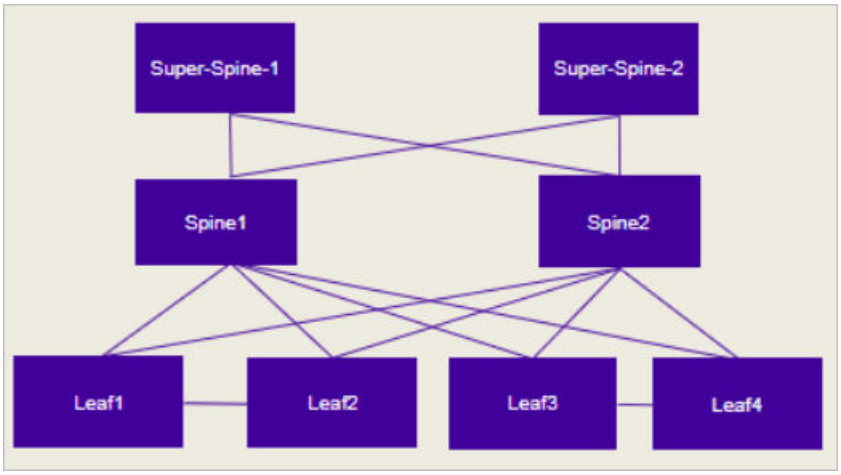


Figure 37: Batches for fabric-wide update

   a. Prepare the firmware download.

```
$ efa inventory device firmware-download prepare add --fabric <fabric name> --
firmware-host <IP of firmware download host>
--firmware-directory <path to target firmware build>
```

   The command returns the following information in a table: IP address, host name, model, chassis name, ASN, role, current firmware, firmware host, firmware directory, target firmware, and last update time.

   b. Download the firmware with or without the `–noAutoCommit`, `–noMaintMode`, and `–drc` options, as desired.

```
$ efa inventory device firmware-download execute
--fabric <fabric name> --noAutoCommit –noMaintMode

Firmware Download Execute [success]

--noAutoCommit    Configure Auto commit in Firmware Download
--noMaintMode     Configure Maintenance Mode in Firmware Download
--drc             Configure a drift reconcilation operation is performed after
Firmware Download
```

   c. Monitor the progress of the firmware download.

```
$ efa inventory device firmware-download show
--fabric <fabric name>
```

   d. Repeat step c until the firmware download is complete.

   Each time you repeat step c, the command returns a table that details the progress of the firmware download. The download is complete when the Update State column shows **Completed** and the Status column shows **Firmware Not Committed** when `–noAutoCommit` is used and **Firmware Committed** without `–noAutoCommit`.

2. Commit the firmware across all devices in the fabric.

```
$ efa inventory device firmware-download commit –fabric <fabric name>

OR

$ efa inventory device firmware-download commit –ip <IP address of all devices in
fabric>
```

The download is complete when the Update State column shows Completed on all devices and the Status column shows Firmware Committed.

Group-based Firmware Download Preparation

The firmware download prepare commands **add** and **delete** accept a new `--group <#>` parameter. Group creation and deletion is inferred by the existence of a prepared device in a group. These commands enable you to build and modify a custom prepared list for the entire fabric.

Fabric-based Firmware Download Preparation

The firmware download prepare **add** and **delete** command accept a new `--fabric <fabric name>` parameter. This parameter automatically generates a group-based prepared list for the entire fabric or delete any existing prepared list for the entire fabric.

You can review and further modify the auto-generated prepared list using the group-based prepare commands if required.

Clos Topology (3-Stage and 5-Stage)

The following rules for Clos topologies generate a fabric-based prepared list:

- The first group contains all non-MCT leaf devices and the MCT peer with the lower IP address from all MCT leaf and border-leaf devices.
- The second group contains the MCT peer with the higher IP address from all MCT leaf and border-leaf devices.
- The subsequent groups contains a single spine per group starting from the lowest to highest IP address of the spines.

The remaining groups contain a single super-spine per group starting from the lowest to highest IP address of the super-spines.

Small Data Center Topology (Centralized Rack and Rack Ring)

The following rules for small data center topologies generate a fabric-based prepared list:

- Due to potential loss of connectivity between racks (rack ring topology), all the devices are prepared for a serial upgrade (one device per group).
- The order is rack-by-rack, with the lower IP address peer followed by the higher IP address peer.

Group-based Firmware Download Execution

A single firmware download execution starts with the smallest group number performing the firmware download simultaneously on all devices prepared in the group and continue processing each group sequentially up through the largest group number. The group numbers need not be contiguously defined.

The prepared list remains until the entire group-based firmware download execution is completed and the firmware is committed or restored.

The following diagrams show an example of the group-based firmware download execution when prepared using the fabric-based auto-generated prepared list.

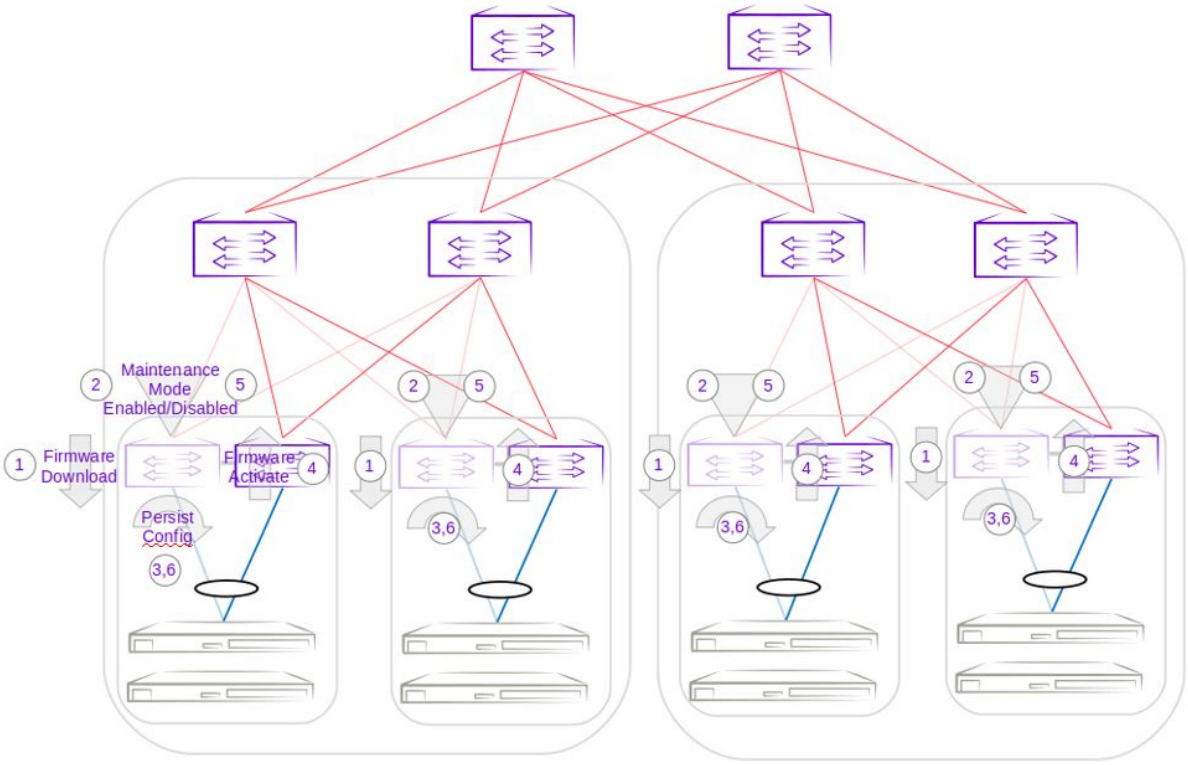


Figure 38: Group 1 - Firmware download execution of lower IP address MCT Peer leaf devices

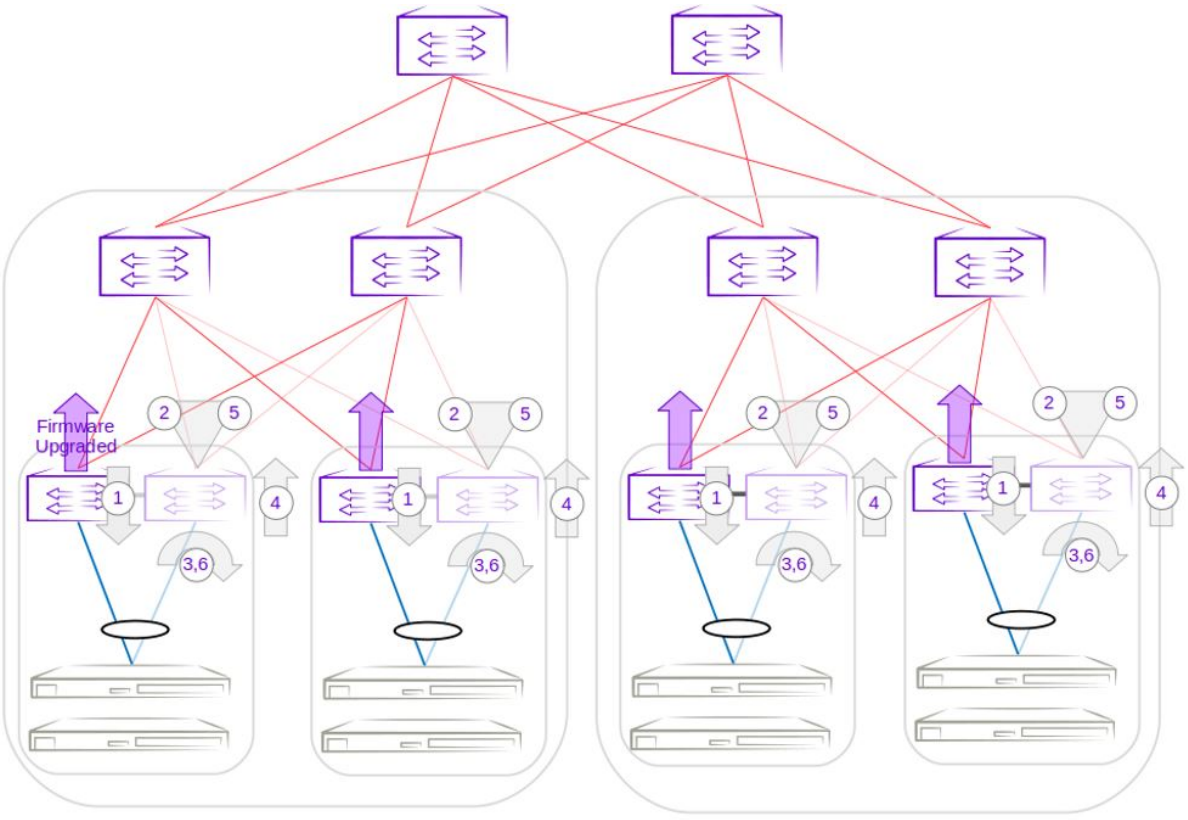


Figure 39: Group 2 - Firmware download execution of higher IP address MCT Peer leaf devices

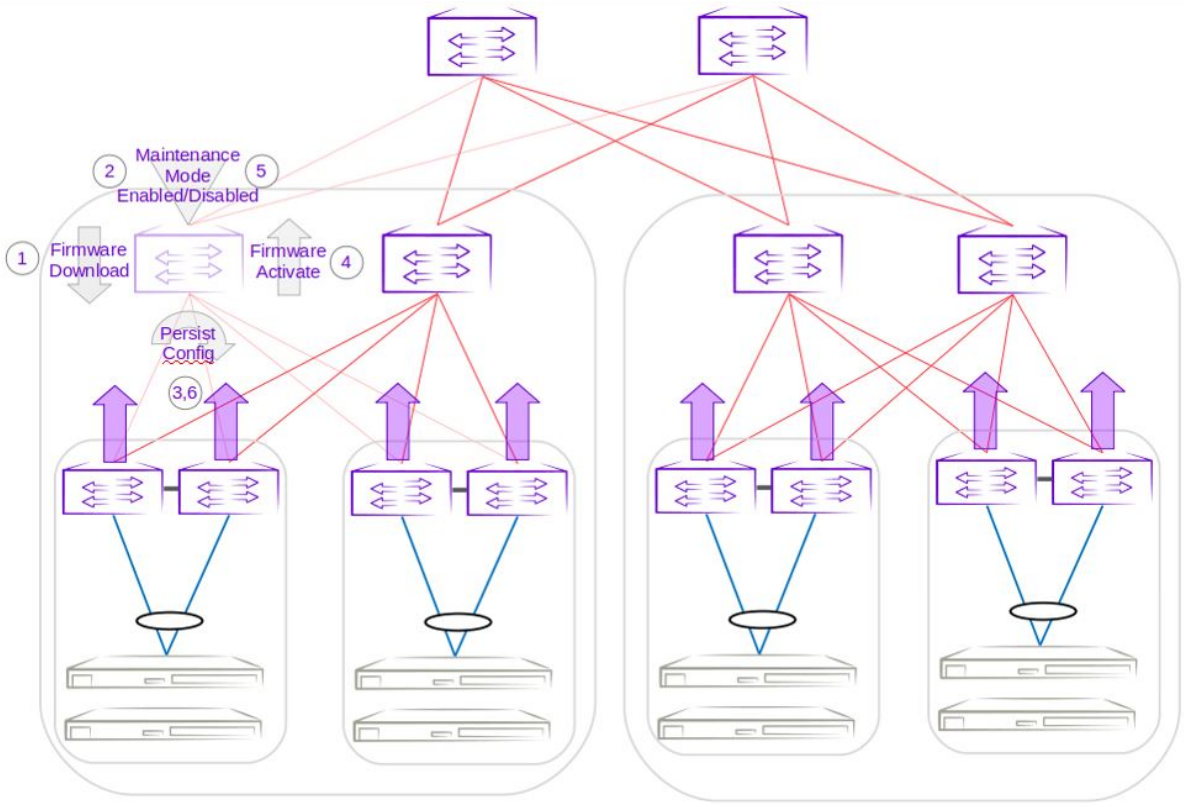


Figure 40: Group 3 - Firmware download execution of lowest IP address spine device

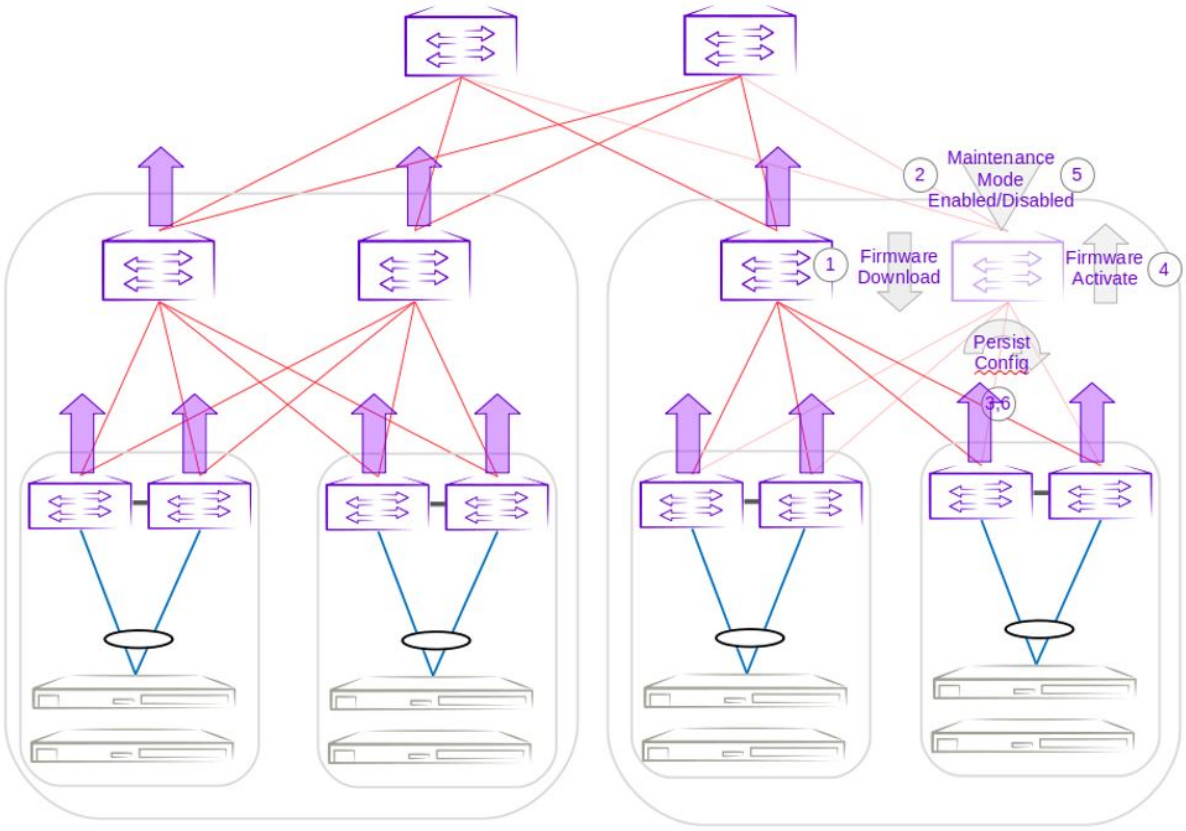


Figure 41: Group 6 - Firmware download execution of highest IP address spine device

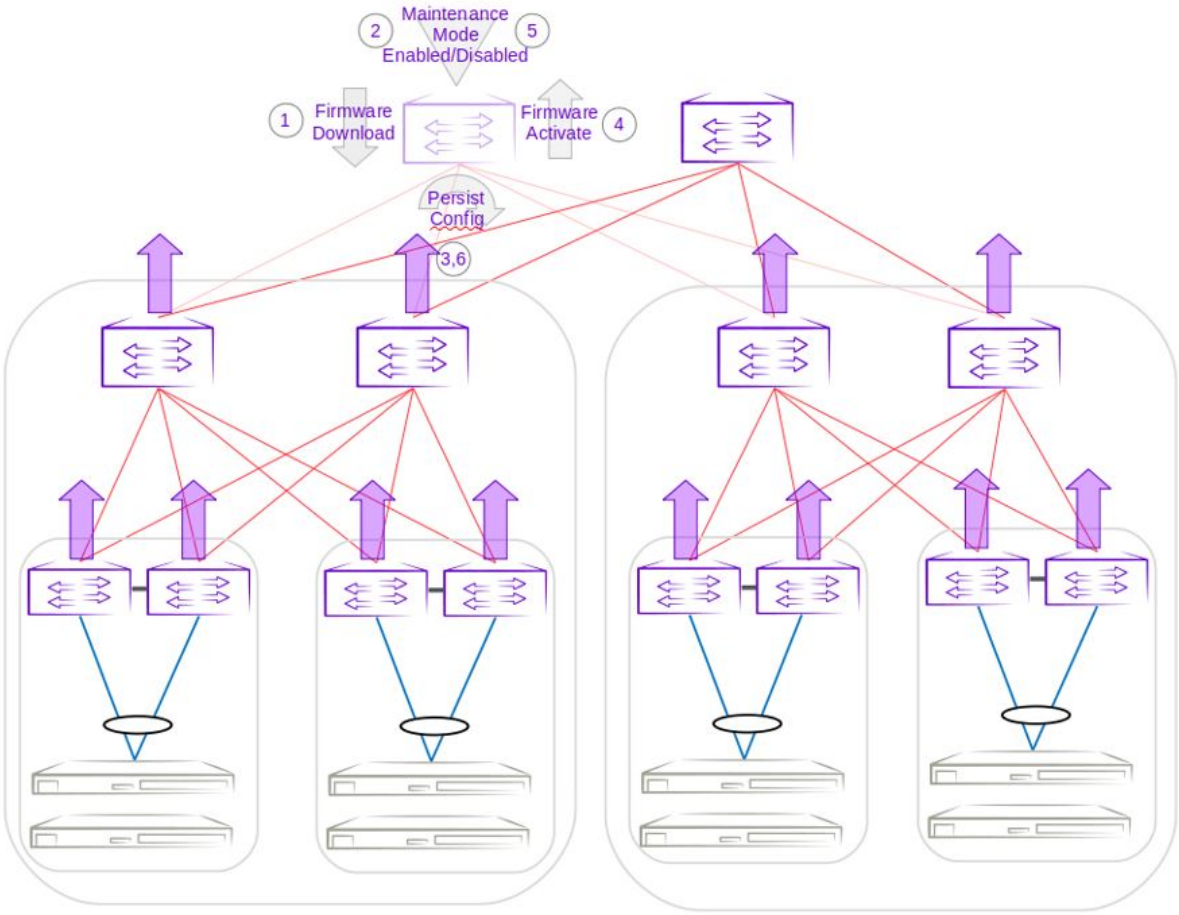


Figure 42: Group 7 - Firmware download execution of lowest IP address super-spine device

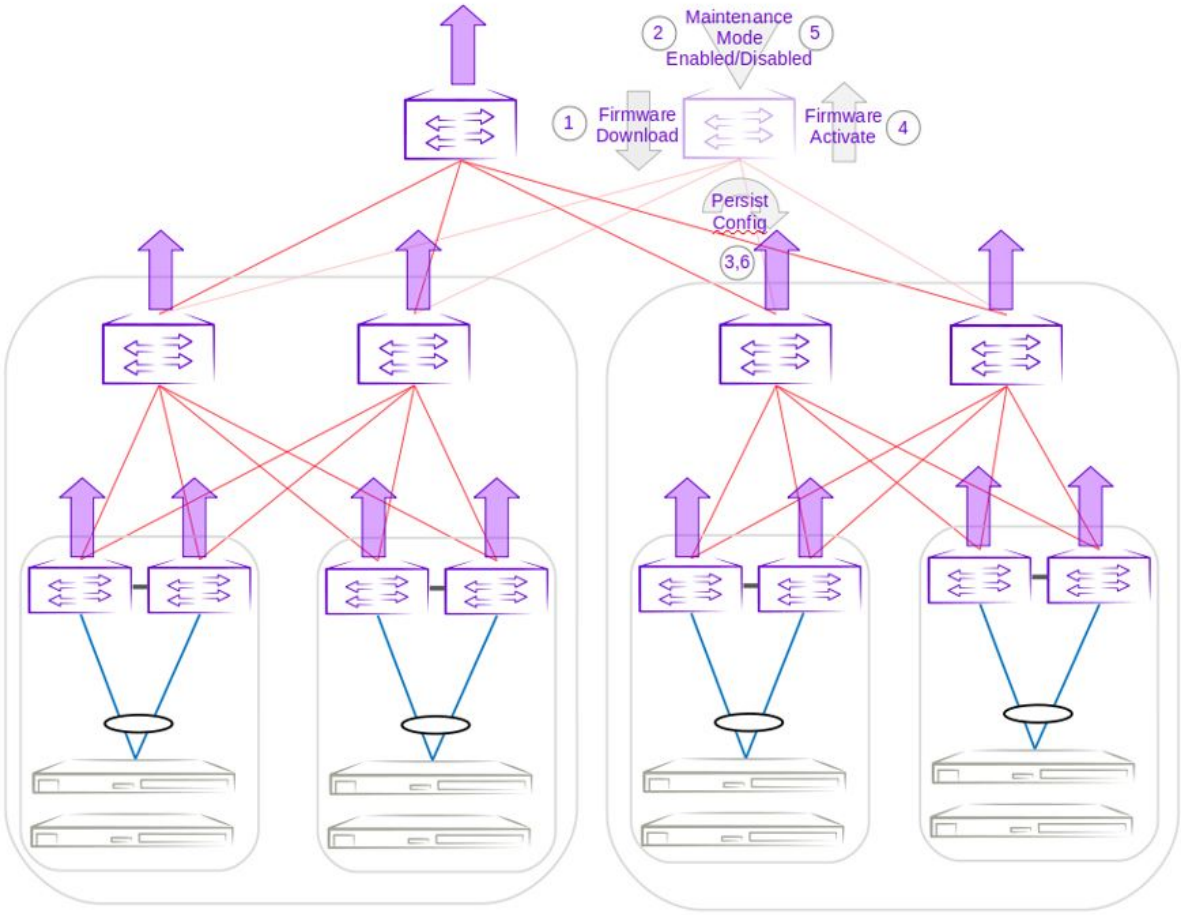


Figure 43: Group 8 - Firmware download execution of highest IP address super-spine device

Roll Back Device Firmware

Firmware on the device can be rolled back when it is in "Firmware Not Committed" status.

This is the recommended method for rolling back firmware when it is not committed. Run firmware restore on all devices in the fabric.

Run the following command to restore the firmware across all devices in the fabric:

```
$ efa inventory device firmware-download restore –fabric <fabric name>
OR
$ efa inventory device firmware-download restore –fabric <IP address of all devices in
fabric>
```

The download is complete when the Update State column shows Completed on all devices and the Status column shows Firmware Committed.

Traffic Loss Scenarios

Single Leaf

Traffic loss is expected when you upgrade a single leaf that is not in an MCT pair. Because there are no alternate paths for the single leaf, maintenance mode is not enabled. Only the configuration is persisted,

and a firmware upgrade is carried out. A traffic loss warning is flagged when you upgrade a single non-MCT leaf.

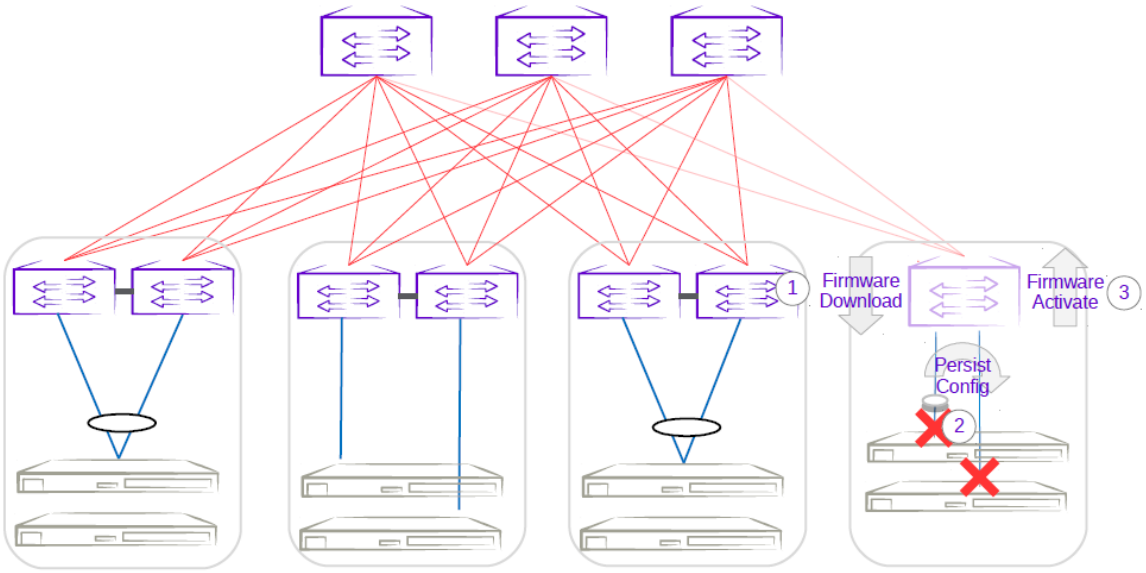


Figure 44: Single-leaf traffic loss

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 is provided at the start of a firmware download.

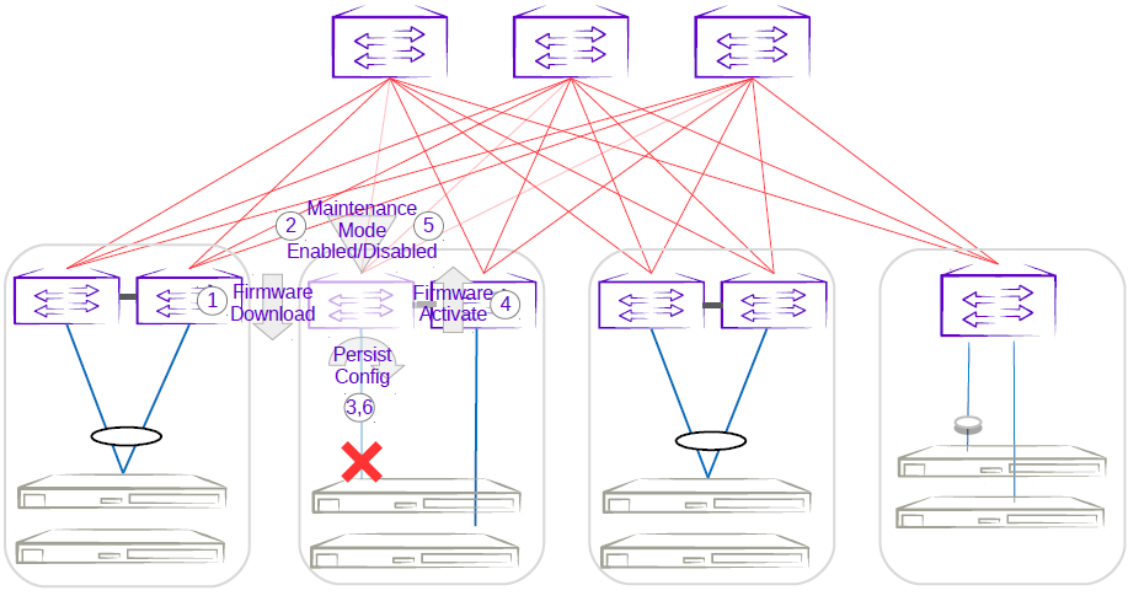


Figure 45: Single-homed server traffic loss

Non-Redundant Spine or Super-Spine

This is not a typical deployment, but traffic loss is expected in this scenario. Because no alternate paths exist for non-redundant devices, maintenance mode is not enabled for this case. A traffic loss warning is flagged when you upgrade non-redundant devices.

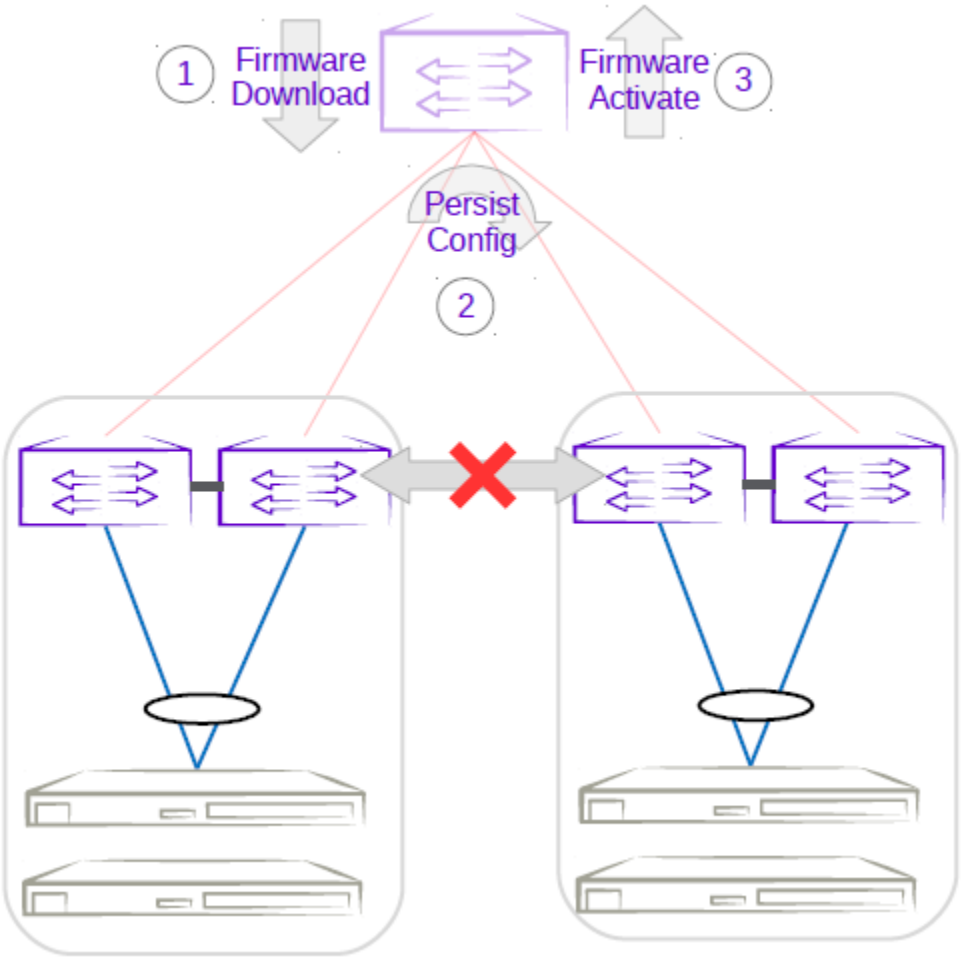


Figure 46: Non-redundant spine traffic loss

Device Health Management

Device Health Management (DHM) performs drift and reconciliation services, restoring fabric related configurations.

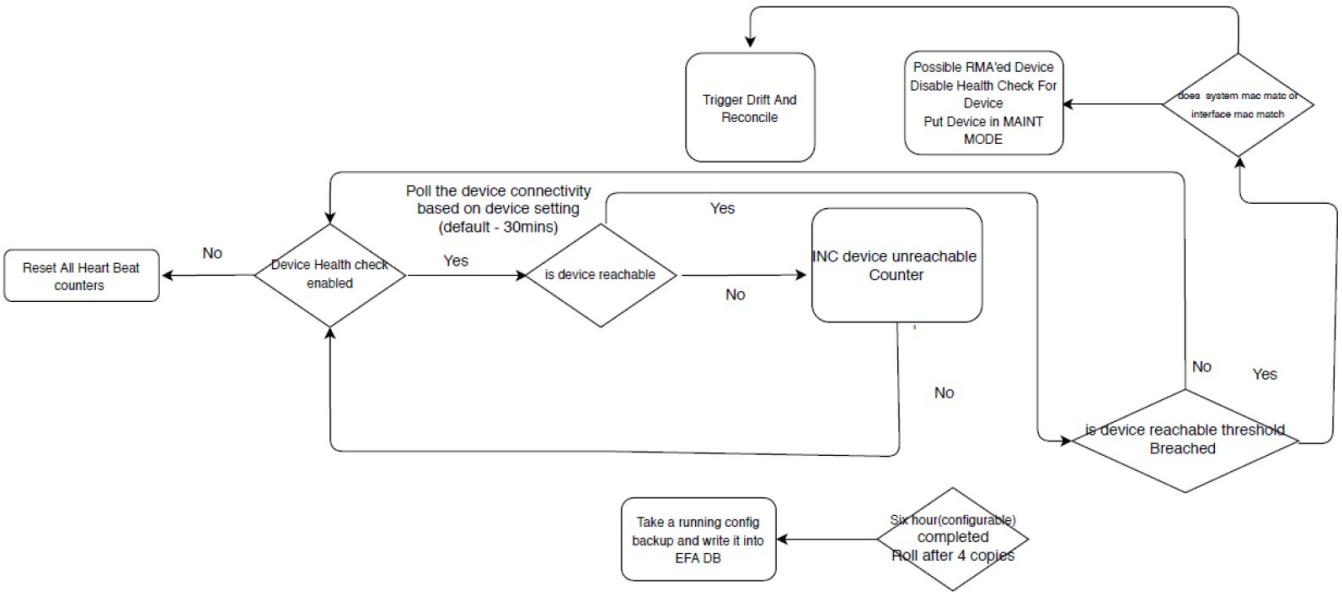


Figure 47: Device Health Management work flow

Monitor Device Health

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

1. Enable device 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 device health check.

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

Device Configuration Backup and Replay

The Device Configuration Backup and Replay feature enables backup of the device configuration based on inventory device settings or user-run commands and REST APIs.

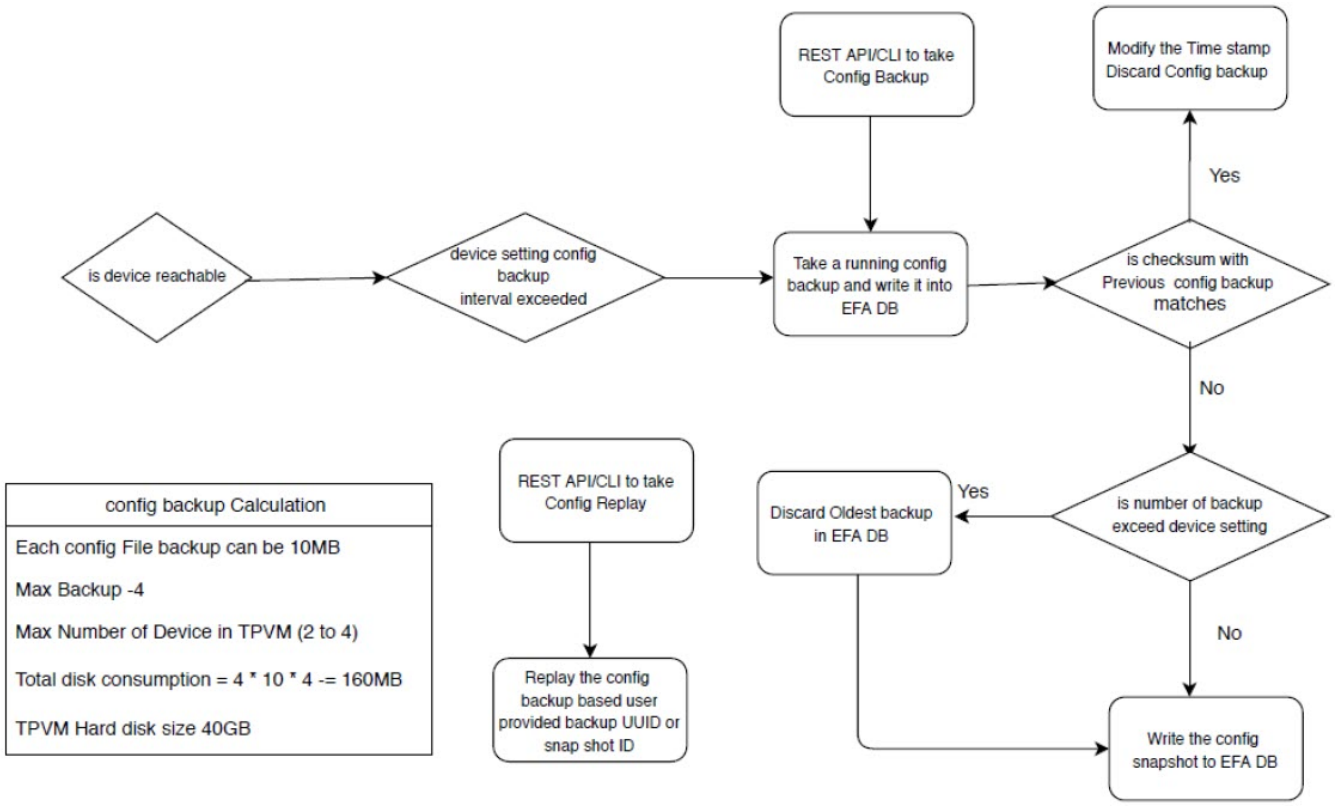


Figure 48: Workflow

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. (Optional) Delete config-backup.

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

5. Determine the backup restore method:
   - To restore backup using the `startup-config` file, proceed to the next step.
   - To restore backup using the `running-config` file, go to Step 7.

   Note
   The `startup-config` backup restore method requires device reboot to restore the configuration.

6. Configure device replay using the appropriate command.
   - Config-replay without rebooting the device:
     ```
     # efa inventory config-replay execute --ip 10.24.14.133 --uuid 1111-1111-111 --
     startup-config --no-reboot
     ```
   - Config-replay with device reboot:
     ```
     # efa inventory config-replay execute --ip 10.24.14.133 --uuid 1111-1111-111 --
     startup-config
     ```
7. Configure device replay using `running-config`.
   ```
   # efa inventory config-replay execute --ip 10.24.14.133 --uuid 1111-1111-111
   ```
8. View config-replay history.
   ```
   # efa inventory config-replay history --ip 10.24.14.133
   # efa inventory config-replay detail --uuid 1111-1111-1111
   ```
9. (Optional) Delete config-replay.
   ```
   # efa inventory config-replay delete --key 10.24.14.133
   # efa inventory config-replay delete --key 1111-1111-111
   ```

Return Material Authorization

With the Return Material Authorization (RMA) process, you can replace a faulty device with a new device that has the same configuration.

The high-level process is as follows. For specific steps and commands, see Replace a Faulty Device on page 264.

1. Verify prerequisites.
   - Periodic configuration backup must be enabled on all devices that may need RMA. This prerequisite ensures that you have the latest configuration file to be used for recovery.
   - Maintenance mode must be enabled upon reboot on all devices.
2. Remove the faulty device and replace it with the new device. Ports on this device must be administratively up and online.

   The ports on the new device must have the same connections as the old device. For example, if the old device Ethernet port1 and port2 went to port3 and port4 of another device, the new replacement device must have these exact same Ethernet port connections.
3. Configure the new device with the same management IP address and credentials as the old device.

4. Start the RMA process from the command line or with the REST API.

   Note
   As a best practice, run the **`efa inventory rma execute`** command with the configuration backup ID so that the configuration is properly restored. If you run the command without the backup ID, you must manually update the configuration on the new device.

   During the RMA process, the following actions occur:

   - The device boots up in maintenance mode.
   - EFA updates the device ID for the connection details in the database.
   - Maintenance mode is initiated if the device is not already in maintenance mode.
   - EFA replays the backed-up configuration specified by the `config-backup-id` parameter of the **`efa inventory rma execute`** command.
   - EFA begins the drift reconcile process, which involves device discovery, device update, and fabric and tenant reconciliation. For more information, see Drift and Reconcile on page 39.

     Note
     If the RMA command fails during this stage, you can manually run the drift reconcile process from the CLI. If the RMA process fails for any other reason, restart the RMA process.

   - When drift reconcile is complete, the device is taken out of maintenance mode.
   - During the RMA process, EFA health checks are deactivated and RASlog does not trigger drift reconcile.

5. Install the HTTPS or OAuth2 certificate on the new device.

   Note
   The following conditions result in the RMA process failing:

   - If there is mismatch in device IP and credentials
   - If the device maintenance mode is not successful

   In both the conditions, reject the device replacement with the new device.

Replace a Faulty Device

You can use the EFA command line to replace a faulty device with a new device and maintain the configuration of the old device.

Prerequisites:

- Ensure that periodic configuration backup is enabled on all devices that may need RMA. This prerequisite ensures that you have the latest configuration file to be used for recovery.
- Ensure that maintenance mode is enabled upon reboot on all devices.

This procedure describes how to replace a faulty device as part of the Return Material Authorization (RMA) process. For more information, see Return Material Authorization on page 263.

1. Obtain the configuration backup of the old device.

   For more information, see Configure Backup and Replay on page 262.

   ```
   # efa inventory config-backup execute --ip <ip-addr>
   ```

   This step generates a configuration backup ID that you use in step 5.
2. Replace the faulty device with the new device.
3. Ensure that the ports of the new device are administratively up and online.

   The ports on the new device must have the same connections as the old device. For example, if the old device Ethernet port1 and port2 went to port3 and port4 of another device, the new replacement device must have these exact same Ethernet port connections.
4. Configure the new device with the same management IP address and credentials as the old device.
5. Start the RMA process.

   ```
   # efa inventory rma execute --ip <ip-addr> --config-backup-id <id>
   ```

   Note

   As a best practice, run the command with the `--config-backup-id <id>` option so that the configuration is properly restored. If you run the command without the backup ID, you must manually update the configuration on the new device.
6. View the RMA history and detail.

   ```
   # efa inventory rma history --ip <ip-addr>
   # efa inventory rma detail -uuid <uuid>
   ```
7. View the drift reconcile history and detail.

   ```
   # efa inventory drift-reconcile history --device-ip <ip-addr>
   # efa inventory drift-reconcile detail --uuid <uuid>
   ```
8. Install the HTTPS or OAuth2 certificate on the new device.

   ```
   # efa certificates device install --ips <device-ip-addr> --certType [https|token]
   ```
9. (Optional) Delete the RMA record.

   ```
   # efa inventory rma delete --ip <ip-addr>
   ```
10. (Optional) Manually start the drift reconcile process if the RMA process fails during the drift reconcile stage.

    ```
    # efa inventory drift-reconcile execute --ip <ip-addr> --reconcile
    ```

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.

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 enter into maintenance mode after reboot, taking those devices out of maintenance mode after successfully reconciling the configuration on them. For more information about Drift and Reconcile, see Drift and Reconcile on page 39.

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.

If both `maint-mode-enable` and `maintenance-mode-enable-on-reboot` are set on the device, the Drift and Reconcile action is not triggered on device reboot.

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.

Tip

In SLX-OS, you can use the **`show interface ethernet`** command to see the current speed setting for the Ethernet interfaces on your device.

You can change the port speed for one or more IP addresses or for a specified fabric. For more configuration examples, see the *Extreme Fabric Automation Command Reference, 2.7.0*.

Note

The **`efa inventory device interface set-speed`** command is an operational (or exec) command, not a configuration command. With operational commands, there is no configuration persistence, no drift identification, and no configuration reconciliation. You run operational commands as needed.

Run the **`efa inventory device interface set-speed`** command.

This example sets the port speed on multiple IP addresses.

```
efa inventory device interface set-speed --ip 10.25.225.167,10.24.48.131,
10.24.51.135 --if-name 0/20-22  --speed 25gbps
```

```
+---------------+-----+------+----------------+------------+---------+--------+
|   DeviceIP    | ID  | Name | Interface Type | Port Speed | Result  | Reason |
+---------------+-----+------+----------------+------------+---------+--------+
| 10.25.225.167 | 9   | 0/21 | ethernet       | 25gbps     | Success |        |
+               +-----+------+----------------+------------+---------+--------+
|               | 89  | 0/20 | ethernet       | 25gbps     | Success |        |
+               +-----+------+----------------+------------+---------+--------+
|               | 1   | 0/22 | ethernet       | 25gbps     | Success |        |
+---------------+-----+------+----------------+------------+---------+--------+
| 10.24.51.135  | 16  | 0/21 | ethernet       | 25gbps     | Success |        |
+               +-----+------+----------------+------------+---------+--------+
|               | 86  | 0/20 | ethernet       | 25gbps     | Success |        |
+               +-----+------+----------------+------------+---------+--------+
|               | 48  | 0/22 | ethernet       | 25gbps     | Success |        |
+---------------+-----+------+----------------+------------+---------+--------+
| 10.24.48.131  | 142 | 0/20 | ethernet       | 25gbps     | Success |        |
+               +-----+------+----------------+------------+---------+--------+
|               | 110 | 0/21 | ethernet       | 25gbps     | Success |        |
+               +-----+------+----------------+------------+---------+--------+
|               | 148 | 0/22 | ethernet       | 25gbps     | Success |        |
+---------------+-----+------+----------------+------------+---------+--------+
Interface Details
--- Time Elapsed: 53.8631425s ---
```

This example sets the port speed for the specified fabric.

```
efa inventory device interface set-speed --fabric nc_no_vni
--if-name 0/20-22 --speed 25gbps
+--------------+-----+------+----------------+------------+---------+--------+
|   DeviceIP   | ID  | Name | Interface Type | Port Speed | Result  | Reason |
+--------------+-----+------+----------------+------------+---------+--------+
| 10.24.51.135 | 86  | 0/20 | ethernet       | 25gbps     | Success |        |
+              +-----+------+----------------+------------+---------+--------+
|              | 48  | 0/22 | ethernet       | 25gbps     | Success |        |
+              +-----+------+----------------+------------+---------+--------+
|              | 16  | 0/21 | ethernet       | 25gbps     | Success |        |
+--------------+-----+------+----------------+------------+---------+--------+
| 10.24.48.131 | 142 | 0/20 | ethernet       | 25gbps     | Success |        |
+              +-----+------+----------------+------------+---------+--------+
|              | 110 | 0/21 | ethernet       | 25gbps     | Success |        |
+              +-----+------+----------------+------------+---------+--------+
|              | 148 | 0/22 | ethernet       | 25gbps     | Success |        |
+--------------+-----+------+----------------+------------+---------+--------+
Interface Details
--- Time Elapsed: 36.9974805s ---
```

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 determine 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.

You can configure breakout for one or more IP addresses or for a specified fabric. For more configuration examples, see the *Extreme Fabric Automation Command Reference, 2.7.0*.

Note

The **efa inventory device interface set-breakout** command is an operational (or exec) command, not a configuration command. With operational commands, there is no configuration persistence, no drift identification, and no configuration reconciliation. You run operational commands as needed.

1. To break a port into multiple ports, run the **efa inventory device interface set-breakout** command.

   This example breaks three interfaces into four ports each.

```
efa inventory device interface set-breakout --ip 10.24.80.158
+--------------+----+-------+----------------+---------+--------+
|   DeviceIP   | ID | Name  | Interface Type | Result  | Reason |
+--------------+----+-------+----------------+---------+--------+
| 10.24.80.158 | 73 | 0/2:2 | ethernet       | Success |        |
+              +----+-------+----------------+---------+--------+
|              | 72 | 0/1:4 | ethernet       | Success |        |
+              +----+-------+----------------+---------+--------+
|              | 74 | 0/3:2 | ethernet       | Success |        |
+              +----+-------+----------------+---------+--------+
|              | 78 | 0/3:3 | ethernet       | Success |        |
+              +----+-------+----------------+---------+--------+
|              | 75 | 0/3:4 | ethernet       | Success |        |
+              +----+-------+----------------+---------+--------+
|              | 70 | 0/1:1 | ethernet       | Success |        |
+              +----+-------+----------------+---------+--------+
|              | 71 | 0/1:3 | ethernet       | Success |        |
+              +----+-------+----------------+---------+--------+
|              | 80 | 0/2:1 | ethernet       | Success |        |
+              +----+-------+----------------+---------+--------+
|              | 79 | 0/1:2 | ethernet       | Success |        |
+              +----+-------+----------------+---------+--------+
|              | 76 | 0/2:3 | ethernet       | Success |        |
+              +----+-------+----------------+---------+--------+
|              | 69 | 0/3:1 | ethernet       | Success |        |
+              +----+-------+----------------+---------+--------+
|              | 77 | 0/2:4 | ethernet       | Success |        |
+--------------+----+-------+----------------+---------+--------+
Interface Details
--- Time Elapsed: 48.3801684s ---
```

   This example configures break out for the specified fabric.

```
efa inventory device interface set-breakout --if-name 0/19-20
--mode 4x25g --fabric fabric1
+--------------+-----+--------+----------------+---------+--------+
|   DeviceIP   | ID  |  Name  | Interface Type | Result  | Reason |
+--------------+-----+--------+----------------+---------+--------+
| 10.24.80.158 | 188 | 0/20:2 | ethernet       | Success |        |
+              +-----+--------+----------------+---------+--------+
|              | 184 | 0/20:3 | ethernet       | Success |        |
+              +-----+--------+----------------+---------+--------+
|              | 190 | 0/20:4 | ethernet       | Success |        |
+              +-----+--------+----------------+---------+--------+
|              | 191 | 0/19:1 | ethernet       | Success |        |
```

```
+               +-----+--------+----------------+---------+--------+
|               | 187 | 0/19:2 | ethernet       | Success |        |
+               +-----+--------+----------------+---------+--------+
|               | 189 | 0/19:3 | ethernet       | Success |        |
+               +-----+--------+----------------+---------+--------+
|               | 185 | 0/19:4 | ethernet       | Success |        |
+               +-----+--------+----------------+---------+--------+
|               | 186 | 0/20:1 | ethernet       | Success |        |
+---------------+-----+--------+----------------+---------+--------+
| 10.24.80.159 | 196 | 0/19:2 | ethernet       | Success |        |
+               +-----+--------+----------------+---------+--------+
|               | 194 | 0/19:3 | ethernet       | Success |        |
+               +-----+--------+----------------+---------+--------+
|               | 197 | 0/19:4 | ethernet       | Success |        |
+               +-----+--------+----------------+---------+--------+
|               | 199 | 0/20:1 | ethernet       | Success |        |
+               +-----+--------+----------------+---------+--------+
|               | 193 | 0/20:2 | ethernet       | Success |        |
+               +-----+--------+----------------+---------+--------+
|               | 195 | 0/20:3 | ethernet       | Success |        |
+               +-----+--------+----------------+---------+--------+
|               | 192 | 0/20:4 | ethernet       | Success |        |
+               +-----+--------+----------------+---------+--------+
|               | 198 | 0/19:1 | ethernet       | Success |        |
+---------------+-----+--------+----------------+---------+--------+
Interface Details
--- Time Elapsed: 1m6.2210288s ---
```

2. To revert the breakout of multiple ports to the original configuration, run the **`efa inventory device interface unset-breakout`** command.

   This example removes breakout mode on multiple devices.

```
efa inventory device interface unset-breakout
--ip 10.24.80.158,10.24.80.159 --if-name 0/9-12
+--------------+--------------+----------------+----------------+---------+
|   DeviceIP   | Interface ID | Interface Name | Interface Type | Result  |
+--------------+--------------+----------------+----------------+---------+
| 10.24.80.158 | 248          | 0/10           | ethernet       | Success |
+              +--------------+----------------+----------------+---------+
|              | 250          | 0/11           | ethernet       | Success |
+              +--------------+----------------+----------------+---------+
|              | 249          | 0/12           | ethernet       | Success |
+              +--------------+----------------+----------------+---------+
|              | 247          | 0/9            | ethernet       | Success |
+--------------+--------------+----------------+----------------+---------+
| 10.24.80.159 | 252          | 0/10           | ethernet       | Success |
+              +--------------+----------------+----------------+---------+
|              | 254          | 0/11           | ethernet       | Success |
+              +--------------+----------------+----------------+---------+
|              | 253          | 0/12           | ethernet       | Success |
+              +--------------+----------------+----------------+---------+
|              | 251          | 0/9            | ethernet       | Success |
+--------------+--------------+----------------+----------------+---------+
Interface Details
--- Time Elapsed: 1m52.8562333s ---
```

   This example removes breakout mode for the specified fabric name.

```
efa inventory device interface unset-breakout --if-name 0/19-20
--fabric fabric1
+--------------+--------------+----------------+----------------+---------+
|   DeviceIP   | Interface ID | Interface Name | Interface Type | Result  |
+--------------+--------------+----------------+----------------+---------+
| 10.24.80.159 | 279          | 0/19           | ethernet       | Success |
```

```
+              +--------------+----------------+----------------+---------+
|              | 280          | 0/20           | ethernet       | Success |
+--------------+--------------+----------------+----------------+---------+
| 10.24.80.158 | 281          | 0/19           | ethernet       | Success |
+              +--------------+----------------+----------------+---------+
|              | 282          | 0/20           | ethernet       | Success |
+--------------+--------------+----------------+----------------+---------+
Interface Details
--- Time Elapsed: 1m19.226463s ---
```

Configure MTU at the Interface or System Level

You can configure the MTU (maximum transmission unit) at the system level or at the physical port level for Layer 2, IPv4, and IPv6.

Tip

In SLX-OS, you can use the **show interface ethernet** command to see the current MTU configuration for an interface.

You can change the MTU for one or more IP addresses or for a specified fabric. For more configuration examples, see the *Extreme Fabric Automation Command Reference, 2.7.0*.

Note

The **efa inventory device interface set-mtu** command is an operational (or exec) command, not a configuration command. With operational commands, there is no configuration persistence, no drift identification, and no configuration reconciliation. You run operational commands as needed.

At the interface level, run the **efa inventory device interface set-mtu** command.

This example configures the MTU on multiple IP addresses.

```
efa inventory device interface set-mtu --ip 10.25.225.167,10.24.48.131,
10.24.51.135 --if-name 0/20-22  --mtu 3600 --ip-mtu 3600 --ipv6-mtu 3600
+---------------+-----+------+----------------+------+--------+----------+---------
|   DeviceIP    | ID  | Name | Interface Type | MTU  | IP MTU | IPv6 MTU | Result
+---------------+-----+------+----------------+------+--------+----------+---------
| 10.25.225.167 | 9   | 0/21 | ethernet       | 3600 | 3600   | 3600     | Success
+               +-----+------+----------------+------+--------+----------+---------
|               | 89  | 0/20 | ethernet       | 3600 | 3600   | 3600     | Success
+               +-----+------+----------------+------+--------+----------+---------
|               | 1   | 0/22 | ethernet       | 3600 | 3600   | 3600     | Success
+---------------+-----+------+----------------+------+--------+----------+---------
| 10.24.48.131  | 142 | 0/20 | ethernet       | 3600 | 3600   | 3600     | Success
+               +-----+------+----------------+------+--------+----------+---------
|               | 148 | 0/22 | ethernet       | 3600 | 3600   | 3600     | Success
+               +-----+------+----------------+------+--------+----------+---------
|               | 110 | 0/21 | ethernet       | 3600 | 3600   | 3600     | Success
+---------------+-----+------+----------------+------+--------+----------+---------
| 10.24.51.135  | 16  | 0/21 | ethernet       | 3600 | 3600   | 3600     | Success
+               +-----+------+----------------+------+--------+----------+---------
|               | 48  | 0/22 | ethernet       | 3600 | 3600   | 3600     | Success
+               +-----+------+----------------+------+--------+----------+---------
|               | 86  | 0/20 | ethernet       | 3600 | 3600   | 3600     | Success
+---------------+-----+------+----------------+------+--------+----------+---------
Interface MTU Details
--- Time Elapsed: 59.5462548s ---
```

This example configures the MTU for the specified fabric name.

```
efa inventory device interface set-mtu --fabric nc_no_vni --if-name 0/20-22,0/55-58
--mtu 3200 --ip-mtu 3200 --ipv6-mtu 3200
+--------------+-----+---------+----------------+------+--------+----------+---------
+--------------------------
|   DeviceIP   | ID  |  Name   | Interface Type | MTU  | IP MTU | IPv6 MTU | Result
|       Reason
+--------------+-----+---------+----------------+------+--------+----------+---------
+--------------------------
| 10.24.51.135 | 16  | 0/21    | ethernet       | 3200 | 3200   | 3200     | Success
|
+              +-----+---------+----------------+------+--------+----------+---------
+--------------------------
|              | 48  | 0/22    | ethernet       | 3200 | 3200   | 3200     | Success
|
+              +-----+---------+----------------+------+--------+----------+---------
+--------------------------
|              | 86  | 0/20    | ethernet       | 3200 | 3200   | 3200     | Success
|
+              +-----+---------+----------------+------+--------+----------+---------
+--------------------------
|              | 0   | 0/55-58 | ethernet       | 0    | 0      | 0        | Failed  |
Interfaces [0/55-58]
|              |     |         |                |      |        |          |         | do
not exist for
                                                          device IP 10.24.51.135

Specify a valid interface
+--------------+-----+---------+----------------+------+--------+----------+---------
+--------------------------
| 10.24.48.131 | 0   | 0/55-58 | ethernet       | 0    | 0      | 0        | Failed  |
Interfaces [0/55-58]
|              |     |         |                |      |        |          |         |
does not exist for
                                                          device IP 10.24.51.131

Specify a valid interface
+              +-----+---------+----------------+------+--------+----------+---------
+--------------------------
|              | 110 | 0/21    | ethernet       | 3200 | 3200   | 3200     | Success
|
+              +-----+---------+----------------+------+--------+----------+---------
+--------------------------
|              | 148 | 0/22    | ethernet       | 3200 | 3200   | 3200     | Success
|
+              +-----+---------+----------------+------+--------+----------+---------
+--------------------------
|              | 142 | 0/20    | ethernet       | 3200 | 3200   | 3200     | Success
|
+--------------+-----+---------+----------------+------+--------+----------+---------
+--------------------------
Interface MTU Details
--- Time Elapsed: 37.3021602s ---
```

Change the Admin Status of an Interface

You can bring an interface administratively up or down.

Tip

In SLX-OS, you can use the **show interface ethernet** command to see the status of the Ethernet interfaces on your device.

You can change the Admin Status for one or more IP addresses or for a specified fabric. For more configuration examples, see the *Extreme Fabric Automation Command Reference, 2.7.0*.

Note

The **efa inventory device interface set-admin-state** command is an operational (or exec) command, not a configuration command. With operational commands, there is no configuration persistence, no drift identification, and no configuration reconciliation. You run operational commands as needed.

Run the **efa inventory device interface set-admin-state** command.

This example changes the Admin State on multiple IP addresses.

```
efa inventory device interface set-admin-state
--ip 10.25.225.167,10.24.48.131,10.24.51.135 --if-name 0/20-22 --state up

+---------------+-----+------+----------------+--------------+---------+--------+
|   DeviceIP    | ID  | Name | Interface Type | Admin Status | Result  | Reason |
+---------------+-----+------+----------------+--------------+---------+--------+
| 10.24.48.131  | 110 | 0/21 | ethernet       | up           | Success |        |
+               +-----+------+----------------+--------------+---------+--------+
|               | 142 | 0/20 | ethernet       | up           | Success |        |
+               +-----+------+----------------+--------------+---------+--------+
|               | 148 | 0/22 | ethernet       | up           | Success |        |
+---------------+-----+------+----------------+--------------+---------+--------+
| 10.24.51.135  | 16  | 0/21 | ethernet       | up           | Success |        |
+               +-----+------+----------------+--------------+---------+--------+
|               | 48  | 0/22 | ethernet       | up           | Success |        |
+               +-----+------+----------------+--------------+---------+--------+
|               | 86  | 0/20 | ethernet       | up           | Success |        |
+---------------+-----+------+----------------+--------------+---------+--------+
| 10.25.225.167 | 9   | 0/21 | ethernet       | up           | Success |        |
+               +-----+------+----------------+--------------+---------+--------+
|               | 89  | 0/20 | ethernet       | up           | Success |        |
+               +-----+------+----------------+--------------+---------+--------+
|               | 1   | 0/22 | ethernet       | up           | Success |        |
+---------------+-----+------+----------------+--------------+---------+--------+
Interface Details
--- Time Elapsed: 56.4964544s ---
```

This example changes the Admin Status for the specified fabric.

```
efa inventory device interface set-admin-state --fabric nc_no_vni
--if-name 0/20-22 --state up
+--------------+-----+------+----------------+--------------+---------+--------+
|   DeviceIP   | ID  | Name | Interface Type | Admin Status | Result  | Reason |
+--------------+-----+------+----------------+--------------+---------+--------+
| 10.24.51.135 | 48  | 0/22 | ethernet       | up           | Success |        |
+              +-----+------+----------------+--------------+---------+--------+
|              | 16  | 0/21 | ethernet       | up           | Success |        |
+              +-----+------+----------------+--------------+---------+--------+
|              | 86  | 0/20 | ethernet       | up           | Success |        |
+--------------+-----+------+----------------+--------------+---------+--------+
```

```
| 10.24.48.131 | 148 | 0/22 | ethernet       | up           | Success |         |
+              +-----+------+----------------+--------------+---------+---------+
|              | 142 | 0/20 | ethernet       | up           | Success |         |
+              +-----+------+----------------+--------------+---------+---------+
|              | 110 | 0/21 | ethernet       | up           | Success |         |
+--------------+-----+------+----------------+--------------+---------+---------+
Interface Details
--- Time Elapsed: 37.9236481s ---
```

Configure NTP at Device and Fabric Levels

Use the native commands to configure the NTP server on the SLX device. The configuration is persisted in the EFA database. DRC is supported.

1. Run the **efa inventory device ntp server create** command to create an NTP server.

```
efa inventory device ntp server create ?
Flags:
      --ip string        Comma separated range of device IP addresses.
      --ntp-ip string    NTP server IP address
      --auth-key int Authentication key ID. Values 1 to 65535
      --auth-key-name string Key name
      --encryption-type string Encryption type. Valid values are md5, sha1
      --trusted-key bool Trusted key.
      --fabric string    fabric name.
-h, --help             help for ntp
```

   Example:

```
efa inventory device ntp server create -ntp-ip 3.3.3.3 --auth-key 1 --auth-key-name
ntpsecret --encryption-type md5 -trusted-key  --ip 10.20.246.10

efa inventory device ntp server create -ntp-ip 3.3.3.3 --auth-key 1 --auth-key-name
ntpsecret --encryption-type md5 -trusted-key  --fabic clos_fabric
```

2. On the SLX device, verify the NTP configuration.

```
SLX# show running-config ntp
ntp authentication-key 1 md5 $9$750C7e0ayuI31YUga1Clmg== encryption-level 7
ntp authenticate
ntp server 3.3.3.3 key 1
```

3. Run the **efa inventory device ntp server delete** command to delete an NTP server.

```
efa inventory device ntp server delete ?
Flags:
      --ip string        Comma separated range of device IP addresses.
      --ntp-ip string    NTP server IP address
      --fabric string    fabric name.
-h, --help             help for ntp
```

   Example:

```
efa inventory device ntp server delete --ntp-ip 3.3.3.3 --ip 10.20.246.10

efa inventory device ntp server delete --ntp-ip 3.3.3.3 --fabric clos_fabric
```

4. Run the **efa inventory device ntp server list** command to display the list of NTP servers configured using EFA.

```
efa inventory device ntp server list ?
Flags:
      --ip string        Comma separated range of device IP addresses.
      --fabric string    fabric name.
-h, --help             help for ntp
```

Example:

```
efa inventory device ntp server ntp server list --ip 10.20.246.10

efa inventory device ntp server ntp server list --fabric clos_fabric
```

5. Run the **efa inventory device ntp disable-server** command to disable SLX acting as an NTP servers to other clients. SLX cannot be an NTP server to other hosts.

```
efa inventory device ntp disable-server ?
Flags:
      --ip string        Comma separated range of device IP addresses.
      --enable           Disable ntp server. Valid values are yes/no.
      --list             List disable-server on devices.
      --fabric string    fabric name.
-h, --help             help for ntp
```

- Disable the NTP server on given device

```
efa inventory device ntp disable-server --enable yes --ip 10.20.246.10
```

- Enable the NTP server on given device

```
efa inventory device ntp disable-server --enable no --ip 10.20.246.10
```

- Disable the NTP server at fabric level

```
efa inventory device ntp disable-server --enable yes --fabric clos_fabric
```

- Enable the NTP server at fabric level

```
efa inventory device ntp disable-server --enable no --fabric clos_fabric
```

- List the NTP disable-server on given device

```
efa inventory device ntp disable-server --list --ip 10.20.246.10
```

- List the NTP disable-server on at fabric level

```
efa inventory device ntp disable-server --list --fabric clos_fabric
```

Configure RME on Interface

The following CLI lets you configure Redundant Management Ethernet (RME) on the SLX interface. This feature is supported only on 9150, 9250, 9740 SLX platforms. The configuration set by you is persisted in EFA database. DRC is supported:

- **efa inventory device interface set-fec**
- **efa inventory device interface set-link-error-disable**
- **efa inventory device interface set-snmp**
  - **efa inventory device snmp community create**
  - **efa inventory device snmp community delete**
  - **efa inventory device snmp community list**
  - **efa inventory device snmp user create**
  - **efa inventory device snmp user delete**
  - **efa inventory device snmp user list**
  - **efa inventory device snmp host create**

- **efa inventory device snmp host delete**
- **efa inventory device snmp host list**

Note

For more information about the commands, see the *Extreme Fabric Automation Command Reference, 2.7.0*

Internally, EFA automatically sets the PPS (packets per second) value after RME is enabled. For SLX 9150 and 9250 devices, PPS is set to 8000.

On SLX 9740, the BPS (bits per second) is set to 20000 Kbps after RME is enabled.

1. Run the **efa inventory device interface redundant-management** command.

```
efa inventory device interface redundant-management [flags]
Flags:
      --ip string        Comma separated range of device IP addresses.
      --if-name string   only one interface name. Example: 0/50
      --enable string    Valid values: true, false
      -h, --help         help for redundant-management
      --- Time Elapsed: 9.610987ms ---
```

   Example:

```
efa inventory device interface redundant-management --ip 10.20.246.10 --if-name 0/17 --
enable true
```

   - Run the following command to disable RME.

```
efa inventory device interface redundant-management --ip 10.20.246.10 --if-name
0/17 --enable false
```

2. Configure device on the SLX.

```
SLX# show running-config interface eth 0/17
interface Ethernet 0/17
 redundant-management enable
 no shutdown
!
```

Device Configuration Synchronization

During the first service boot after upgrade to EFA, EFA queries the SNMP and NTP configuration on the device. These configurations persist in the database, which is managed by EFA.

Breakout interfaces and interfaces that are in admin-state down, with non-auto speed or non-default MTU values, persist in the database and are managed by EFA

If you update the configuration, use the EFA CLI, not the SLX-OS CLI on the device. This ensures that the device configuration matches the EFA configuration.

SLX Configuration Backup

You can back up the device running the configuration that can be included as part of the existing EFA backup.

Only one copy per device is included in a particular backup file.

During a backup operation, no configuration changes are allowed through EFA.

You can run a backup on demand or at an interval of your choosing.

Any failure during the backup process is reported.

CLI Commands for Backups

Device backup is integrated with the existing system backup CLI.

Additional parameters allow you to specify either the fabric name or list of devices for which to run a backup.

Showing and Updating Backup Settings

There are commands for showing and updating backup settings.

Showing Backup Settings

`efa system settings show` lets you display all the system settings that have been configured.

Example

```
efa system settings show
+-----------------------------+-------------+
|          SETTING            |   VALUE     |
+-----------------------------+-------------+
| Max Backup File Limit       | 5           |
+-----------------------------+-------------+
| Max Supportsave File Limit | 5           |
+-----------------------------+-------------+
| Backup Schedule             | 0 0 * * *   |
+-----------------------------+-------------+
| Remote Server IP            | 10.20.241.7 |
+-----------------------------+-------------+
| Remote Server Username      | root        |
+-----------------------------+-------------+
| Remote Server Password      | ****      |
+-----------------------------+-------------+
| Remote Server Directory     | /root/vinod/    |
+-----------------------------+-------------+
| Remote Transfer Protocol      | scp         |
+-------------------------------+-------------+
| Periodic Device Config Backup | yes         |
+-------------------------------+-------------+
--- Time Elapsed: 831.836375ms ---
```

For more information, see `efa system settings show` in the *Extreme Fabric Automation Command Reference, 2.7.0*.

Updating Backup Settings

There are settings for configuring remote server details where the backup is copied. In case remote server details are missing, the backup is copied on the same server where EFA is installed, which is also the current behavior of the system backup.

Passwords are encrypted using an AES algorithm and stored in the database.

Remote server validation is performed to validate whether the provided details of a remote server are valid or reachable (if you enter only an IP address, the application checks in the database for the remaining parameters - if they are missing, then it is treated as an error). All four parameters (IP, username, password, and directory-path) are expected for validation, either from the user or the database.

Transfer of a backup archive on a remote server is done through the SCP protocol.

The `efa system settings update` command lets you make the updates.

Example

```
efa system settings update --remote-server-ip 10.20.241.7
--remote-server-username root --remote-server-password pass --remote-server-directory /
root/vinod/
Setting Update Successful
--- Time Elapsed: 148.800033ms ---
```

Resetting System Backup Settings

You can reset the updated system backup settings to default values.

The `efa system settings reset` command lets you make the updates.

Example

```
efa system settings reset --max-backup-files
Reset System Settings is Successful
```

For more information, see `efa system settings reset` in the *Extreme Fabric Automation Command Reference Guide*.

Backup

To perform the backup, use the `efa system backup` command with options for specifying fabric or device details.

For more information, see `efa system backup` in the *Extreme Fabric Automation Command Reference, 2.7.0*.

Backup Scenarios

The following backup scenarios showcase the use of different options with the `efa system backup` command.

Run system backup on remote

```
efa system backup --remote
Generating backup of EFA...
Backup Location: /var/log/efa/backup/EFA-2.7.0-110-2022-03-28T11-37-00.936.tar
--- Time Elapsed: 5.741750131s ---
```

Run system backup without device configuration backup

```
efa system backup --remote
Generating backup of EFA...
```

```
Backup Location: root@10.20.241.7:/root/vinod/EFA--2.7.0-110-2022-03-28T11-37-00.936.tar
--- Time Elapsed: 5.741750131s ---
```

Run system backup by taking configuration backup of all devices that are part of the fabric specified

```
efa system backup --fabric default --remote
Generating backup of EFA...
Backup Location: root@10.20.241.7:/root/vinod/EFA--2.7.0-110-2022-03-28T11-37-00.936.tar
--- Time Elapsed: 5.741750131s ---
```

Run system backup by taking configuration backup of all fabrics and its devices

```
efa system backup --fabric-all --remote
Generating backup of EFA...
Backup Location: root@10.20.241.7:/root/vinod/EFA--2.7.0-110-2022-03-28T11-37-00.936.tar
--- Time Elapsed: 5.741750131s ---
```

Run system backup by taking configuration backup of all devices that are specified

```
efa system backup --device-ip 10.20.1.2,10.20.1.3,10.20.1.4 --remote
Generating backup of EFA...
Backup Location: root@10.20.241.7:/root/vinod/EFA--2.7.0-110-2022-03-28T11-37-00.936.tar
--- Time Elapsed: 5.741750131s ---
```

Error message: Fabric does not exist

```
efa system backup --fabric default --remote
Generating backup of EFA...
Backup Location: root@10.20.241.7:/root/vinod/EFA--2.7.0-110-2022-03-28T11-37-00.936.tar
Backup is partially success:
    Fabric does not exist
--- Time Elapsed: 5.741750131s ---
```

Error message: Device not found

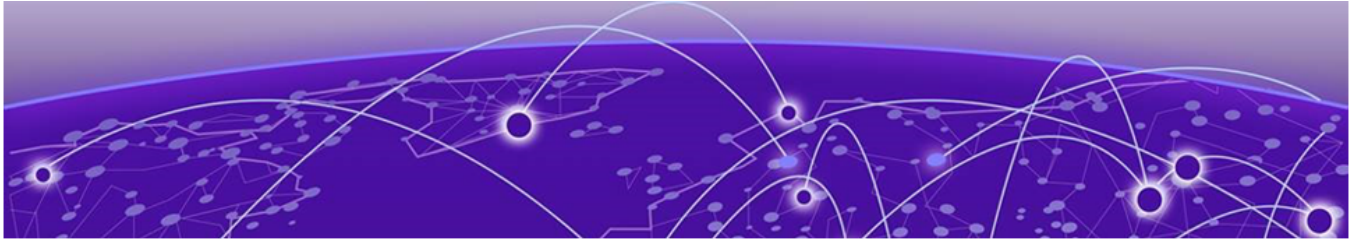
```
efa system backup --device-ip 10.20.1.5,10.20.1.6 --remote
Generating backup of EFA...
Backup Location: root@10.20.241.7:/root/vinod/EFA--2.7.0-110-2022-03-28T11-37-00.936.tar
Backup is partially success:
    Device 10.20.1.5 not found
--- Time Elapsed: 5.741750131s ---
```

Error message: Operation not allowed

```
efa system backup --device-ip 10.20.1.2,10.20.1.3 --remote
Generating backup of EFA...
Backup Location: root@10.20.241.7:/root/vinod/EFA--2.7.0-110-2022-03-28T11-37-00.936.tar
Backup is partially success:
    Devices [10.20.1.2] failed to get config backup as its locked for configuration
change by process [Firmware download].
--- Time Elapsed: 5.741750131s ---
```

Backup Schedule

Existing features of the wider backup schedule work here, too. Additionally, the backup gets SLX configuration backup of all those devices that are associated with a valid fabric.



EFA Event Management

RASlog Service on page 279
Notification Service on page 280
EFA as SNMP Proxy on page 282

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.

Notification Service

Notifications are alerts, tasks, and events that EFA sends to subscribers by HTTPS webhook or syslog over RELP (Reliable Event Logging Protocol).

Overview

All notifications are derived from the syslog events that are received from the devices that EFA manages.

Alerts are notifications that EFA services send for unexpected conditions, such as the following:

- Loss of switch connectivity
- Failure to configure the fabric, tenant, or endpoint group (EPG) on the device
- Failure to perform operations such as port up or port down, set speeds, and breakout mode
- Firmware download failure
- Devices exiting maintenance mode

Task notifications are based on user-driven or timer-based operations, such as the following:

- Registering or updating a device
- Device timer collection completed
- Adding devices to a fabric
- Creating, updating, or deleting a fabric
- Creating, updating, or deleting a tenant
- Creating, updating, or deleting an endpoint group

Notification methods

EFA supports two methods of notification: HTTPS webhook and syslog (using RELP over TLS). The format of the notifications is the same for both methods. You can configure one or both methods.

Webhook

This REST API-based method is a POST operation. The notification payload is in the body of the HTTPS call. Use the **`efa notification subscribers add-https`** command to register a subscriber for this method of notification.

Syslog over RELP

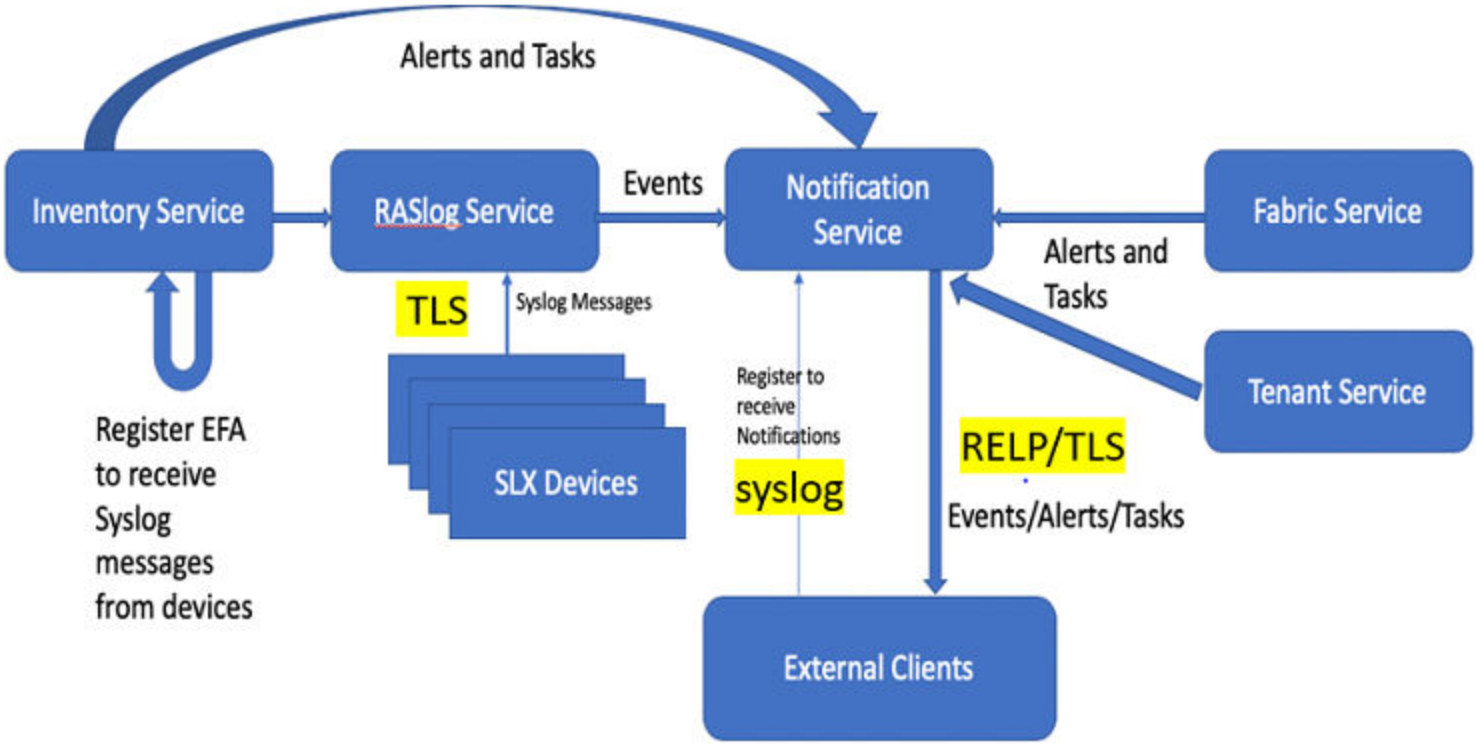
In this client-server method, the client initiates the connection and the server listens. In this scenario, the client is the Notification service and the server is the remote system where syslog is configured to work with RELP. Any external server that is configured with RELP can be registered as a subscriber to EFA notifications.

When RELP is configured with mTLS, EFA must be installed in secure mode. For more information, see the 'EFA Installation Modes" topic in the *Extreme Fabric Automation Deployment Guide, 2.7.0*.

Communication from SLX devices occurs over TLS. The certificates required for SLX devices to work with secure syslog are generated when the devices are registered.

Use the **`efa notification subscribers add-syslog-relp`** command to register a subscriber for this method of notification.

Notification workflow



In general, the order of operations is as follows:

1. For each SLX devices that is registered in EFA, EFA registers itself as a syslog receiver on each device.
2. The RASlog service receives syslog messages from the registered devices over TLS.
3. Messages are processed and converted to JSON, a format that is easier to search, sort, and perform operations on.
4. The Inventory service, Fabric service, and Tenant service send alerts and tasks to the Notification service.
5. External clients (subscribers) receive notifications by HTTPS webhook, syslog (using RELP over TLS), or both, depending on the subscription.

Notification example

In this example, the creation of a tenant sends a task notification to the syslog server.

Here, the tenant is created.

```
$ efa tenant create --name tnt1
Tenant created successfully.
--- Time Elapsed: 167.716003ms ---
```

And here are the corresponding syslog messages.

```
<5>Aug 20 11:25:28 10.24.95.154(10.24.95.154)
{"application":"ts","device_ip":"","message":"Tenant create request received
:request={\"name\":\"tnt1\"}","scope":"user","status":"started","task":
"EFA-001001","timestamp":"2020-08-20T13:42:55Z","type":"Task","username":"extreme"}
```

```
<5>Aug 20 11:25:28 10.24.95.154(10.24.95.154)
{"application":"ts","device_ip":"","message":"Tenant create request success
:request={\"name\":\"tnt1\"}","scope":"user","status":"succeeded","task":
"EFA-001002","timestamp":"2020-08-20T13:42:55Z","type":"Task","username":"extreme"}
```

EFA as SNMP Proxy

EFA acts as the SNMP Manager for all the SLX devices and agents and receives the traps from all the devices in its inventory.

Simple Network Management Protocol (SNMP) traps are alert messages sent from a remote SNMP-enabled device to a central collector, the SNMP Manager. Trap messages are the main form of communication between SNMP monitoring tools – an SNMP Agent and an SNMP Manager.

EFA acts as the SNMP Manager for all the SLX devices and agents and receives the traps from all the devices in its inventory. Once you register an SLX device with EFA, EFA automatically configures the SLX device to send v3 traps to EFA.

EFA acts as an SNMP proxy for all the SNMP v2 and v3 traps received from the SLX devices, forwarding them onto an external trap receiver, if there is one.

- EFA subscribes to be a v3 trap receiver with a predefined v3 user name, authentication key, and privacy key.
- If you set up EFA to be a v2c trap receiver, you must provide a community string.

During an update operation, EFA verifies that it is still registered to receive traps from the SLX devices. If a device is unregistered from EFA, the SNMP configuration on the device is updated to no longer send traps to the EFA IP address.

Commands for configuring SNMP

The following commands are available for configuring SNMP on the SLX device. The configuration you set is persisted in the EFA database. DRC is supported.

- `efa inventory device snmp community create`
- `efa inventory device snmp community delete`
- `efa inventory device snmp community list`
- `efa inventory device snmp user create`
- `efa inventory device snmp user delete`
- `efa inventory device snmp user list`
- `efa inventory device snmp host create`
- `efa inventory device snmp host delete`
- `efa inventory device snmp host list`

For more information about these commands, see *Extreme Fabric Automation Command Reference, 2.7.0*.

Notes

- The device IP address is the one included in `SNMP-COMMUNITY-MIB::snmpTrapAddress.0`. It is not the EFA IP address.
- EFA forwards all received traps. In other words, no trap is filtered out.

- Port 162 on the host where EFA is installed must be available. During a fresh installation, the port availability is checked and the installer returns an error if the port is not available. However, during an upgrade from a previous version of EFA, you must ensure that the port is free.

For more information about SLX-OS MIBs, see the *Extreme SLX-OS MIB Reference* for your version of SLX-OS.

Limitations

- A maximum of four trap subscribers is supported.
- V2c and v3 SNMP subscribers are not validated.
- Only traps generated by SLX devices are forwarded. Alerts and alarms from EFA itself are not forwarded.
- Only traps are forwarded. Current EFA tasks or alerts and syslog messages are not forwarded as traps.
- SNMP Informs are not supported.
- There is no in-band support for trap forwarding.
- The Drift and Reconcile process does not show a drift in device configuration for SNMP v3 trap configuration that EFA has pushed. However, every time the device update operation runs, EFA checks if the device is configured to send traps to EFA and if not, pushes the configuration again.
- For a multi-node deployment during failover of the active node, some traps might be missed while the SNMP service is bootstrapping on the new active node. There is no loss of traps if the standby node goes down.

Migration of existing switch configuration

When you boot the service for the first time after upgrade, any SNMP and NTP configuration on the switch are queried and persisted in the database and managed by EFA.

Similarly, any breakout interfaces or interfaces that have status admin-state DOWN and have a non-auto speed or non-default MTU value are persisted in the database and managed by EFA.

If you have additional updates to make to these configurations, you must make them manually using the EFA commands only. See Commands for Configuring SNMP.

If these configurations are updated using the SLX commands directly on the switch (meaning, not by using the EFA CLI), they are considered as drifted and are reconciled.

gosnmp-service

The gosnmp-service is responsible for persisting the trap subscribers, receiving the SNMP traps, and forwarding them to the subscribers.

The service is stateless, so no historical data (that is, previously received traps) is persisted.

For high availability deployment, the service runs in active-active mode, however, since the VIP is bound to one host at a time, the pod running on the active node receives the traps. On failover, the standby node takes over and the SNMP service running on that node forwards the traps.

You may have multiple IP subnets configured to access EFA. In such a case, EFA creates multiple subinterfaces under the management interface to which EFA is bound. EFA does not determine which interface sends out the trap, syslog or webhook. The administrator is responsible for configuring a route

to the recipient. If one is found, the server sends out the trap. For more information, see Multiple Management IP Networks on page 53.

Drift and Reconcile (DRC) and Idempotency for SNMP

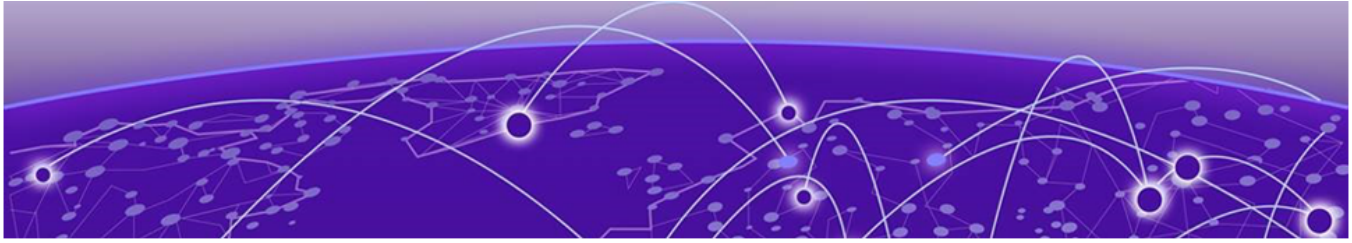
The table below captures the various attributes of the SNMP configuration interface for which DRC and idempotency is supported. A drift is identified if any of the fields below is modified through the SLX CLI or other management tool. A reconcile operation pushes the intended configuration to SLX, so keeping the SLX configuration in sync with EFA.

Regarding idempotency for creating an entry which already exists in EFA, an error message is returned stating that the user already exists.

Field	Identify Drift	Reconcile config	Idempotency	Comments
Community deleted	Yes	Yes	No	A valid error message is shown when a non-existent community is deleted.
Group name associated with community is modified	Yes	Yes	Not Applicable	
Group deleted	Yes	Yes	Not Applicable	
Modify group version.	No	No	Not Applicable	SLX does not support editing the SNMP group version.
Modify read, review, or write view or notify view associated with group.	No	No	Not Applicable	SLX does not support editing the SNMP views associated with the group.
Modify groupname associated with SNMP user.	Yes	Yes	Not Applicable	
Modify authentication protocol associated with SNMP user.	Yes	Yes	Not Applicable	
Modify authentication password associated with SNMP user.	Yes	Yes	Not Applicable	

Field	Identify Drift	Reconcile config	Idempotency	Comments
Modify privacy protocol associated with SNMP user.	Yes	Yes	Not Applicable	
Modify privacy password associated with SNMP user.	Yes	Yes	Not Applicable	
Delete SNMP user.	Yes	Yes	Not Applicable	A valid error message is shown when a non existent user is deleted.
Modify encrypted keyword associated with SNMP user.	Yes	Yes	Not Applicable	
Modify authentication type associated with group, meaning: auth, noauth, notify.	Yes	Yes	Not Applicable	
Delete SNMP host entry.	Yes	Yes	No	A valid error message is shown when a non existent host is deleted.
Modify encrypted keyword associated with SNMP user.	Yes	Yes	Not Applicable	
Modify authentication type associated with group, meaning: auth, noauth, notify.	Yes	Yes	Not Applicable	
Delete SNMP host entry.	Yes	Yes	No	A valid error message is shown when a non existent host is deleted.
Update SNMP host security level.	No	No	Not Applicable	
Update SNMP host source interface.	No	No	Not Applicable	
Update SNMP host UDP port.	No	No	Not Applicable	

Field	Identify Drift	Reconcile config	Idempotency	Comments
Update SNMP host VRF.	No	No	Not Applicable	
Update SNMP host engine id.	Yes	Yes	Not Applicable	
Update of SNMP host notification type [traps, informs]	Yes	Yes	Not Applicable	
Update of SNMP view MIB OID access [included, excluded]	Yes	Yes	Not Applicable	



Known Limitations

Known Limitations in EFA on page 287

Known Limitations in EFA

Provided VNI: 1006 already consumed in fabric

Symptom	Condition	Workaround
Endpoint group (EPG) create fails with VNI resource not being available in the fabric.	Run EPG create, delete, and re-create CLI in quick succession: 1. Create EPG or Networks with user-provided VNI parameter. 2. Delete the EPG. 3. Create the EPG again with the same parameters as in step 1.	Provide a wait of 30 seconds between the create and delete CLI on the same EPG.

VRF delete from EPG and re-adding VRF to EPG fails intermittently

Symptom	Condition	Workaround
Endpoint group (EPG) update **`vrf-add`** operation fails with the reason as VRF to be added has conflicting VRF on the switch.	Run EPG update **`vrf-add`**, **`vrf-delete`**, and **`vrf-add`** operation CLI in quick succession: 1. Update EPG for operation **`vrf-add`**. 2. Update EPG for operation **`vrf-delete`**. 3. Update the same EPG again with operation **`vrf-add`** for the same VRF which was deleted in step 2.	Wait of 30 seconds between the EPG update **`vrf-add`** and vrf-delete operations on the same EPG.

vrf_route_target_mapping error while creating EPG (after VRF delete)

Symptom	Condition	Workaround
When VRF is added and deleted to or from and Endpoint Group, in quick succession, multiple times, events received from inventory service can get interleaved with the commands. This causes EFA command execution path to find database entries that are yet to be deleted due to previous command run.	Issue is observed when **`vrf-add`** and **`vrf-delete`** operations are run multiple times on Endpoint Group in quick succession.	Wait for a few minutes before running the **`vrf-add`** again on Endpoint Group.

Inventory device update fails to change device table router-ip field

Symptom	Condition	Workaround
If the router-id is changed on the device (add, delete, or modified), the device table router-ip field does not change after you do inventory device update.	It is found that inventory did detect the diff and generate the RouterBgpUpdatedMsg to Tenant. Tenant did not handle this message.	Add hook in the RouterBgpUpdatedMsg on tenant handler to set the RouterIP. This enables the tenant DB to have the same value with the device after update.

REST operations are not retried (as applicable) during the service boot

Symptom	Condition	Workaround
REST operations are not retried (as applicable) during the service boot up.	The status are not set for all the REST operations AFTER publishing all the necessary events on the message bus.	For all the REST operations, set the status AFTER publishing all the necessary events on the message bus.

RBAC: EFA shows "export EFA_TOKEN" command suggestion when a tenant user logs in

Symptom	Condition	Workaround
EFA shows the following message after a tenant user with RBAC logs in to the system: Please type this in your shell: **export EFA_TOKEN**=eyJhbGciOiJSUzI1NiIsImtpZCI6IjEuMCIsInR5cCI6IkpXVCJ9.eyJjb21tb25fbmFtZSI6IkVGQSBUb2tlbiBTZXJ2aWNlIiwidWFzIjpbeyJ0YXJnZXQiOiJFRkEiLCJyb2xlIjoiVlIyLVRudEFkbWluIn1dLCJvcmciOiJFeHRyZW1lIE5ldHdvcmtzIiwidmVyIjoiMS4wIiwiaWQiOiIiLCJleHAiOjE2NDUyNDcxNDIsImp0aSI6IjZjMjA4ZDUxLTkwNzgtMTFlYy1iZjk5LWNhNzk1MDY1YzIwNyIsImlhdCI6MTY0NTE2MDc0MiwiaXNzIjoiRUZBIFRva2VuIFNlcnZpY2UiLCJuYmYiOjE2NDUxNjA3NDIsInN1YiI6InVzZXIyIn0.b7m5PINijeEdNSqnTeE2ZhUrqKLKQAu079vXyBIdgHbXKt9ULfa03vMU1jfBO1qFb1-x0oHmsAQ0pSsF5JLeMaMzMflLf78ktZO8U5IePq72vM5en35IR-DNLyoGIZBeFeG6ZbBMoETzz5vf9OuefgQID3YdjcALr7yllCgDmLVFlgson77yCBpkTK15xm1GRbtL7JKXZzShBE7E3kdW7N71MdM85Gc3r4l-c8sfz7eo06gKrfTq9wXCv4_LVzR6-KRSg6NyLq363WEpcK1A2Hs0Wo3T9TpquYHNaCWA5I1QTsG-RHFdg4kxZP2fQpUp6Bgy1s6k59PVPn4-M-a8lA— Time Elapsed: 4.619465187s —	When a user is created with the default login shell as sh.	EFA supports only bash shell for login or any other CLI commands.

EFA CLI or REST request with scale config takes longer than 15 minutes fails

Symptom	Condition	Workaround
Tenant2 delete is successful whereas deleting Tenant1 took more than 15 minutes and failed with the following message: Error : service is not available or internal server error has occurred, please try again later Tenant service was running. Tenant1 was not available after the error.	When you try to delete tenants in a single rack small data center deployment configured with scale tenant config	Any CLI or REST tenant operations, and any fabric operations taking more than 15 minutes will timeout at the client side. The operation completes in the background. Run the efa tenant show command to view the actual state of the operation.