Engineering
Simplicity

Network Configuration Example

Collapsed Spine with EVPN Multihoming

Published
2025-01-17

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

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Network Configuration Example Collapsed Spine with EVPN Multihoming

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

Table of Contents

About This Guide

Use this network configuration example to configure a collapsed spine architecture with EVPN multihoming, also known as ESI-LAG, for a data center environment.

1

CHAPTER

Example: How to Configure a Collapsed Spine Data Center Architecture With EVPN Multihoming

Overview of Collapsed Spine Architecture with EVPN Multihoming

IN THIS SECTION

About This Network Configuration Example

This Network Configuration Example (NCE) shows how to set up a collapsed spine data center fabric that lets you use your existing Layer 2 top-of-rack switches in place of leaf devices. It also shows how to use EVPN multihoming to provide multichassis LAG functionality for Layer 2 top-of-rack switches.

In addition, it optionally shows how to set up data center interconnect and advanced security services for inter-tenant traffic through an SRX chassis cluster.

NOTE: Juniper Networks requires a license for EVPN-VXLAN on QFX Series switches. See the Licensing Guide for more information.

SEE ALSO

Collapsed Core with EVPN Multihoming

Use Case Overview

Large enterprise data centers are migrating to overlay-based architectures using an end-to-end IP fabric with a VXLAN overlay and an EVPN control plane. Using a Layer 3 IP-based underlay in the core coupled with an EVPN-VXLAN overlay on the top-of-rack (ToR) switches, data center and cloud operators can deploy much larger networks than are possible with traditional Layer 2 Ethernet-based architectures.

However, legacy ToR switches may not support EVPN-VXLAN. In data centers with these ToR switches that only support Layer 2 traffic, the spine switches are responsible for inter-VLAN routing. A data center architecture is needed that decouples the underlay network from the tenant overlay network with technologies such as VXLAN. You can accomplish this with a collapsed spine architecture.

A collapsed spine architecture has no leaf layer. Instead, the Layer 3 IP-based underlay and the EVPN-VXLAN overlay functionality that normally runs on leaf switches is collapsed onto the spine switches. The spine switches also act as a border gateway.

A collapsed spine architecture with EVPN multihoming is ideal for organizations with:

- Plans to move to an IP fabric-based architecture with EVPN-VXLAN overlay.
- Small data centers with a mostly north-south traffic pattern.
- A need to extend Layer 2 traffic across data centers.
- Multi-vendor legacy ToR switches that do not support EVPN-VXLAN.
- Current or future requirements to support more than two spine switches to ensure adequate bandwidth during maintenance or a spine failure.
- A need for an alternative to an MC-LAG (ICCP protocol) architecture.

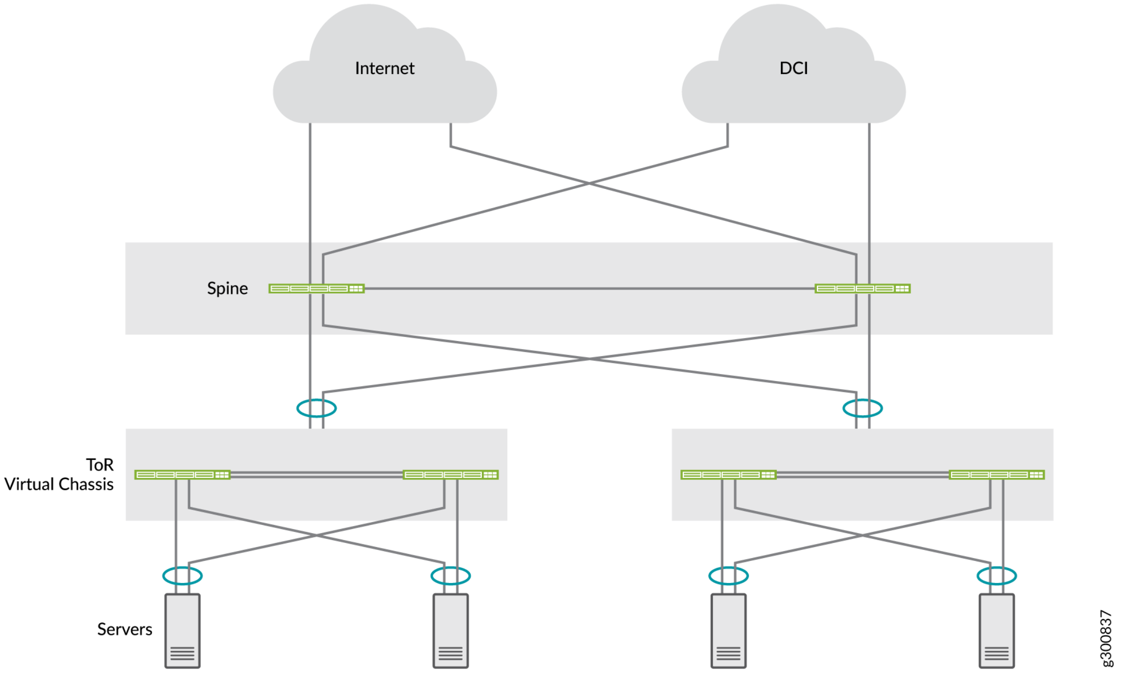
Technical Overview

IN THIS SECTION

Collapsed Spine with EVPN Multihoming Architecture Overview

This NCE shows how to deploy a collapsed spine architecture for two data centers that each have two QFX5120 spine switches and two Layer 2 ToR switches deployed as a Virtual Chassis. The data centers are connected to each other through the spine devices with Layer 3 Data Center Interconnect (DCI). Use EVPN multihoming to multihome the ToR switches to the spine devices. The the servers are multihomed to the ToR switches. Figure 1 on page 4 shows the completed collapsed spine architecture.

Figure 1: Collapsed Spine Architecture with EVPN Multihoming



For multicast support, we deliver:

- Layer 3 multicast in a QFX5120 collapsed-spine design using EVPN OISM with symmetric bridge domains.
- Layer 2 multicast IGMPv2 snooping in EVPN-VXLAN using:

- EVPN Selective Multicast Ethernet Tag (SMET) Type 6 routes
- EVPN Join and Leave Sync (Type 7 and Type 8) routes when a multicast receiver is multihomed at Layer 2 to the collapsed spine devices using an ESI-LAG.

Understanding Collapsed Spine Architecture

In a collapsed spine architecture, the spine devices act as both spine and leaf devices. Because the ToRs are Layer 2 only and do not support VXLAN, they do not act as leaf devices. Normal leaf device activity is handled, or collapsed, onto the spine devices, which means that VXLAN is required only on the spine devices. The collapsed spine operates as a Layer 3 gateway and handles traffic between the VXLANs using IRB interfaces.

Understanding EVPN Multihoming

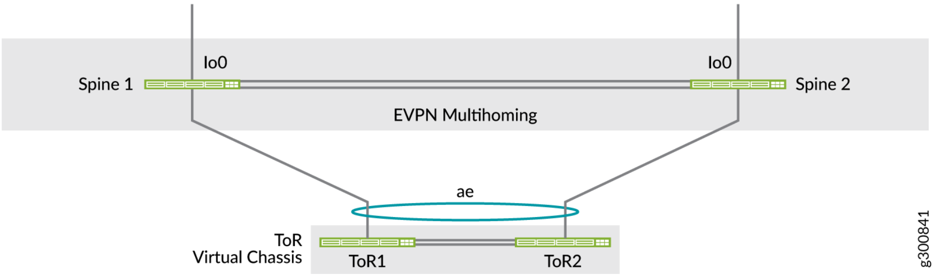
In a legacy data center with a collapsed spine architecture, the ToR switches need to be connected to the spine switches with multichassis link aggregation groups (MC-LAGs) to improve network resiliency. MC-LAG provides node-level redundancy and link-level redundancy. Traditionally, spine switches in these data centers use Inter-Chassis Control Protocol (ICCP) to provide MC-LAG functionality. However, MC-LAG with ICCP:

- Is a proprietary technology.
- Cannot efficiently stretch Layer 2 between data centers.
- Does not support more than two spine switches.

EVPN provides a standards-based multihoming solution that scales horizontally across two or more spine switches for additional resiliency and bandwidth in case of a spine failure. EVPN multihoming, also known as ESI-LAG, provides MC-LAG functionality for the Layer 2 ToR switches and the servers in this architecture without the drawbacks of ICCP-based MC-LAG.

A collapsed spine architecture where the ToR switches are multihomed to the spines is a data center architecture that supports legacy ToR switches when they do not support EVPN-VXLAN. Figure 2 on page 6 shows a collapsed spine architecture with two spine switches for simplicity and a ToR device implemented as a Virtual Chassis (see "Understanding Virtual Chassis" on page 10).

Figure 2: EVPN Multihoming of ToR Switches



SEE ALSO

> EVPN Multihoming Overview

Understanding VXLAN

Network overlays are created by encapsulating traffic and tunneling it over a physical network. The VXLAN tunneling protocol encapsulates Layer 2 Ethernet frames in Layer 3 UDP packets. VXLAN enables virtual Layer 2 subnets or segments that can span the underlying physical Layer 3 network.

In a VXLAN overlay network, each Layer 2 subnet or segment is uniquely identified by a virtual network identifier (VNI). A VNI segments traffic the same way that a VLAN ID segments traffic. As is the case with VLANs, endpoints within the same virtual network can communicate directly with each other. Endpoints in different virtual networks require a device that supports inter-VNI routing.

The entity that performs VXLAN encapsulation and decapsulation is called a VXLAN tunnel endpoint (VTEP). Each VTEP is typically assigned a unique IP address.

Understanding EVPN

EVPN is one of the extensions to BGP that allows the network to carry network layer reachability information (NLRI) such as Layer 2 MAC addresses and Layer 3 IP addresses. This control plane technology uses MP-BGP for MAC and IP address endpoint distribution, where MAC addresses are treated as routes. EVPN enables devices acting as VTEPs to exchange reachability information with each other about their endpoints.

EVPN provides multipath forwarding and redundancy through an all-active model. The access layer can connect to two or more spine devices and forward traffic using all of the links. If an access link or spine device fails, traffic flows from the access layer toward the spine layer using the remaining active links. For traffic in the other direction, remote spine devices update their forwarding tables to send traffic to the remaining active spine devices connected to the multihomed Ethernet segment.

Overlay Network

IN THIS SECTION

This architecture uses VXLAN as the overlay data plane encapsulation protocol and MP-BGP with EVPN signaling as the overlay control plane protocol.

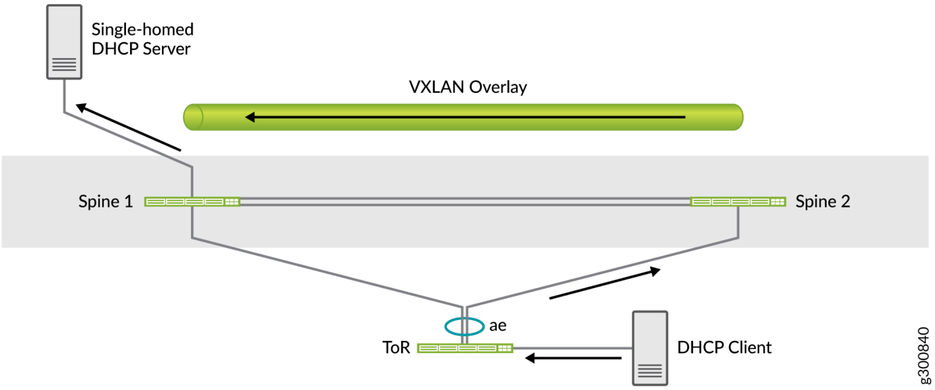
Data Plane Overlay

This architecture uses VXLAN as the overlay data plane encapsulation protocol on the collapsed spine switches. A switch that functions as a Layer 2 or Layer 3 VXLAN gateway acts as the VXLAN tunnel endpoint and can encapsulate and decapsulate data packets.

In a single data center deployment with two spine switches, the VXLAN overlay between the spine switches is used for traffic between the two devices. For example, if there is a single-homed server connected to one of the spine devices, the VXLAN overlay carries the traffic to the other spine device either by design or in the case of a link failure.

As shown in the figure below, the DHCP server is single-homed to Spine 1. Traffic from the DHCP client might get sent to Spine 2 because of load sharing. Spine 2 sends the traffic to the DHCP server over the VXLAN overlay with Spine 1.

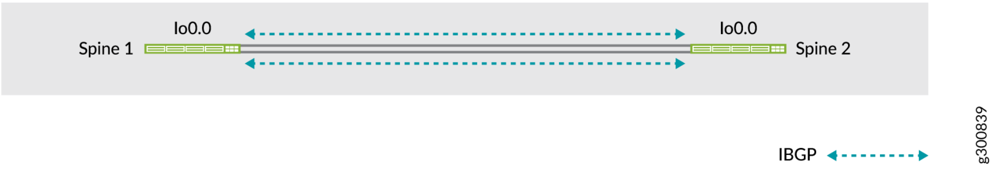
Figure 3: Data Plane Overlay Topology



Control Plane Overlay

MP-BGP with EVPN signaling acts as the overlay control plane protocol in this example. The spine switches establish IBGP sessions between each other. Figure 4 on page 8 shows the topology of the overlay network.

Figure 4: Control Plane Overlay Topology



SEE ALSO

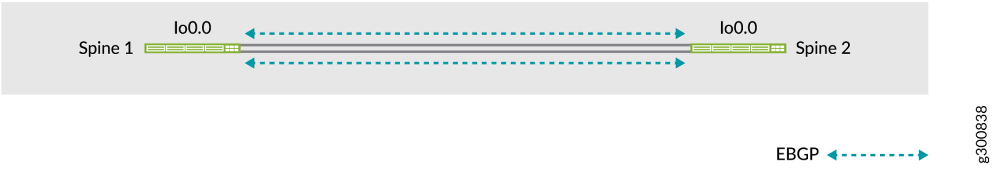
> Understanding EVPN with VXLAN Data Plane Encapsulation

Underlay Network

In smaller data centers there is no super spine layer so the spine switches are directly connected to each other. The spine switches can use a dynamic routing protocol in the underlay. The primary requirement in the underlay network is that all spine devices have loopback reachability. You can use any Layer 3 routing protocol to exchange loopback addresses between the core and spine devices.

In this example, we use EBGP as the underlay routing protocol between the spine switches. EBGP provides benefits like better prefix filtering, traffic engineering, and traffic tagging. Figure 5 on page 8 shows the topology of the spine underlay network.

Figure 5: Spine Underlay Topology



NOTE: Use at least two links between the spine switches. Loss of connectivity between the spine switches could lead to a split-brain state. See "Split-Brain State" on page 53 for more information.

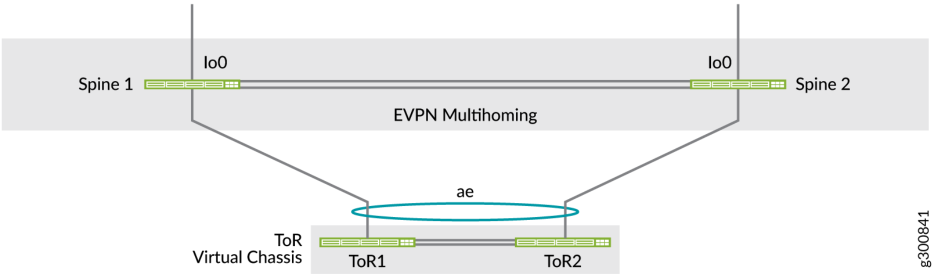
Top-of-Rack Switches

IN THIS SECTION

Because the ToR switches do not participate in the EVPN-VXLAN fabric and operate at Layer 2 only, you can implement them as a Virtual Chassis. In this example, the ToR switches are deployed as a two-member Virtual Chassis.

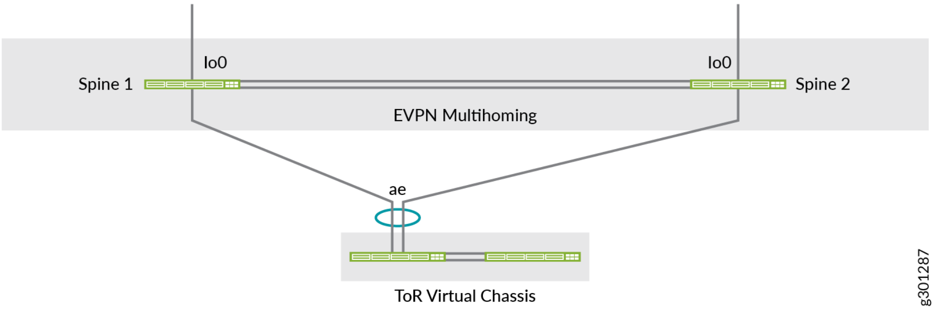
The uplinks from the ToR switches to the spine switches are Layer 2 trunk LAG ports with VLANs relevant to the ToR switch. Each Virtual Chassis is multihomed to two spine switches using EVPN multihoming. Figure 6 on page 9 shows the topology of a Virtual Chassis as a ToR device that is multihomed to the two spine devices. For redundancy and better resiliency, this figure shows spine to ToR Virtual Chassis connections that link to different Virtual Chassis members, so the Virtual Chassis ToR device is still reachable even if one of the Virtual Chassis members goes down.

Figure 6: ToR Switch Topology



The spine to ToR Virtual Chassis connections in the multihoming aggregated Ethernet links can also include links to the same Virtual Chassis member, which is how this network configuration example is configured. Figure 7 on page 10 shows a logical view of the multihoming topology that matches the configuration in this document.

Figure 7: ToR Switch EVPN Multihoming Topology in this Network Configuration Example



Understanding Virtual Chassis

In this example, we implement the ToR switches in a Virtual Chassis. Virtual Chassis can interconnect multiple standalone switches into one logical device and manage the logical device as a single chassis. Use Virtual Chassis for the ToR switches to:

- Manage multiple devices as a single device with the same or similar capabilities as the standalone device.
- Increase fault tolerance and high availability.
- Flatten your network and reduce networking overhead by allowing network devices to synchronize to one resilient logical device.
- Enable a simplified Layer 2 network topology that minimizes or eliminates the need for loop prevention protocols such as Spanning Tree Protocol (STP).
- Provide redundancy and load sharing for servers that are multihomed across the Virtual Chassis members.

> **NOTE**: Virtual Chassis provides a single control plane and distributed data plane for simplified management at the ToR layer. The ToR switches behave like line cards on a single chassis. Because the Virtual Chassis behaves like a single chassis, servers connected to the Virtual Chassis might experience downtime during software upgrades of the ToR switches.

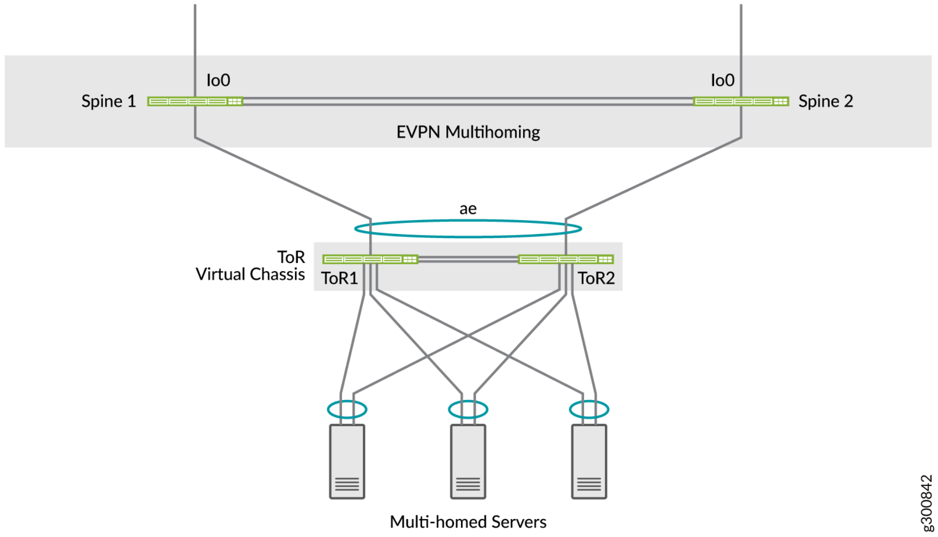
SEE ALSO

Understanding EX Series Virtual Chassis

Servers

The data center servers in this example are multihomed to the ToR switches that are deployed as a Virtual Chassis. Server connectivity can be distributed across the two ToR switches with LAG.

Figure 8: ToR Topology With Multihomed Servers

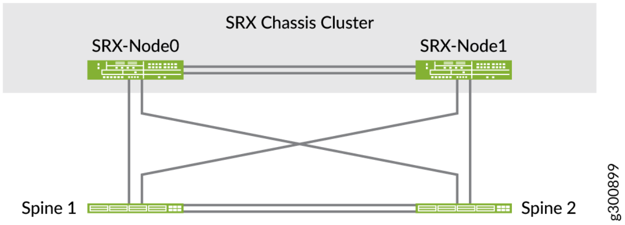


SRX Chassis Cluster

In this example, we are deploying SRX security devices in a chassis cluster that is connected to the spine devices to provide advanced security. In a chassis cluster, two SRX Series Firewalls operate as a single device to provide device, interface, and service-level redundancy. Configuration files and the dynamic runtime session states are synchronized between SRX Series Firewalls in a chassis cluster. Use an SRX chassis cluster to:

- Prevent single device failure that results in a loss of connectivity.
- Provide high availability between security devices when connecting branch and remote site links to larger corporate offices.
- Ensure connectivity in the event of a device or link failure.

Figure 9: SRX Chassis Cluster Implementation



SEE ALSO

SRX Series Chassis Cluster Configuration Overview

How to Configure a Collapsed Spine with EVPN Multihoming

IN THIS SECTION

Requirements

This example assumes that you have two data centers (DC1 and DC2) with separate networks. This example uses the following devices and software:

- DC1:
  - Two spine switches: QFX5120-48Y running Junos OS Release 18.4R2-S1.4
  - Two ToR switches: EX4300-48T running Junos OS Release 18.1R3-S6.1
  - Two security devices: SRX345 devices running Junos OS Release 18.2R3.4 (Optional add-on configuration)
  - Four servers
- DC2:
  - Two spine switches: QFX5120-48Y running Junos OS Release 18.4R2-S1.4
  - Two ToR switches: EX4300-48T running Junos OS Release 18.1R3-S6.1
  - Two servers

Each pair of ToR switches should already be configured as a Virtual Chassis. See Understanding EX Series Virtual Chassis for more information about forming a Virtual Chassis with EX4300 switches. This example configuration uses multihoming aggregated Ethernet links between the ToR Virtual Chassis and the two spine devices on only one member in the Virtual Chassis. If possible, for better resiliency, you can connect the multihoming aggregated Ethernet links between the Virtual Chassis and the spine devices using interfaces from different Virtual Chassis members.

Overview

IN THIS SECTION

Use this example to configure a collapsed spine architecture with EVPN multihoming of the ToR switches. We have two data centers with an optional Data Center Interconnect (DCI) configuration, an optional SRX cluster for added security, and an optional DHCP relay configuration. This configuration example shows you how to configure this architecture in DC1. You can use a similar configuration in DC2.

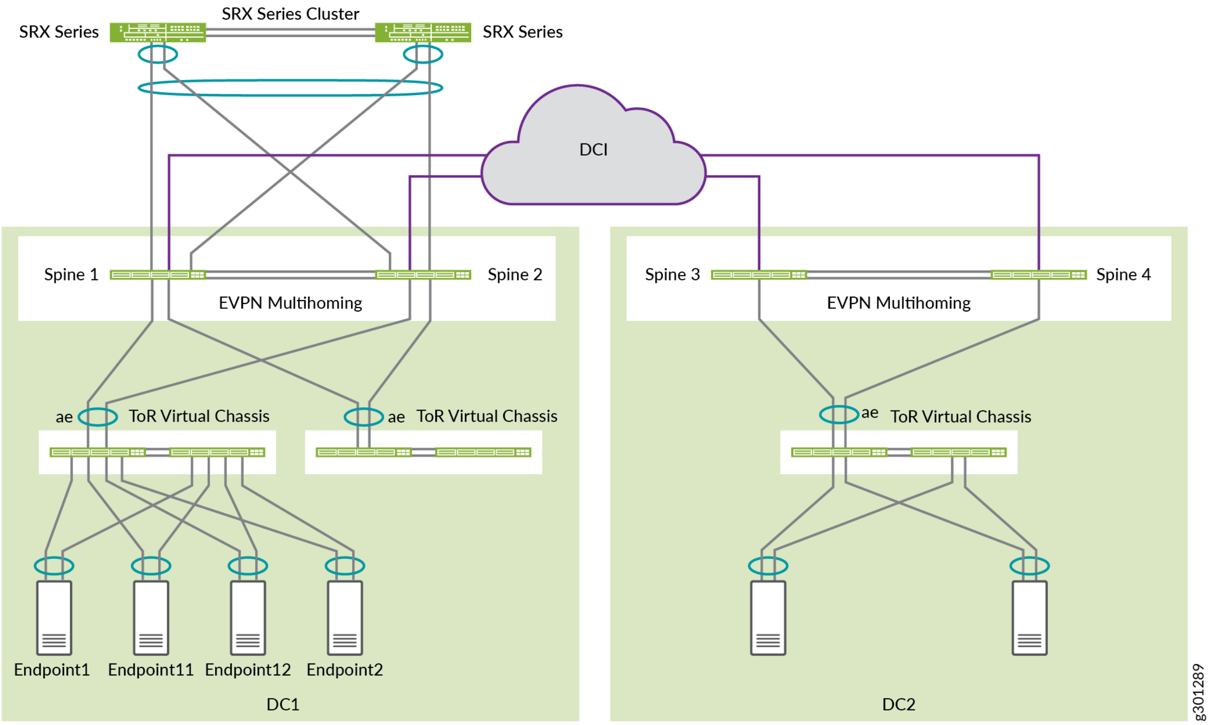
Topology

In this deployment, there are two data centers: DC1 and DC2. The data center networks are configured with a collapsed spine architecture using QFX5120 as the spine switches. In this case, we recommend that you limit the EVPN-VXLAN fabric to the local data center.

You can optionally connect the data centers using Layer 3 DCI in the underlay. This use case does not require Layer 2 stretch between the data centers. Inter-data center traffic is Layer 3 only and is routed through the SRX cluster in DC1 for advanced inspection.

Figure 10 on page 14 shows the logical connectivity between the components used in this NCE.

Figure 10: Logical Topology



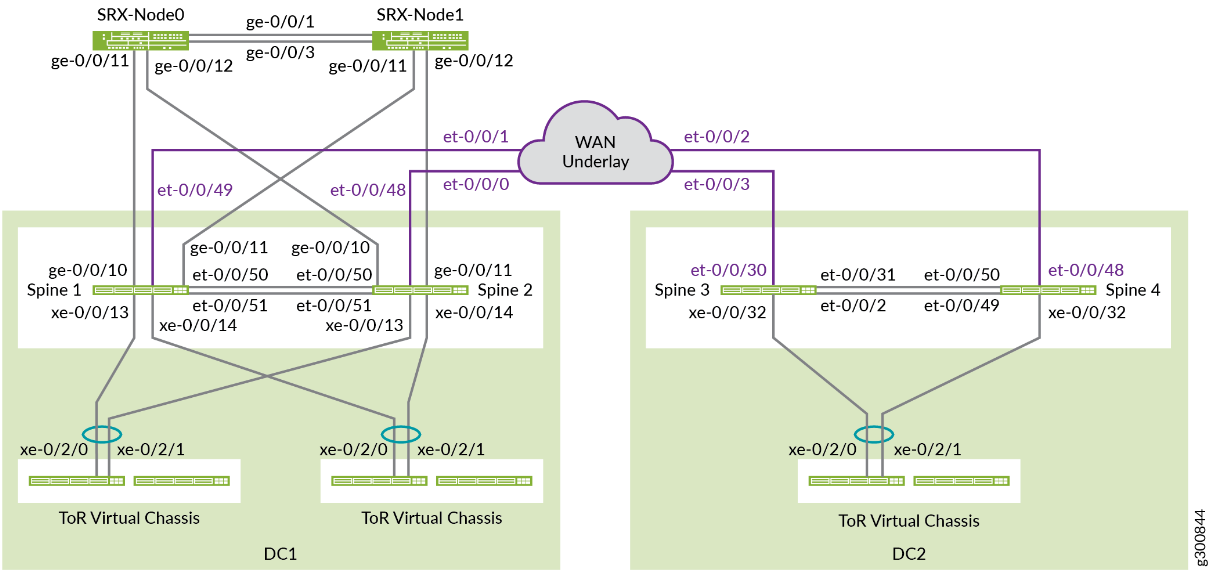
There are two tenants in DC1: JNPR1 and JNPR2. Any inter-tenant traffic between JNPR1 and JNPR2 in DC1 is routed through the SRX firewall cluster for security.

- DC1:
  - VLANs 201 and 202 belong to JNPR1.

  - VLANs 211 and 212 belong to JNPR2.
  - DC1 has servers in VLANs 201, 202, 211, and 212.
- DC2:
  - VLANs 221 and 222 belong to the default tenant, which is the same as the default routing instance.
  - DC2 has servers in VLANs 221 and 222.

Figure 11 on page 15 shows the physical connectivity between the components used in this NCE.

Figure 11: Physical Topology



Before You Begin

IN THIS SECTION

You need to implement some basic configuration on your devices before you configure the fabric.

Procedure

Step-by-Step Procedure

1. By default, no aggregated Ethernet interfaces are created. You must set the number of aggregated Ethernet interfaces before you can configure them. Once you set the device count, the system creates that number of empty aggregated Ethernet interfaces, each with a globally unique MAC address. You can create more aggregated Ethernet interfaces by increasing the device count to the number of ESI-LAG interfaces required on the device.

   Set the number of aggregated Ethernet interfaces on all spine switches and ToR switches.

   ```
   set chassis aggregated-devices ethernet device-count 15
   ```

2. Ports 0 to 47 on a QFX5120-48Y operate as 10-gigabit ports by default. The SRX devices support only 1 gigabit. Configure the ports on Spine 1 and Spine 2 that are connected to the SRX Series Firewall to be 1-gigabit ports. In this case, these ports are ge-0/0/10 and ge-0/0/11. To enable 1 gigabit on these ports, configure the speed of the first port in the quad, which in this case is ge-0/0/8.

   Use the following statement on Spine 1 and Spine 2:

   ```
   set chassis fpc 0 pic 0 port 8 speed 1G
   ```

   NOTE: You can configure the 1-gigabit and 25-gigabit port speeds only per quad (group of four ports) and not individually. All ports operate at a single speed within the quad. For instance, if you configure ports 8 through 11 to operate as 1-gigabit Ethernet ports and you insert a 10-gigabit SFP+ transceiver in port 10, an interface is not created for this port.

3. Auto speed detection mode detects 100-gigabit Ethernet interfaces and 40-gigabit Ethernet interfaces and automatically channelizes them. Automatic channelization and speed detection are enabled by default. In this example, auto channelization would divide each 40-gigabit Ethernet interface into four 10-gigabit Ethernet interfaces.

   Disable auto channelization on ports et-0/0/2 and et-0/0/31 on Spine 3 and ports et-0/0/49 and et-0/0/50 on Spine 4 so that they remain 40-gigabit Ethernet interfaces.

Spine 3:

```
set chassis fpc 0 pic 0 port 2 channel-speed disable-auto-speed-detection
set chassis fpc 0 pic 0 port 31 channel-speed disable-auto-speed-detection
```

Spine 4:

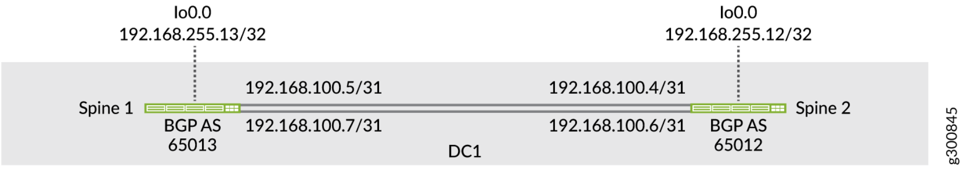
```
set chassis fpc 0 pic 0 port 49 channel-speed disable-auto-speed-detection
set chassis fpc 0 pic 0 port 50 channel-speed disable-auto-speed-detection
```

Configure the Underlay

IN THIS SECTION

In this topology, the IP fabric is only between the two spine switches, as shown in Figure 12 on page 17. The two spine switches establish EBGP peering over the point-to-point links to exchange loopback addresses with each other.

Figure 12: IP Fabric Topology



Configure Spine 1

Step-by-Step Procedure

1. Configure the interfaces on Spine 1.

```
set interfaces et-0/0/50 description "* connected to DC1-Spine2"
set interfaces et-0/0/50 traps
set interfaces et-0/0/50 mtu 9216
set interfaces et-0/0/50 unit 0 family inet address 192.168.100.5/31
set interfaces et-0/0/51 description "* connected to DC1-Spine2"
set interfaces et-0/0/51 traps
set interfaces et-0/0/51 mtu 9216
set interfaces et-0/0/51 unit 0 family inet address 192.168.100.7/31
set interfaces lo0 unit 0 description "** DC1 Spine1 Loopback"
set interfaces lo0 unit 0 family inet address 192.168.255.13/32
```

2. Configure the EBGP underlay.

```
set protocols bgp log-updown
set protocols bgp graceful-restart restart-time 30
set protocols bgp group UNDERLAY type external
set protocols bgp group UNDERLAY description "Connection to EBGP UNDERLAY"
set protocols bgp group UNDERLAY import UNDERLAY-IMPORT
set protocols bgp group UNDERLAY family inet unicast
set protocols bgp group UNDERLAY authentication-key "$ABC123"
set protocols bgp group UNDERLAY export UNDERLAY-EXPORT
set protocols bgp group UNDERLAY local-as 65013
set protocols bgp group UNDERLAY multipath multiple-as
set protocols bgp group UNDERLAY neighbor 192.168.100.4 peer-as 65012
set protocols bgp group UNDERLAY neighbor 192.168.100.6 peer-as 65012
```

3. Configure the import and export policies.

```
set policy-options policy-statement UNDERLAY-EXPORT term LOOPBACK from route-filter
192.168.255.0/24 orlonger
set policy-options policy-statement UNDERLAY-EXPORT term LOOPBACK then accept
set policy-options policy-statement UNDERLAY-EXPORT term DEFAULT then reject
set policy-options policy-statement UNDERLAY-IMPORT term LOOPBACK from route-filter
192.168.255.0/24 orlonger
```

```
set policy-options policy-statement UNDERLAY-IMPORT term LOOPBACK then accept
set policy-options policy-statement UNDERLAY-IMPORT term DEFAULT then reject
```

4. Enable ECMP and ECMP fast reroute protection. Enable per-flow load balancing, which you do with the `per-packet` keyword.

```
set policy-options policy-statement ECMP-POLICY then load-balance per-packet
set routing-options forwarding-table export ECMP-POLICY
```

   If a link goes down, ECMP uses fast reroute protection to shift packet forwarding to operational links, which decreases packet loss. Fast reroute protection updates ECMP sets for the interface without having to wait for the route table to update. When the next route table update occurs, a new ECMP set can be added with fewer links, or the route can point to a single next hop.

```
set routing-options forwarding-table ecmp-fast-reroute
```

5. By default, the ARP aging timer is set at 20 minutes and the MAC aging timer is set at 5 minutes. To avoid synchronization issues with MAC and MAC-IP binding entries in an EVPN-VXLAN environment, configure ARP aging to be faster than MAC aging.

```
set system arp aging-timer 5
set protocols l2-learning global-mac-ip-table-aging-time 300
set protocols l2-learning global-mac-table-aging-time 600
```

Configure Spine 2

Step-by-Step Procedure

Repeat the configuration from Spine 1 on Spine 2.

1. Configure the interfaces on Spine 2.

```
set interfaces et-0/0/50 description "* connected to DC1-Spine1"
set interfaces et-0/0/50 traps
set interfaces et-0/0/50 mtu 9216
set interfaces et-0/0/50 unit 0 family inet address 192.168.100.4/31
set interfaces et-0/0/51 description "* connected to DC1-Spine1"
set interfaces et-0/0/51 traps
set interfaces et-0/0/51 mtu 9216
set interfaces et-0/0/51 unit 0 family inet address 192.168.100.6/31
set interfaces lo0 unit 0 description "** DC1 Spine2 Loopback"
set interfaces lo0 unit 0 family inet address 192.168.255.12/32
```

2. Configure the EBGP underlay.

```
set protocols bgp log-updown
set protocols bgp graceful-restart restart-time 30
set protocols bgp group UNDERLAY type external
set protocols bgp group UNDERLAY description "EBGP UNDERLAY"
set protocols bgp group UNDERLAY import UNDERLAY-IMPORT
set protocols bgp group UNDERLAY family inet unicast
set protocols bgp group UNDERLAY authentication-key "$ABC123"
set protocols bgp group UNDERLAY export UNDERLAY-EXPORT
set protocols bgp group UNDERLAY local-as 65012
set protocols bgp group UNDERLAY multipath multiple-as
set protocols bgp group UNDERLAY neighbor 192.168.100.5 peer-as 65013
set protocols bgp group UNDERLAY neighbor 192.168.100.7 peer-as 65013
```

3. Configure the import and export policies.

```
set policy-options policy-statement UNDERLAY-EXPORT term LOOPBACK from route-filter
192.168.255.0/24 orlonger
set policy-options policy-statement UNDERLAY-EXPORT term LOOPBACK then accept
set policy-options policy-statement UNDERLAY-EXPORT term DEFAULT then reject
set policy-options policy-statement UNDERLAY-IMPORT term LOOPBACK from route-filter
192.168.255.0/24 orlonger
set policy-options policy-statement UNDERLAY-IMPORT term LOOPBACK then accept
set policy-options policy-statement UNDERLAY-IMPORT term DEFAULT then reject
```

4. Enable ECMP and ECMP fast reroute protection.

```
set policy-options policy-statement ECMP-POLICY then load-balance per-packet
set routing-options forwarding-table export ECMP-POLICY
set routing-options forwarding-table ecmp-fast-reroute
```

5. To avoid synchronization issues with MAC and MAC-IP binding entries in an EVPN-VXLAN environment, configure ARP aging to be faster than MAC aging.

```
set system arp aging-timer 5
set protocols l2-learning global-mac-ip-table-aging-time 300
set protocols l2-learning global-mac-table-aging-time 600
```

Verify the Underlay

Step-by-Step Procedure

1. Verify that both BGP neighbor sessions are established on Spine 1.

```
user@spine1> show bgp neighbor 192.168.100.4
Peer: 192.168.100.4+179 AS 65012 Local: 192.168.100.5+51424 AS 65013
Description: Connection to EBGP UNDERLAY
Group: UNDERLAY              Routing-Instance: master
Forwarding routing-instance: master
Type: External    State: Established    Flags: <Sync>
Last State: OpenConfirm   Last Event: RecvKeepAlive
Last Error: Cease
Export: [ UNDERLAY-EXPORT ] Import: [ UNDERLAY-IMPORT ]
. . .
```

```
user@spine1> show bgp neighbor 192.168.100.6
Peer: 192.168.100.6+59705 AS 65012 Local: 192.168.100.7+179 AS 65013
Description: Connection to EBGP UNDERLAY
Group: UNDERLAY              Routing-Instance: master
Forwarding routing-instance: master
Type: External    State: Established    Flags: <Sync>
Last State: OpenConfirm   Last Event: RecvKeepAlive
Last Error: Cease
Export: [ UNDERLAY-EXPORT ] Import: [ UNDERLAY-IMPORT ]
. . .
```

2. Verify that the loopback address of Spine 2 (192.168.255.12) is received by Spine 1 from both BGP neighbor sessions.

```
user@spine1> show route receive-protocol bgp 192.168.100.4
inet.0: 17 destinations, 25 routes (17 active, 0 holddown, 0 hidden)
Restart Complete
  Prefix                  Nexthop              MED     Lclpref    AS path
  * 192.168.255.12/32       192.168.100.4                           65012 I
. . .
```

```
user@spine1> show route receive-protocol bgp 192.168.100.6
inet.0: 17 destinations, 25 routes (17 active, 0 holddown, 0 hidden)
Restart Complete
 Prefix                  Nexthop              MED     Lclpref    AS path
 192.168.255.12/32       192.168.100.6                           65012 I
```

```
user@spine1> show route 192.168.255.12
inet.0: 17 destinations, 25 routes (17 active, 0 holddown, 0 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

192.168.255.12/32  *[BGP/170] 00:39:43, localpref 100, from 192.168.100.4

        AS path: 65012 I, validation-state: unverified
        to 192.168.100.4 via et-0/0/50.0
        >  to 192.168.100.6 via et-0/0/51.0
        [BGP/170] 00:39:43, localpref 100
        AS path: 65012 I, validation-state: unverified
         >  to 192.168.100.6 via et-0/0/51.0
```

3. Ping the loopback of the other spine device from Spine 1.

```
user@spine1> ping 192.168.255.12 source 192.168.255.13
PING 192.168.255.12 (192.168.255.12): 56 data bytes
64 bytes from 192.168.255.12: icmp_seq=0 ttl=64 time=0.746 ms
64 bytes from 192.168.255.12: icmp_seq=1 ttl=64 time=0.699 ms
64 bytes from 192.168.255.12: icmp_seq=2 ttl=64 time=0.784 ms
```

Configure the Overlay

IN THIS SECTION

This section shows how to configure the overlay. It includes IBGP peerings and the VLAN to VXLAN mappings for the virtual networks.

Configure Spine 1

Step-by-Step Procedure

1. Configure IBGP peering between the Spine 1 and Spine 2 loopback addresses.

```
set protocols bgp group EVPN_FABRIC type internal
set protocols bgp group EVPN_FABRIC local-address 192.168.255.13
set protocols bgp group EVPN_FABRIC family evpn signaling
set protocols bgp group EVPN_FABRIC authentication-key "$ABC123"
set protocols bgp group EVPN_FABRIC local-as 65100
set protocols bgp group EVPN_FABRIC multipath
set protocols bgp group EVPN_FABRIC bfd-liveness-detection minimum-interval 1000
set protocols bgp group EVPN_FABRIC bfd-liveness-detection multiplier 3
set protocols bgp group EVPN_FABRIC neighbor 192.168.255.12
set protocols bgp group EVPN_FABRIC vpn-apply-export
```

2. Configure the VLANs and VLAN to VXLAN mapping.

```
set vlans VLAN-201 description "jnpr_1 - bridge domain id 201"
set vlans VLAN-201 vlan-id 201
set vlans VLAN-201 vxlan vni 5201

set vlans VLAN-202 description "jnpr_1 - bridge domain id 202"
set vlans VLAN-202 vlan-id 202
set vlans VLAN-202 vxlan vni 5202

set vlans VLAN-211 description "jnpr_2 - bridge domain id 211"
set vlans VLAN-211 vlan-id 211
set vlans VLAN-211 vxlan vni 5211

set vlans VLAN-212 description "jnpr_2 - bridge domain id 212"
set vlans VLAN-212 vlan-id 212
set vlans VLAN-212 vxlan vni 5212
```

3. Configure the following switch options:

   - The virtual tunnel endpoint (VTEP) source interface. This is the loopback address on Spine 1.

   - The route distinguisher for routes generated by this device.

   - The route target.

   ```
   set switch-options vtep-source-interface lo0.0
   set switch-options route-distinguisher 192.168.255.13:1
   set switch-options vrf-target target:1:999
   set switch-options vrf-target auto
   ```

   The route target configured under `vrf-target` is used by Type 1 EVPN routes. Type 2 and Type 3 EVPN routes use the auto-derived per-VNI route target for export and import.

4. Configure the EVPN protocol. First, configure VXLAN as the data plane encapsulation for EVPN.

   ```
   set protocols evpn encapsulation vxlan
   ```

   Next, configure the VNIs that are part of this EVPN-VXLAN MP-BGP domain. Use `set protocols evpn extended-vni-list all` to configure all VNIs, or configure each VNI separately as shown below.

   ```
   set protocols evpn extended-vni-list 5201
   set protocols evpn extended-vni-list 5202
   set protocols evpn extended-vni-list 5211
   set protocols evpn extended-vni-list 5212
   ```

5. If the data center has only two spine switches that have only BGP neighbor sessions with each other, you must disable core isolation on both spine switches. Otherwise, if a spine switch goes down, the other spine switch loses all BGP neighbor sessions, which places the ToR-facing ports into LACP

standby mode and results in complete traffic loss. See "Split-Brain State" on page 53 and Understanding When to Disable EVPN-VXLAN Core Isolation for more information.

```
set protocols evpn no-core-isolation
```

Configure Spine 2

Step-by-Step Procedure

1. To avoid synchronization issues with MAC and MAC-IP binding entries in an EVPN-VXLAN environment, configure ARP aging to be faster than MAC aging.

```
set system arp aging-timer 5
set protocols l2-learning global-mac-ip-table-aging-time 300
set protocols l2-learning global-mac-table-aging-time 600
```

2. Configure IBGP peering.

```
set protocols bgp group EVPN_FABRIC type internal
set protocols bgp group EVPN_FABRIC local-address 192.168.255.12
set protocols bgp group EVPN_FABRIC family evpn signaling
set protocols bgp group EVPN_FABRIC authentication-key "$ABC123"
set protocols bgp group EVPN_FABRIC local-as 65100
set protocols bgp group EVPN_FABRIC multipath
set protocols bgp group EVPN_FABRIC bfd-liveness-detection minimum-interval 1000
set protocols bgp group EVPN_FABRIC bfd-liveness-detection multiplier 3
set protocols bgp group EVPN_FABRIC neighbor 192.168.255.13
set protocols bgp group EVPN_FABRIC vpn-apply-export
```

3. Configure the VLANs and VLAN to VXLAN mapping.

```
set vlans VLAN-201 description "jnpr_1 - bridge domain id 201"
set vlans VLAN-201 vlan-id 201
set vlans VLAN-201 vxlan vni 5201

set vlans VLAN-202 description "jnpr_1 - bridge domain id 202"
set vlans VLAN-202 vlan-id 202
set vlans VLAN-202 vxlan vni 5202
set vlans VLAN-211 description "jnpr_2 - bridge domain id 211"
set vlans VLAN-211 vlan-id 211
set vlans VLAN-211 vxlan vni 5211

set vlans VLAN-212 description "jnpr_2 - bridge domain id 212"
set vlans VLAN-212 vlan-id 212
set vlans VLAN-212 vxlan vni 5212
```

4. Configure the following switch options.

```
set switch-options vtep-source-interface lo0.0
set switch-options route-distinguisher 192.168.255.12:1
set switch-options vrf-target target:1:999
set switch-options vrf-target auto
```

5. Configure the EVPN protocol.

```
set protocols evpn encapsulation vxlan
```

   Next, configure the VNIs that are part of this EVPN-VXLAN MP-BGP domain. Use `set protocols evpn extended-vni-list all` to configure all VNIs, or configure each VNI separately as shown below.

```
set protocols evpn extended-vni-list 5201
set protocols evpn extended-vni-list 5202
set protocols evpn extended-vni-list 5211
set protocols evpn extended-vni-list 5212
```

6. If the data center has only two spine switches that only have BGP neighbor sessions with each other, you must disable core isolation on both spine switches.

```
set protocols evpn no-core-isolation
```

Verify the Overlay

Step-by-Step Procedure

1. Verify the IBGP peering between Spine 1 and Spine 2 is established.

```
user@spine1> show bgp neighbor 192.168.255.12
Peer: 192.168.255.12+179 AS 65100 Local: 192.168.255.13+62666 AS 65100
  Description: Overlay neighbor with peer
  Group: EVPN_FABRIC           Routing-Instance: master
  Forwarding routing-instance: master
  Type: Internal    State: Established    Flags:<Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: Hold Timer Expired Error
  Options: <Preference LocalAddress HoldTime AuthKey GracefulRestart LogUpDown AddressFamily
Multipath LocalAS Rib-group Refresh>
  Authentication key is configured
  Address families configured: evpn
```

2. Verify the source VTEP for the EVPN domain.

```
user@spine1> show ethernet-switching vxlan-tunnel-end-point source
Logical System Name       Id  SVTEP-IP         IFL   L3-Idx    SVTEP-Mode
    <default>                 0   192.168.255.13   lo0.0    0
    L2-RTT                   Bridge Domain              VNID     MC-Group-IP
    default-switch           VLAN-201+201               5201     0.0.0.0
    default-switch           VLAN-202+202               5202     0.0.0.0
    default-switch           VLAN-211+211               5211     0.0.0.0
    default-switch           VLAN-212+212               5212     0.0.0.0
```

3. Verify all the source VTEP and remote VTEPs.

```
user@spine1> show interfaces vtep
Physical interface: vtep, Enabled, Physical link is Up
  Interface index: 641, SNMP ifIndex: 506
  Type: Software-Pseudo, Link-level type: VxLAN-Tunnel-Endpoint, MTU: Unlimited, Speed:
Unlimited
  Device flags   : Present Running
  Link type      : Full-Duplex
  Link flags     : None
  Last flapped   : Never
    Input packets : 0
    Output packets: 0

  Logical interface vtep.32768 (Index 545) (SNMP ifIndex 548)
    Flags: Up SNMP-Traps 0x4000 Encapsulation: ENET2
    VXLAN Endpoint Type: Source, VXLAN Endpoint Address: 192.168.255.13, L2 Routing Instance:
default-switch, L3 Routing Instance: default
    Input packets : 0
    Output packets: 0

  Logical interface vtep.32769 (Index 560) (SNMP ifIndex 550)
    Flags: Up SNMP-Traps Encapsulation: ENET2
    VXLAN Endpoint Type: Remote, VXLAN Endpoint Address: 192.168.255.12, L2 Routing Instance:
default-switch, L3 Routing Instance: default
    Input packets : 9140
    Output packets: 0
    Protocol eth-switch, MTU: Unlimited
      Flags: Trunk-Mode
```

Configure and Segment Layer 3

IN THIS SECTION

Configure Spine 1

Step-by-Step Procedure

1. Configure routing and forwarding options.

NOTE: Changing routing and forwarding options like `next-hop`, `overlay-ecmp`, or `chained-composite-next-hop` causes the Packet Forwarding Engine to restart, which interrupts all forwarding operations.

- Set the number of next hops to at least the expected number of ARP entries in the overlay. See next-hop (VXLAN Routing) for more information about configuring `vxlan-routing next-hop`.

- Enable two-level equal-cost multipath next hops using the `overlay-ecmp` statement. This statement is required for a Layer 3 EVPN-VXLAN overlay network when pure Type 5 routing is also configured. We strongly recommend that you configure this statement when pure Type 5 routes are enabled.

- The `chained-composite-next-hop` configuration is a must for EVPN pure Type 5 with VXLAN encapsulation. Without this, the PFE will not configure the tunnel next hop.

- Configure the router ID to be the same as the loopback IP address used as the VTEP source and the overlay BGP local address.

```
set forwarding-options vxlan-routing next-hop 32768
set forwarding-options vxlan-routing overlay-ecmp
set routing-options forwarding-table chained-composite-next-hop ingress evpn
set routing-options router-id 192.168.255.13
```

2. To enable the default gateway function, configure IRB interfaces each with a unique IP address and a virtual gateway address (VGA), which must be an anycast IP address. When you specify an IPv4 address for the VGA, the Layer 3 VXLAN gateway automatically generates 00:00:5e:00:01:01 as the MAC address. This example shows you how to manually configure the virtual gateway MAC address. Configure the same virtual gateway MAC address on both spine devices for a given IRB.

NOTE: If the VGA IP address is lower than the IRB IP address, you must use the `preferred` option in the IRB configuration as shown in this example.

```
set interfaces irb unit 201 virtual-gateway-accept-data
set interfaces irb unit 201 description "** L3 interface for VLAN-201 in jnpr_1"
set interfaces irb unit 201 family inet address 192.168.201.3/24 virtual-gateway-address
192.168.201.1
set interfaces irb unit 201 family inet address 192.168.201.3/24 preferred
set interfaces irb unit 201 virtual-gateway-v4-mac 3c:8c:93:2e:20:01
set vlans VLAN-201 l3-interface irb.201
set interfaces irb unit 202 virtual-gateway-accept-data
set interfaces irb unit 202 description "** L3 interface for VLAN-202 in jnpr_1"
set interfaces irb unit 202 family inet address 192.168.202.3/24 virtual-gateway-address
192.168.202.1
set interfaces irb unit 202 family inet address 192.168.202.3/24 preferred
set interfaces irb unit 202 virtual-gateway-v4-mac 3c:8c:93:2e:20:02
set vlans VLAN-202 l3-interface irb.202

set interfaces irb unit 211 virtual-gateway-accept-data
set interfaces irb unit 211 description "** L3 interface for VLAN-211 in jnpr_2"
set interfaces irb unit 211 family inet address 192.168.211.3/24 virtual-gateway-address
192.168.211.1
set interfaces irb unit 211 family inet address 192.168.211.3/24 preferred
set interfaces irb unit 211 virtual-gateway-v4-mac 3c:8c:93:2e:21:11
set vlans VLAN-211 l3-interface irb.211

set interfaces irb unit 212 virtual-gateway-accept-data
set interfaces irb unit 212 description "** L3 interface for VLAN-212 in jnpr_2"
set interfaces irb unit 212 family inet address 192.168.212.3/24 virtual-gateway-address
192.168.212.1
set interfaces irb unit 212 family inet address 192.168.212.3/24 preferred
set interfaces irb unit 212 virtual-gateway-v4-mac 3c:8c:93:2e:21:12
set vlans VLAN-212 l3-interface irb.212
```

3. You will configure the same anycast IRB IP and MAC addresses on the IRB interfaces of each spine device. Because the spine devices act as both the spine and leaf devices in a collapsed spine architecture, they are the only devices that need to know about the IRB interfaces. Disable the advertisement of the IRB interfaces to the other devices.

```
set protocols evpn default-gateway do-not-advertise
```

4. Place the IRBs belonging to the different tenants into their respective routing instances. This allows the IRBs in the same routing instances to share a routing table. As a result, the IRBs in a routing instance can route to each other. IRBs in different routing instances can communicate with each other either through an external security policy enforcer like SRX firewalls or if we explicitly leak routes between the routing instances.

```
set routing-instances JNPR_1_VRF description "VRF for tenant jnpr_1"
set routing-instances JNPR_1_VRF instance-type vrf
set routing-instances JNPR_1_VRF interface irb.201
set routing-instances JNPR_1_VRF interface irb.202
set routing-instances JNPR_1_VRF vrf-table-label
set routing-instances JNPR_1_VRF routing-options multipath



set routing-instances JNPR_2_VRF description "VRF for tenant jnpr_2"
set routing-instances JNPR_2_VRF instance-type vrf
set routing-instances JNPR_2_VRF interface irb.211
set routing-instances JNPR_2_VRF interface irb.212
set routing-instances JNPR_2_VRF vrf-table-label
set routing-instances JNPR_2_VRF routing-options multipath
```

5. Configure Type 5 VNI for the routing instances. When setting up a routing instance for EVPN-VXLAN, you must include a loopback interface and its IP address. If you omit the loopback interface and associated IP address, EVPN control packets cannot be processed.

```
set routing-instances JNPR_1_VRF protocols evpn ip-prefix-routes advertise direct-nexthop
    set routing-instances JNPR_1_VRF protocols evpn ip-prefix-routes encapsulation vxlan
set routing-instances JNPR_1_VRF protocols evpn ip-prefix-routes vni 1101
set routing-instances JNPR_1_VRF protocols evpn ip-prefix-routes export T5_EXPORT

set routing-instances JNPR_2_VRF protocols evpn ip-prefix-routes advertise direct-nexthop
set routing-instances JNPR_2_VRF protocols evpn ip-prefix-routes encapsulation vxlan
set routing-instances JNPR_2_VRF protocols evpn ip-prefix-routes vni 1102
set routing-instances JNPR_2_VRF protocols evpn ip-prefix-routes export T5_EXPORT

set interfaces lo0 unit 1 description "Tenant 1 T5 Loopback"
set interfaces lo0 unit 1 family inet address 192.168.255.21/32
set routing-instances JNPR_1_VRF interface lo0.1

set interfaces lo0 unit 2 description "Tenant 2 T5 Loopback"
set interfaces lo0 unit 2 family inet address 192.168.255.22/32
set routing-instances JNPR_2_VRF interface lo0.2

set policy-options policy-statement T5_EXPORT term 1 from protocol direct
set policy-options policy-statement T5_EXPORT term 1 then accept
set policy-options policy-statement T5_EXPORT term 2 from protocol bgp
set policy-options policy-statement T5_EXPORT term 2 then accept
```

Configure Spine 2

Step-by-Step Procedure

1. Configure routing and forwarding options.

   NOTE: Changing routing and forwarding options like `next-hop`, `overlay-ecmp`, or `chained-composite-next-hop` causes the Packet Forwarding Engine to restart, which interrupts all forwarding operations.

   ```
   set forwarding-options vxlan-routing next-hop 32768
   set forwarding-options vxlan-routing overlay-ecmp
   set routing-options forwarding-table chained-composite-next-hop ingress evpn
   set routing-options router-id 192.168.255.12
   ```

2. Configure IRB.

   ```
   set interfaces irb unit 201 virtual-gateway-accept-data
   set interfaces irb unit 201 description "** L3 interface for VLAN-201 in jnpr_1"
   set interfaces irb unit 201 family inet address 192.168.201.2/24 virtual-gateway-address
   192.168.201.1
   set interfaces irb unit 201 family inet address 192.168.201.2/24 preferred
   set interfaces irb unit 201 virtual-gateway-v4-mac 3c:8c:93:2e:20:01
   set vlans VLAN-201 l3-interface irb.201

   set interfaces irb unit 202 virtual-gateway-accept-data
   set interfaces irb unit 202 description "** L3 interface for VLAN-202 in jnpr_1"
   set interfaces irb unit 202 family inet address 192.168.202.2/24 virtual-gateway-address
   192.168.202.1
   set interfaces irb unit 202 family inet address 192.168.202.2/24 preferred
   set interfaces irb unit 202 virtual-gateway-v4-mac 3c:8c:93:2e:20:02
   set vlans VLAN-202 l3-interface irb.202

   set interfaces irb unit 211 virtual-gateway-accept-data
   set interfaces irb unit 211 description "** L3 interface for VLAN-211 in jnpr_2"
   set interfaces irb unit 211 family inet address 192.168.211.2/24 virtual-gateway-address
   192.168.211.1
   set interfaces irb unit 211 family inet address 192.168.211.2/24 preferred
   set interfaces irb unit 211 virtual-gateway-v4-mac 3c:8c:93:2e:21:11
   ```

```
set vlans VLAN-211 l3-interface irb.211

set interfaces irb unit 212 virtual-gateway-accept-data
set interfaces irb unit 212 description "** L3 interface for VLAN-212 in jnpr_2"
set interfaces irb unit 212 family inet address 192.168.212.2/24 virtual-gateway-address
192.168.212.1
set interfaces irb unit 212 family inet address 192.168.212.2/24 preferred
set interfaces irb unit 212 virtual-gateway-v4-mac 3c:8c:93:2e:21:12
set vlans VLAN-212 l3-interface irb.212
```

3. Since you have configured the same anycast IRB IP and MAC addresses on the IRB interfaces of both spine switches, disable the advertisement of the IRB interfaces to other devices.

```
set protocols evpn default-gateway do-not-advertise
```

4. Place the IRBs belonging to the different tenants into their respective routing instances.

```
set routing-instances JNPR_1_VRF description "VRF for tenant jnpr_1"
set routing-instances JNPR_1_VRF instance-type vrf
set routing-instances JNPR_1_VRF interface irb.201
set routing-instances JNPR_1_VRF interface irb.202
set routing-instances JNPR_1_VRF vrf-table-label
set routing-instances JNPR_1_VRF routing-options multipath

set routing-instances JNPR_2_VRF description "VRF for tenant jnpr_2"
set routing-instances JNPR_2_VRF instance-type vrf
set routing-instances JNPR_2_VRF interface irb.211
set routing-instances JNPR_2_VRF interface irb.212
set routing-instances JNPR_2_VRF vrf-table-label
set routing-instances JNPR_2_VRF routing-options multipath
```

5. Configure Type 5 VNI for the routing instances.

```
set routing-instances JNPR_1_VRF protocols evpn ip-prefix-routes advertise direct-nexthop
set routing-instances JNPR_1_VRF protocols evpn ip-prefix-routes encapsulation vxlan
set routing-instances JNPR_1_VRF protocols evpn ip-prefix-routes vni 1101
set routing-instances JNPR_1_VRF protocols evpn ip-prefix-routes export T5_EXPORT

set routing-instances JNPR_2_VRF protocols evpn ip-prefix-routes advertise direct-nexthop
set routing-instances JNPR_2_VRF protocols evpn ip-prefix-routes encapsulation vxlan
set routing-instances JNPR_2_VRF protocols evpn ip-prefix-routes vni 1102
set routing-instances JNPR_2_VRF protocols evpn ip-prefix-routes export T5_EXPORT

set interfaces lo0 unit 101 description "Tenant 1 T5 Loopback"
set interfaces lo0 unit 101 family inet address 192.168.255.31/32
set routing-instances JNPR_1_VRF interface lo0.101

set interfaces lo0 unit 102 description "Tenant 2 T5 Loopback"
set interfaces lo0 unit 102 family inet address 192.168.255.32/32
set routing-instances JNPR_2_VRF interface lo0.102

set policy-options policy-statement T5_EXPORT term 1 from protocol direct
set policy-options policy-statement T5_EXPORT term 1 then accept
set policy-options policy-statement T5_EXPORT term 2 from protocol bgp
set policy-options policy-statement T5_EXPORT term 2 then accept
```

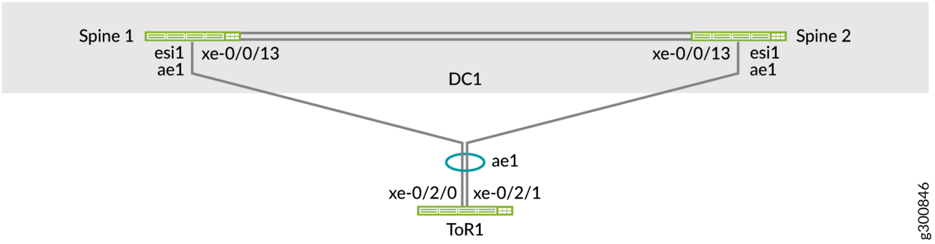
Configure EVPN Multihoming for the ToR Switches

IN THIS SECTION

EVPN multihoming uses ESIs. An ESI is a mandatory attribute that enables EVPN LAG server multihoming. ESI values are encoded as 10-byte integers and are used to identify a multihomed segment. The same ESI value enabled on all spine switches connected to a ToR switch forms an EVPN LAG. This EVPN LAG supports active-active multihoming towards the ToR switch.

The ToR switches (implemented as ToR Virtual Chassis in this example) use a LAG to connect to the two spine switches. As shown in Figure 13 on page 35, ToR1 is connected to the spine switches with LAG ae1. This LAG on the spine switches is enabled by the EVPN multihoming feature.

Figure 13: EVPN Multihoming Configuration for ToR 1



Configure Spine 1

Step-by-Step Procedure

1. By default, aggregated Ethernet interfaces are not created. You must set the number of aggregated Ethernet interfaces on the switch before you can configure them.

```
set chassis aggregated-devices ethernet device-count 15
set interfaces ae1 description "to ToR1"
set interfaces ae1 mtu 9216
```

2. Configure an ESI. Set it the same on both spine switches. Enable all-active modes.

```
set interfaces ae1 esi 00:00:00:00:00:00:00:00:01:01
set interfaces ae1 esi all-active
set interfaces ae1 aggregated-ether-options link-speed 10g
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp periodic fast
```

> **NOTE**: You can also auto-derive ESI. In this example, you manually configure ESI.

3. Configure the LACP system ID. Set it the same on both spine switches to indicate to the ToR switches that uplinks to the two spine switches belong to the same LAG bundle. As a result, the ToR switches places the uplinks to the two spine switches in the same LAG bundle and load shares traffic across the member links.

```
set interfaces ae1 aggregated-ether-options lacp system-id 00:00:00:00:01:01
set interfaces ae1 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae1 unit 0 family ethernet-switching vlan members VLAN-201
set interfaces ae1 unit 0 family ethernet-switching vlan members VLAN-202
set interfaces ae1 unit 0 family ethernet-switching vlan members VLAN-211
set interfaces ae1 unit 0 family ethernet-switching vlan members VLAN-212
```

4. Configure the physical interface on Spine 1 connected to ToR 1 as a member of the ae1 LAG.

```
set interfaces xe-0/0/13 ether-options 802.3ad ae1
```

Configure Spine 2

Step-by-Step Procedure

1. Set the number of aggregated Ethernet interfaces on the switch.

```
set chassis aggregated-devices ethernet device-count 15
set interfaces ae1 description "to ToR1"
set interfaces ae1 mtu 9216
```

2. Configure an ESI. Set it the same on both spine switches. Enable all-active modes.

```
set interfaces ae1 esi 00:00:00:00:00:00:00:00:01:01
set interfaces ae1 esi all-active
set interfaces ae1 aggregated-ether-options link-speed 10g
set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 aggregated-ether-options lacp periodic fast
```

3. Configure the LACP system ID. Set it the same on both spine switches.

```
set interfaces ae1 aggregated-ether-options lacp system-id 00:00:00:00:01:01
set interfaces ae1 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae1 unit 0 family ethernet-switching vlan members VLAN-201
set interfaces ae1 unit 0 family ethernet-switching vlan members VLAN-202
set interfaces ae1 unit 0 family ethernet-switching vlan members VLAN-211
set interfaces ae1 unit 0 family ethernet-switching vlan members VLAN-212
```

4. Configure the physical interface on Spine 2 connected to ToR 1 as a member of the ae1 LAG.

```
set interfaces xe-0/0/13 ether-options 802.3ad ae1
```

Configure ToR 1

Step-by-Step Procedure

1. By default, aggregated Ethernet interfaces are not created. You must set the number of aggregated Ethernet interfaces on the switch before you can configure them.

```
set chassis aggregated-devices ethernet device-count 4
```

2. Configure the aggregated Ethernet interfaces.

```
set interfaces xe-0/2/0 ether-options 802.3ad ae1
set interfaces xe-0/2/1 ether-options 802.3ad ae1

set interfaces ae1 aggregated-ether-options lacp active
set interfaces ae1 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae1 unit 0 family ethernet-switching vlan members VLAN-201
set interfaces ae1 unit 0 family ethernet-switching vlan members VLAN-202
set interfaces ae1 unit 0 family ethernet-switching vlan members VLAN-211
set interfaces ae1 unit 0 family ethernet-switching vlan members VLAN-212
```

3. Configure the VLANs.

```
set vlans VLAN-201 vlan-id 201
set vlans VLAN-202 vlan-id 202
set vlans VLAN-211 vlan-id 211
set vlans VLAN-212 vlan-id 212
```

Verify EVPN Multihoming

Step-by-Step Procedure

1. Check the status of ae1 and the ESI associated with the LAG.

```
user@spine1> show interfaces ae1
Physical interface: ae1, Enabled, Physical link is Up
  Interface index: 689, SNMP ifIndex: 552
  Description: to ToR1
  Link-level type: Ethernet, MTU: 9216, Speed: 10Gbps, BPDU Error: None, Ethernet-Switching
Error: None, MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled, Flow
control: Disabled, Minimum links needed: 1,
  Minimum bandwidth needed: 1bps
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Current address: 3c:8c:93:2e:a9:80, Hardware address: 3c:8c:93:2e:a9:80
  Ethernet segment value: 00:00:00:00:00:00:00:00:01:01, Mode: all-active
  Last flapped   : 2019-11-10 14:50:49 PST (00:26:56 ago)
  Input rate     : 624 bps (0 pps)
  Output rate    : 936 bps (1 pps)
  ...
```

2. Verify that the members of ae1 are collecting and distributing.

```
user@spine1> show lacp interfaces ae1
Aggregated interface: ae1
    LACP state:       Role   Exp   Def  Dist  Col  Syn  Aggr  Timeout  Activity
      xe-0/0/13      Actor    No    No   Yes  Yes  Yes   Yes     Fast    Active
      xe-0/0/13    Partner    No    No   Yes  Yes  Yes   Yes     Fast    Active
    LACP protocol:        Receive State  Transmit State          Mux State
      xe-0/0/13                 Current   Fast periodic Collecting distributing
```

3. Verify the status of EVPN Multihoming in the EVPN instance is `Resolved` on Spine 1. You can also see which spine switch is the designated forwarder for BUM traffic.

```
user@spine1> show evpn instance extensive
Instance: __default_evpn__
  Route Distinguisher: 192.168.255.13:0
  Number of bridge domains: 0
  Number of neighbors: 1
    Address               MAC    MAC+IP        AD        IM        ES Leaf-label
    192.168.255.12          0         0         0         0         2

Instance: default-switch
  Route Distinguisher: 192.168.255.13:1
  Encapsulation type: VXLAN
  Duplicate MAC detection threshold: 5
  Duplicate MAC detection window: 180
  MAC database status                     Local  Remote
    MAC advertisements:                       6      10
    MAC+IP advertisements:                   10      10
    Default gateway MAC advertisements:       8       0
  Number of local interfaces: 5 (3 up)
    Interface name  ESI                            Mode             Status     AC-Role
    .local..6       00:00:00:00:00:00:00:00:00:00  single-homed     Up         Root
    ae1.0           00:00:00:00:00:00:00:00:01:01  all-active       Up         Root

...

Number of neighbors: 1
    Address               MAC    MAC+IP        AD        IM        ES Leaf-label
    192.168.255.12         10        10         8         4         0
  Number of ethernet segments: 10
    ESI: 00:00:00:00:00:00:00:00:01:01
      Status: Resolved by IFL ae1.0
      Local interface: ae1.0, Status: Up/Forwarding
      Number of remote PEs connected: 1
        Remote PE        MAC label  Aliasing label  Mode
        192.168.255.12   5212       0               all-active
      DF Election Algorithm: MOD based
      Designated forwarder: 192.168.255.13
      Backup forwarder: 192.168.255.12
      Last designated forwarder update: Nov 10 14:50:49
```

4. Verify that all member links of the ae1 interface are collecting and distributing on ToR 1.

```
user@tor1> show lacp interfaces
Aggregated interface: ae1
    LACP state:       Role   Exp   Def  Dist  Col  Syn  Aggr  Timeout  Activity
      xe-0/2/0       Actor    No    No   Yes  Yes  Yes   Yes     Fast    Active
      xe-0/2/0     Partner    No    No   Yes  Yes  Yes   Yes     Fast    Active
```

```
  xe-0/2/1       Actor    No    No   Yes  Yes  Yes   Yes     Fast    Active
  xe-0/2/1     Partner    No    No   Yes  Yes  Yes   Yes     Fast    Active
LACP protocol:        Receive State  Transmit State          Mux State
  xe-0/2/0                  Current   Fast periodic Collecting distributing
  xe-0/2/1                  Current   Fast periodic Collecting distributing
```

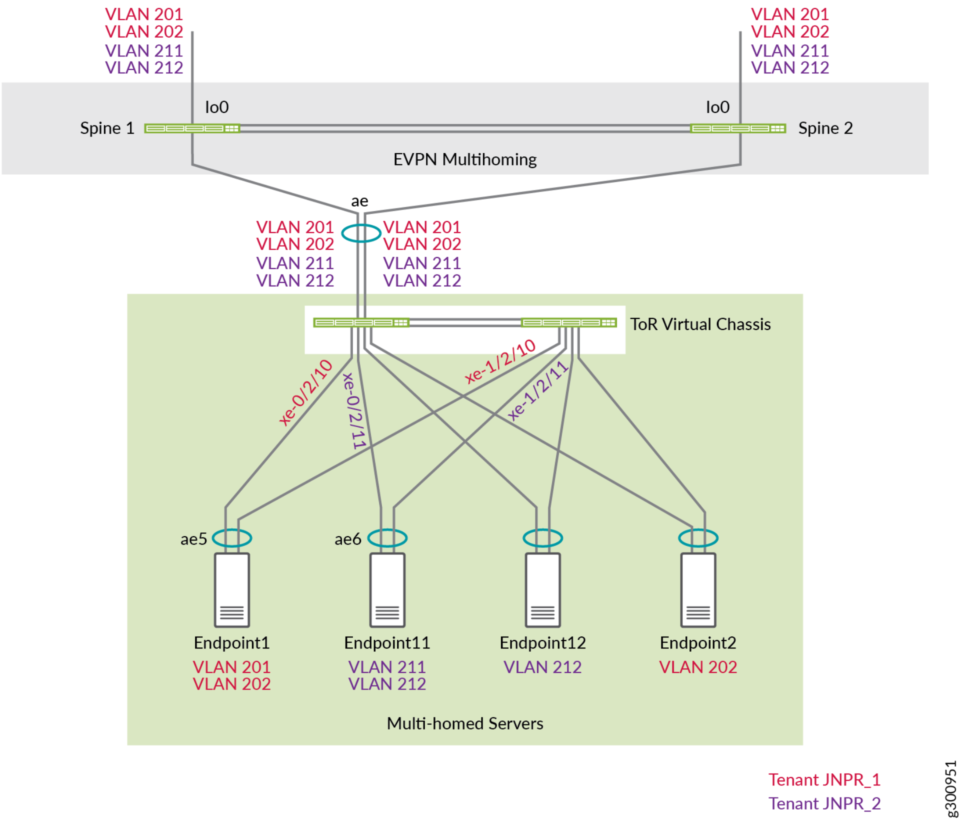
Configure Multihoming for the Servers

IN THIS SECTION

Multihome the servers to the ToR Virtual Chassis for redundancy and load sharing. The servers use LAG to connect to the two ToR Virtual Chassis member switches.

As shown in Figure 14 on page 41, Endpoint 1 is connected to the ToR Virtual Chassis through LAG ae5 and belongs to the JNPR_1 tenant. Endpoint 11 is connected to the ToR Virtual Chassis through LAG ae6 and belongs to the JNPR_2 tenant.

Figure 14: Multihomed Server Topology



Configure ToR 1

Step-by-Step Procedure

Since the ToR switches are configured in a Virtual Chassis, you only need to commit the configuration on the primary switch. In this example, ToR 1 is the primary switch.

1. Configure LAG on the interfaces connected to Endpoint 1: interface xe-0/2/10 on ToR 1 and interface xe-1/2/10 on ToR 2. Endpoint 1 belongs to VLANs 201 and 202.

```
set interfaces xe-0/2/10 ether-options 802.3ad ae5
set interfaces xe-1/2/10 ether-options 802.3ad ae5

set interfaces ae5 aggregated-ether-options lacp active
set interfaces ae5 description "Connected to Endpoint1"
set interfaces ae5 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae5 unit 0 family ethernet-switching vlan members VLAN-201
set interfaces ae5 unit 0 family ethernet-switching vlan members VLAN-202
```

2. Configure LAG on the interfaces connected to Endpoint 11. Endpoint 11 belongs to VLANs 211 and 212.

```
set interfaces xe-0/2/11 ether-options 802.3ad ae6
set interfaces xe-1/2/11 ether-options 802.3ad ae6

set interfaces ae6 aggregated-ether-options lacp active
set interfaces ae6 description "Connected to Endpoint11"
set interfaces ae6 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae6 unit 0 family ethernet-switching vlan members VLAN-211
set interfaces ae6 unit 0 family ethernet-switching vlan members VLAN-212
```

Verify Server Connectivity

IN THIS SECTION

Use this section to verify the servers are connected to each other through the ToR and spine switches. How you do this depends on whether they are part of the same VLAN or two different VLANs.

NOTE: We recommend multihoming your servers to the ToR switches for redundancy and load sharing as described in the previous section. This section shows single-homed servers for simplicity.

Verify Intra-VLAN Server Connectivity

Step-by-Step Procedure

1. Verify the MAC addresses of both endpoints appear in the Ethernet switching table on both the ToR switches.

```
user@tor1> show ethernet-switching table
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static, C -
Control MAC
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC)



Ethernet switching table : 4 entries, 4 learned
Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
NH        RTR
    name                address             flags           interface
Index     ID
    VLAN-201            f4:b5:2f:40:9f:01   D             -   ae1.0
0         0
    VLAN-202            00:10:94:00:01:01   D             -   xe-0/2/2.0
0         0
    VLAN-202            00:10:94:00:01:02   D             -   ae1.0
0         0
    VLAN-202            3c:8c:93:2e:a8:c0   D             -   ae1.0
0         0
```

```
user@tor2> show ethernet-switching table
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static, C -
Control MAC
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC)



Ethernet switching table : 4 entries, 4 learned
Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
NH        RTR
    name                address             flags           interface
Index     ID
```

```
    VLAN-201            f4:b5:2f:40:9f:01   D              -   ae1.0
0         0
    VLAN-202            00:10:94:00:01:01   D              -   xe-0/2/2.0
0         0
    VLAN-202            00:10:94:00:01:02   D              -   ae1.0
0         0
    VLAN-202            3c:8c:93:2e:a8:c0   D              -   ae1.0
0         0
```

2. Verify that the two MAC addresses appear in the Ethernet Switching table on both the spine switches. The two MAC addresses are learned from the ToR switches over the LAG (ae1 and ae2) connected to each ToR switch. The MAC flags `DL`, `DR`, and `DLR` indicate whether traffic for the MAC address was learned locally by the spine switch, by the remote spine switch, or by both the spine switches.

```
user@spine1> show ethernet-switching table vlan-id 202
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC)


Ethernet switching table : 4 entries, 4 learned
Routing instance : default-switch
   Vlan                MAC                 MAC      Logical                Active
   name                address             flags    interface              source
   VLAN-202            00:00:5e:00:01:01   DR       esi.1723
05:00:00:fe:4c:00:00:14:52:00
   VLAN-202            00:10:94:00:01:01   DR       ae1.0
   VLAN-202            00:10:94:00:01:02   DL       ae2.0
   VLAN-202            3c:8c:93:2e:da:c0   D        vtep.32769             192.168.255.12
```

```
user@spine2> show ethernet-switching table vlan-id 202
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC)


Ethernet switching table : 4 entries, 4 learned
Routing instance : default-switch
   Vlan                MAC                 MAC      Logical                Active
   name                address             flags    interface              source
   VLAN-202            00:00:5e:00:01:01   DR       esi.1723
```

```
05:00:00:fe:4c:00:00:14:52:00
   VLAN-202            00:10:94:00:01:01   DR       ae1.0
   VLAN-202            00:10:94:00:01:02   DL       ae2.0
   VLAN-202            3c:8c:93:2e:da:c0   D        vtep.32769             192.168.255.12
```

3. Verify the first MAC address is in the EVPN database on Spine 1. This output indicates that the MAC address was learned locally by this spine switch over the ESI 00:00:00:00:00:00:00:00:01:02 and LAG ae2. This MAC address is advertised in EVPN to the other spine switch.

```
user@spine1> show evpn database mac-address 00:10:94:00:01:02 extensive
Instance: default-switch

VN Identifier: 5202, MAC address: 00:10:94:00:01:02
  State: 0x0
  Source: 00:00:00:00:00:00:00:00:01:02, Rank: 1, Status: Active
    Local origin: ae2.0
    Mobility sequence number: 0 (minimum origin address 192.168.255.13)
    Timestamp: Nov 10 16:48:41 (0x5dc8afe9)
    State: <Local-MAC-Only Local-To-Remote-Adv-Allowed>
    MAC advertisement route status: Created
    History db:
      Time                  Event
      Nov 10 16:48:41 2019  Updating output state (change flags 0x20 <ESI-Added>)
      Nov 10 16:48:41 2019  Active ESI changing (not assigned ->
00:00:00:00:00:00:00:00:01:02)
      Nov 10 16:48:41 2019  Creating all output state
      Nov 10 16:48:41 2019  Creating MAC advertisement route
      Nov 10 16:48:41 2019  Adding to instance ESI list
      Nov 10 16:48:41 2019  Clearing change flags <ESI-Added>
      Nov 10 16:48:41 2019  Clearing change flags <Intf ESI-Local-State>
      Nov 10 16:48:42 2019  Updating output state (change flags 0x0)
      Nov 10 16:48:42 2019  Active ESI unchanged (00:00:00:00:00:00:00:00:01:02)
      Nov 10 16:48:42 2019  Updating output state (change flags 0x0)
```

4. Verify the second MAC address is in the EVPN database on Spine 1. This MAC address was learned by the remote spine switch and advertised to the local spine switch over EVPN. This output also shows that this MAC address is mapped to ESI 00:00:00:00:00:00:00:00:01:01. Traffic destined for this MAC address can be switched locally to ToR 1 using the same Ethernet segment.

```
user@spine1> show evpn database mac-address 00:10:94:00:01:01 extensive
Instance: default-switch
```

```
VN Identifier: 5202, MAC address: 00:10:94:00:01:01
  State: 0x0
  Source: 00:00:00:00:00:00:00:00:01:01, Rank: 1, Status: Active
    Remote origin: 192.168.255.12
    Mobility sequence number: 0 (minimum origin address 192.168.255.12)
    Timestamp: Nov 10 16:48:41 (0x5dc8afe9)
    State: <Remote-To-Local-Adv-Done>
    MAC advertisement route status: Not created (no local state present)
    History db:
      Time                  Event
      Nov 10 16:48:41 2019  Adding to instance ESI list
      Nov 10 16:48:41 2019  Clearing change flags <ESI-Added>
      Nov 10 16:48:41 2019  Clearing change flags <ESI-Peer-Added ESI-Remote-Peer-Com-Chg>
      Nov 10 16:48:42 2019  Updating output state (change flags 0x0)
      Nov 10 16:48:42 2019  Active ESI unchanged (00:00:00:00:00:00:00:00:01:01)
      Nov 10 16:48:42 2019  Updating output state (change flags 0x0)
      Nov 10 16:48:42 2019  Advertisement route cannot be created (no local state present)
      Nov 10 16:48:42 2019  ESI 00:00:00:00:00:00:00:00:01:01, peer 192.168.255.12 per-ES AD
route not rcvd, remote peer found
      Nov 10 16:48:42 2019  Sent MAC add with NH 0, interface ae1.0 (index 0), RTT 6, remote
addr 192.168.255.12, ESI 0101, VLAN 0, VNI 5202, flags 0x0, timestamp 0x5dc8afe9 to L2ALD
      Nov 10 16:48:42 2019  Sent peer 192.168.255.12 record created
```

5. Verify the EVPN routes on Spine 1. This output shows that these MAC addresses are advertised by the spine switches as BGP routes.

```
user@spine1> show route table bgp.evpn.0 evpn-mac-address 00:10:94:00:01:01
bgp.evpn.0: 75 destinations, 75 routes (75 active, 0 holddown, 0 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

2:192.168.255.13:1::5202::00:10:94:00:01:01/304 MAC/IP
                   *[EVPN/170] 00:01:52
                       Indirect
user@spine1> show route table bgp.evpn.0 evpn-mac-address 00:10:94:00:01:02
bgp.evpn.0: 75 destinations, 75 routes (75 active, 0 holddown, 0 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

2:192.168.255.13:1::5202::00:10:94:00:01:02/304 MAC/IP
```

```
                    *[EVPN/170] 00:02:02
                        Indirect
```

6. Verify the EVPN routes on Spine 2. This output shows the BGP routes received the IBGP peering with Spine 1. Let us look at these routes in detail.

```
user@spine2> show route receive-protocol bgp 192.168.255.13
inet.0: 13 destinations, 14 routes (13 active, 0 holddown, 0 hidden)
Restart Complete

JNPR_1_VRF.inet.0: 9 destinations, 11 routes (9 active, 0 holddown, 0 hidden)

:vxlan.inet.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
Restart Complete

JNPR_2_VRF.inet.0: 9 destinations, 11 routes (9 active, 0 holddown, 0 hidden)

mpls.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
Restart Complete

inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Restart Complete

JNPR_1_VRF.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

JNPR_2_VRF.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

bgp.evpn.0: 75 destinations, 75 routes (75 active, 0 holddown, 0 hidden)
Restart Complete
  Prefix                  Nexthop              MED     Lclpref    AS path
  1:192.168.255.13:0::0101::FFFF:FFFF/192 AD/ESI
*                         192.168.255.13               100        I
  1:192.168.255.13:0::0102::FFFF:FFFF/192 AD/ESI
*                         192.168.255.13               100        I
...
  1:192.168.255.13:0::050000fe4c0000145c00::FFFF:FFFF/192 AD/ESI
*                         192.168.255.13               100        I
  1:192.168.255.13:1::0101::0/192 AD/EVI
*                         192.168.255.13               100        I
  1:192.168.255.13:1::0102::0/192 AD/EVI
```

```
*                          192.168.255.13               100        I
...
```

The two Type 1 routes emphasized above show that Spine 1 is connected to two Ethernet Segments (ES). The ESI numbers are 0101 and 0102.

```
...
  2:192.168.255.13:1::5202::00:00:5e:00:01:01/304 MAC/IP
*                          192.168.255.13               100        I
  2:192.168.255.13:1::5202::00:10:94:00:01:01/304 MAC/IP
*                          192.168.255.13               100        I
  2:192.168.255.13:1::5202::00:10:94:00:01:02/304 MAC/IP
*                          192.168.255.13               100        I
...
```

These two routes are Type 2 routes shown above are advertised by Spine 1. They show that the two MAC addresses are reachable from Spine 1.

7. Verify the control plane for the following MAC addresses on Spine 1.

```
user@spine1> show route table bgp.evpn.0 evpn-mac-address 00:10:94:00:01:01
bgp.evpn.0: 78 destinations, 78 routes (78 active, 0 holddown, 0 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

2:192.168.255.13:1::5202::00:10:94:00:01:01/304 MAC/IP
                   *[EVPN/170] 00:11:49
                       Indirect
```

```
user@spine1> show route table bgp.evpn.0 evpn-mac-address 00:10:94:00:01:02
bgp.evpn.0: 78 destinations, 78 routes (78 active, 0 holddown, 0 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

2:192.168.255.13:1::5202::00:10:94:00:01:02/304 MAC/IP
                   *[EVPN/170] 00:11:52
                       Indirect
```

8. Verify the forwarding table entries for these MAC addresses on Spine 1. The following output shows that the local aggregated Ethernet interface is used for switching traffic destined for these MAC addresses.

```
user@spine1> show route forwarding-table destination 00:10:94:00:01:01
Routing table: default-switch.bridge
Bridging domain: VLAN-202.bridge
VPLS:
Enabled protocols: Bridging, ACKed by all peers,
Destination        Type RtRef Next hop           Type Index    NhRef Netif
00:10:94:00:01:01/48 user     0                  ucst     1710     7 ae1.0
```

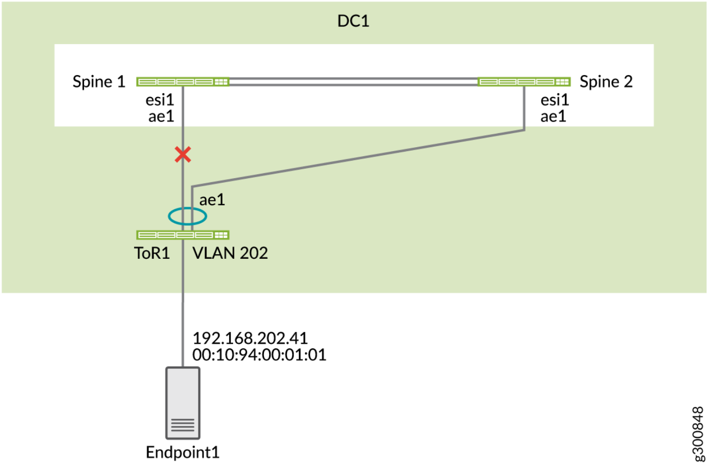
```
user@spine1> show route forwarding-table destination 00:10:94:00:01:02
Routing table: default-switch.bridge
Bridging domain: VLAN-202.bridge
VPLS:
Enabled protocols: Bridging, ACKed by all peers,
Destination        Type RtRef Next hop           Type Index    NhRef Netif
00:10:94:00:01:02/48 user     0                  ucst     1754     9 ae2.0
```

9. Test what happens when an uplink fails. If an uplink from ToR 1 fails, the output shows that the state at that interface is `Detached`.

```
user@spine1> show lacp interfaces ae1
Aggregated interface: ae1
    LACP state:       Role    Exp    Def  Dist   Col   Syn   Aggr  Timeout  Activity
      xe-0/0/13       Actor     No    Yes     No    No    No    Yes     Fast    Active
      xe-0/0/13     Partner     No    Yes     No    No    No    Yes     Fast   Passive
    LACP protocol:        Receive State  Transmit State          Mux State
      xe-0/0/13           Port disabled     No periodic           Detached
```

Figure 15 on page 50 shows the topology when the interface connected to ToR 1 on Spine 1 is down.

Figure 15: Topology When Uplink Fails



Verify that Spine 1 is now learning this MAC address from Spine 2 since Spine 1 does not have a direct connection to ToR 1.

```
user@spine1> show route table bgp.evpn.0 evpn-mac-address 00:10:94:00:01:01
bgp.evpn.0: 76 destinations, 76 routes (76 active, 0 holddown, 0 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

2:192.168.255.12:1::5202::00:10:94:00:01:01/304 MAC/IP
                   *[BGP/170] 00:01:05, localpref 100, from 192.168.255.12
                      AS path: I, validation-state: unverified
                       to 192.168.100.4 via et-0/0/50.0
                    >  to 192.168.100.6 via et-0/0/51.0
```

The forwarding table details on Spine 1 show that the traffic destined for this MAC address is sent to Spine 2.

```
user@spine1> show route forwarding-table destination 00:10:94:00:01:01 extensive
Routing table: default-switch.bridge [Index 6]
Bridging domain: VLAN-202.bridge [Index 6]
VPLS:
Enabled protocols: Bridging, ACKed by all peers,

Destination:  00:10:94:00:01:01/48
  Learn VLAN: 0                        Route type: user
  Route reference: 0                   Route interface-index: 560
```

```
Multicast RPF nh index: 0
P2mpidx: 0
IFL generation: 514                  Epoch: 0
Sequence Number: 0                   Learn Mask: 0x4000000000000000010000000000000000000000
L2 Flags: control_dyn
Flags: sent to PFE
Nexthop:
Next-hop type: composite             Index: 1724     Reference: 26
Next-hop type: indirect              Index: 524289   Reference: 3
Next-hop type: unilist               Index: 524288   Reference: 6
Nexthop: 192.168.100.4
Next-hop type: unicast               Index: 1708     Reference: 4
Next-hop interface: et-0/0/50.0      Weight: 0x0
Nexthop: 192.168.100.6
Next-hop type: unicast               Index: 1709     Reference: 4
Next-hop interface: et-0/0/51.0      Weight: 0x0
```

Verify Inter-VLAN Server Connectivity

Step-by-Step Procedure

1. On Spine 1, verify that the two MAC addresses are in different VLANs.

```
user@spine1> show ethernet-switching table | match 00:10:94:00:11:11
   VLAN-201            00:10:94:00:11:11   DLR      ae1.0
```

```
user@spine1> show ethernet-switching table | match 00:10:94:00:01:02
   VLAN-202            00:10:94:00:01:02   DL       ae2.0
```

2. On Spine 1, verify the ARP resolution for the two endpoints.

```
user@spine1> show arp no-resolve | match 00:10:94:00:11:11
00:10:94:00:11:11 192.168.201.41  irb.201 [ae1.0]         permanent remote
```

```
user@spine1> show arp no-resolve | match 00:10:94:00:01:02
00:10:94:00:01:02 192.168.202.42  irb.202 [ae2.0]         permanent remote
```

3. On Spine 1, check the control plane learning for the MAC address 00:10:94:00:11:11. You can see that there is a MAC route for the MAC address and a MAC/IP route for this MAC address.

```
user@spine1> show route table bgp.evpn.0 evpn-mac-address 00:10:94:00:11:11
bgp.evpn.0: 82 destinations, 82 routes (82 active, 0 holddown, 0 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

2:192.168.255.12:1::5201::00:10:94:00:11:11/304 MAC/IP
                   *[BGP/170] 00:08:43, localpref 100, from 192.168.255.12
                      AS path: I, validation-state: unverified
                       to 192.168.100.4 via et-0/0/50.0
                    >  to 192.168.100.6 via et-0/0/51.0
2:192.168.255.13:1::5201::00:10:94:00:11:11/304 MAC/IP
                   *[EVPN/170] 00:09:01
                       Indirect
2:192.168.255.12:1::5201::00:10:94:00:11:11::192.168.201.41/304 MAC/IP
                   *[BGP/170] 00:08:43, localpref 100, from 192.168.255.12
                      AS path: I, validation-state: unverified
                       to 192.168.100.4 via et-0/0/50.0
                    >  to 192.168.100.6 via et-0/0/51.0
2:192.168.255.13:1::5201::00:10:94:00:11:11::192.168.201.41/304 MAC/IP
                   *[EVPN/170] 00:09:01
                       Indirect
```

4. Verify the forwarding table entries for these MAC addresses. Since Spine 1 is connected to both ToR switches locally, the traffic is switched locally to the corresponding ToR switch from Spine 1.

```
user@spine1> show route forwarding-table destination 00:10:94:00:11:11
Routing table: default-switch.bridge
Bridging domain: VLAN-201.bridge
VPLS:
Enabled protocols: Bridging, ACKed by all peers,
Destination        Type RtRef Next hop           Type Index    NhRef Netif
00:10:94:00:11:11/48 user     0                  ucst     1710     8 ae1.0
```

```
user@spine1> show route forwarding-table destination 00:10:94:00:01:02
Routing table: default-switch.bridge
Bridging domain: VLAN-202.bridge
VPLS:
```

```
Enabled protocols: Bridging, ACKed by all peers,
Destination        Type RtRef Next hop           Type Index    NhRef Netif
00:10:94:00:01:02/48 user     0                  ucst     1754    10 ae2.0
```

What's Next

You have configured and verified a collapsed spine architecture for your first data center. If needed, repeat the configuration on the devices in the second data center.

Go to the next page to configure advanced security and connect your data centers.

Split-Brain State

IN THIS SECTION

How to Prevent a Split-Brain State

Problem

If the links between the spine switches are down, causing the BGP peering to go down, both spine switches are active and forwarding. The downstream aggregated Ethernet interfaces are active and forwarding. This scenario is known as a split-brain state and can cause multiple problems.

Solution

To prevent this issue from occurring, choose one spine switch to be the standby switch.

We also recommend:

- Using at least two links between the spine switches. This makes it less likely all the links between the spine switches will go down.

- Multihoming all servers. If there is a single-homed server on one of the spine switches, the server could be unreachable.

What's Next

You have configured and verified a collapsed spine architecture for your first data center. If needed, repeat the configuration on the devices in the second data center.

Go to the next page to configure advanced security and connect your data centers.

Advanced Security and Data Center Interconnect Configurations

IN THIS SECTION

Use these examples to configure advanced security and DCI on your collapsed spine data center architecture.

Configure Advanced Security for Inter-Tenant Traffic

IN THIS SECTION

The SRX Series is a next-generation firewall that can provide advanced security services for inter-tenant traffic. Use this section to route inter-tenant traffic between JNPR_1 and JNPR_2 in DC1 through the SRX chassis cluster.

Requirements

- The devices that you configured in "How to Configure a Collapsed Spine with EVPN Multihoming" on page 12.
- The SRX chassis cluster must already be configured and running. See Configuring Chassis Clustering on SRX Series Devices for details about enabling a SRX chassis cluster.

Overview

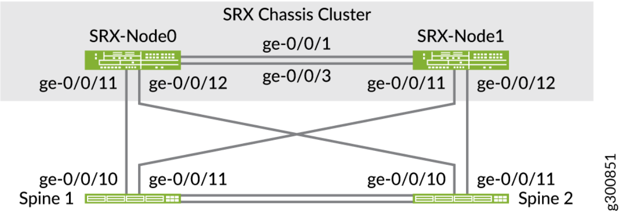
IN THIS SECTION

The SRX Series Firewalls in your chassis cluster operate as a single device to provide device, interface, and service-level redundancy. Use this section to separate the chassis cluster into zones and configure the routing policies so that the correct traffic is routed through the security devices.

Topology

Both spine switches are physically connected to both SRX nodes as shown in Figure 16 on page 55.

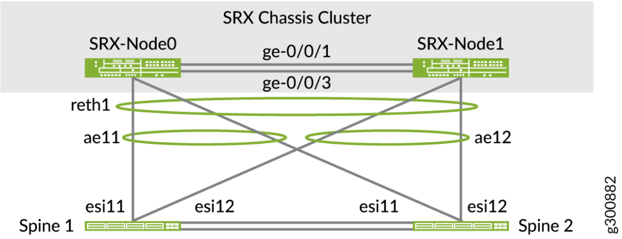
Figure 16: Physical Topology of SRX Cluster



NOTE: This example is based on SRX345 devices. Once placed into a HA cluster, the interfaces on node 1 are associated with FPC slot 5. This means the ge-0/0/11 interface shown for node 1 is actually configured as ge-5-0/11 once the cluster is formed. The FPC number for node 1 in a HA cluster can vary by SRX model type.

Reth1 is a logical interface in the SRX cluster. It is active on one of the nodes of the SRX cluster. If the primary node or interconnect link between the SRX devices and the spine switches fails, Reth1 will failover to the secondary node. Figure 17 on page 56 shows the logical interfaces between the SRX devices and the spine switches.

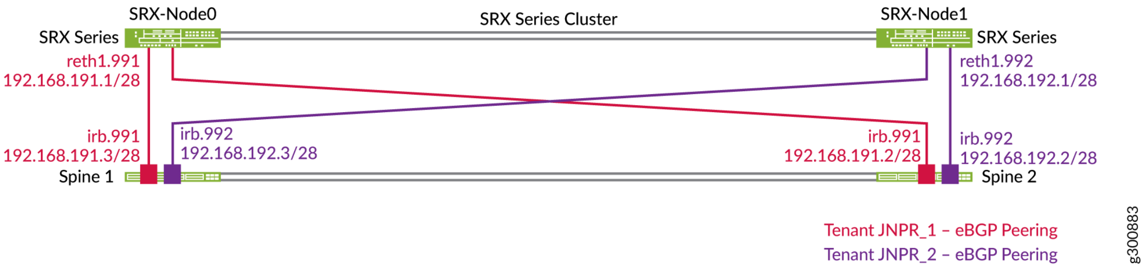
Figure 17: Overlay Topology of SRX Cluster



Each spine switch establishes separate EBGP peerings with the SRX cluster in each routing instance or tenant as shown in Figure 18 on page 56. For example, Spine 1 has two peerings with the SRX cluster, one in each routing instance: JNPR_1 and JNPR_2. Reth1.991 peers with the JNPR_1 routing instance on the spine switches and belongs to the JNPR_1 security zone. Reth1.992 peers with the JNPR_2 routing instance on the spine switches and belongs to the JNPR_2 security zone.

The SRX Series Firewall advertises a summary route that covers all prefixes (for example, 192.168.0.0/16). The spine switches advertise specific subnets in each routing instance.

Figure 18: Topology of SRX Cluster with EBGP Peering



Configure the Interfaces

IN THIS SECTION

Configure the SRX Device

Step-by-Step Procedure

1. Configure the group for the logical interfaces on the SRX device.

```
set chassis cluster reth-count 3
set chassis cluster redundancy-group 0 node 0 priority 100
set chassis cluster redundancy-group 0 node 1 priority 1
set chassis cluster redundancy-group 1 node 0 priority 200
set chassis cluster redundancy-group 1 node 1 priority 100
set chassis cluster redundancy-group 1 preempt
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/11 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/12 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-5/0/11 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-5/0/12 weight 255
```

2. Configure the logical interfaces. Reth1 is a tagged Layer 3 interface on the SRX cluster. Reth1.991 peers with the JNPR_1 routing instance on the spine switches. Reth1.992 peers with the JNPR_2 routing instance on the spine switches.

```
set interfaces reth1 vlan-tagging
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 redundant-ether-options lacp active
set interfaces reth1 redundant-ether-options lacp periodic fast
set interfaces reth1 unit 991 description "Spine Interconnect for JNPR_1"
set interfaces reth1 unit 991 vlan-id 991
set interfaces reth1 unit 991 family inet address 192.168.191.1/28
set interfaces reth1 unit 992 description "Spine Interconnect for JNPR_2"
set interfaces reth1 unit 992 vlan-id 992
set interfaces reth1 unit 992 family inet address 192.168.192.1/28

set interfaces ge-0/0/11 description "To Spine1 | ge-0/0/10"
set interfaces ge-0/0/11 gigether-options no-auto-negotiation
set interfaces ge-0/0/11 gigether-options redundant-parent reth1

set interfaces ge-5/0/11 description "To Spine1 | ge-0/0/11"
set interfaces ge-5/0/11 gigether-options no-auto-negotiation
set interfaces ge-5/0/11 gigether-options redundant-parent reth1

set interfaces ge-0/0/12 description "To Spine2 | ge-0/0/10"
set interfaces ge-0/0/12 gigether-options no-auto-negotiation
set interfaces ge-0/0/12 gigether-options redundant-parent reth1

set interfaces ge-5/0/12 description "To Spine2 | ge-0/0/11"
set interfaces ge-5/0/12 gigether-options no-auto-negotiation
set interfaces ge-5/0/12 gigether-options redundant-parent reth1
```

3. Place the logical interfaces into separate security zones. Reth1.991 belongs in the JNPR_1 security zone and Reth1.992 belongs in the JNPR_2 security zone.

```
set security zones security-zone JNPR_1-Zone host-inbound-traffic system-services ping
set security zones security-zone JNPR_1-Zone host-inbound-traffic protocols bgp
set security zones security-zone JNPR_1-Zone interfaces reth1.991

set security zones security-zone JNPR_2-Zone host-inbound-traffic system-services ping
set security zones security-zone JNPR_2-Zone host-inbound-traffic protocols bgp
set security zones security-zone JNPR_2-Zone interfaces reth1.992
```

4. Check the status of the chassis cluster.

```
user@srx1> show chassis cluster status
Monitor Failure codes:
    CS  Cold Sync monitoring        FL  Fabric Connection monitoring
    GR  GRES monitoring             HW  Hardware monitoring
    IF  Interface monitoring        IP  IP monitoring
    LB  Loopback monitoring         MB  Mbuf monitoring
    NH  Nexthop monitoring          NP  NPC monitoring
    SP  SPU monitoring              SM  Schedule monitoring
    CF  Config Sync monitoring      RE  Relinquish monitoring

Cluster ID: 1
Node   Priority Status               Preempt Manual   Monitor-failures

Redundancy group: 0 , Failover count: 1
node0  100      primary              no      no       None
node1  1        secondary            no      no       None

Redundancy group: 1 , Failover count: 5
node0  200      primary              yes     no       None
node1  100      secondary            yes     no       None
```

Configure Spine 1

Step-by-Step Procedure

1. Configure the SRX Series Firewall interconnected interfaces on Spine 1.

```
set interfaces ge-0/0/10 ether-options 802.3ad ae11
set interfaces ge-0/0/11 ether-options 802.3ad ae12

set interfaces ae11 description "to SRX Cluster | SRX-0"
set interfaces ae11 mtu 9216
set interfaces ae11 esi 00:00:00:00:00:00:00:00:01:11
set interfaces ae11 esi all-active
set interfaces ae11 aggregated-ether-options lacp active
set interfaces ae11 aggregated-ether-options lacp periodic fast
set interfaces ae11 aggregated-ether-options lacp system-id 00:00:00:00:01:11
set interfaces ae11 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae11 unit 0 family ethernet-switching vlan members VLAN-991

set interfaces ae12 description "to SRX Cluster | SRX-1"
set interfaces ae12 mtu 9216
set interfaces ae12 esi 00:00:00:00:00:00:00:00:01:12
set interfaces ae12 esi all-active
set interfaces ae12 aggregated-ether-options lacp active
set interfaces ae12 aggregated-ether-options lacp periodic fast
set interfaces ae11 aggregated-ether-options lacp system-id 00:00:00:00:01:12
set interfaces ae12 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae12 unit 0 family ethernet-switching vlan members VLAN-992
```

2. Configure IRB interfaces.

```
set interfaces irb unit 991 description "Tenant1 SRX Interconnect"
set interfaces irb unit 991 family inet address 192.168.191.3/28
set routing-instances JNPR_1_VRF interface irb.991

set interfaces irb unit 992 description "Tenant2 SRX Interconnect"
set interfaces irb unit 992 family inet address 192.168.192.3/28
set routing-instances JNPR_2_VRF interface irb.992
```

3. Configure the VLANs.

```
set vlans VLAN-991 vlan-id 991
set vlans VLAN-991 l3-interface irb.991
set vlans VLAN-991 vxlan vni 5991

set vlans VLAN-992 vlan-id 992
set vlans VLAN-992 l3-interface irb.992
set vlans VLAN-992 vxlan vni 5992
```

4. Configure VNIs as part of the EVPN MP-BGP domain.

```
set protocols evpn extended-vni-list 5991
set protocols evpn extended-vni-list 5992
```

Configure Spine 2

Step-by-Step Procedure

1. Configure the SRX Series Firewall interconnected interfaces on Spine 2.

```
set interfaces ge-0/0/10 ether-options 802.3ad ae11
set interfaces ge-0/0/11 ether-options 802.3ad ae12

set interfaces ae11 description "to SRX Cluster | SRX-0"
set interfaces ae11 mtu 9216
set interfaces ae11 esi 00:00:00:00:00:00:00:00:01:11
set interfaces ae11 esi all-active
set interfaces ae12 aggregated-ether-options lacp active
set interfaces ae12 aggregated-ether-options lacp periodic fast
set interfaces ae11 aggregated-ether-options lacp system-id 00:00:00:00:01:11
set interfaces ae11 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae11 unit 0 family ethernet-switching vlan members VLAN-991

set interfaces ae12 description "to SRX Cluster | SRX-1"
set interfaces ae12 mtu 9216
set interfaces ae12 esi 00:00:00:00:00:00:00:00:01:12
set interfaces ae12 esi all-active
set interfaces ae12 aggregated-ether-options lacp active
set interfaces ae12 aggregated-ether-options lacp periodic fast
set interfaces ae11 aggregated-ether-options lacp system-id 00:00:00:00:01:12
set interfaces ae12 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae12 unit 0 family ethernet-switching vlan members VLAN-992
```

2. Configure IRB interfaces.

```
set interfaces irb unit 991 description "Tenant1 SRX Interconnect"
set interfaces irb unit 991 family inet address 192.168.191.2/28
set routing-instances JNPR_1_VRF interface irb.991

set interfaces irb unit 992 description "Tenant2 SRX Interconnect"
set interfaces irb unit 992 family inet address 192.168.192.2/28
set routing-instances JNPR_2_VRF interface irb.992
```

3. Configure the VLANs.

```
set vlans VLAN-991 vlan-id 991
set vlans VLAN-991 l3-interface irb.991
set vlans VLAN-991 vxlan vni 5991

set vlans VLAN-992 vlan-id 992
set vlans VLAN-992 l3-interface irb.992
set vlans VLAN-992 vxlan vni 5992
```

4. Configure VNIs as part of the EVPN MP-BGP domain.

```
set protocols evpn extended-vni-list 5991
set protocols evpn extended-vni-list 5992
```

Configure EBGP

IN THIS SECTION

Configure the SRX Device

Step-by-Step Procedure

1. Configure the EBGP interconnect.

```
set protocols bgp group INTERCONNECT type external
set protocols bgp group INTERCONNECT import INTERCONNECT-IMPORT
set protocols bgp group INTERCONNECT family inet unicast
set protocols bgp group INTERCONNECT authentication-key "$ABC123"
set protocols bgp group INTERCONNECT export INTERCONNECT-EXPORT
set protocols bgp group INTERCONNECT local-as 65200
set protocols bgp group INTERCONNECT multipath multiple-as
set protocols bgp group INTERCONNECT bfd-liveness-detection minimum-interval 1000
set protocols bgp group INTERCONNECT bfd-liveness-detection multiplier 3
set protocols bgp group INTERCONNECT neighbor 192.168.191.2 peer-as 65112
set protocols bgp group INTERCONNECT neighbor 192.168.191.3 peer-as 65113
set protocols bgp group INTERCONNECT neighbor 192.168.192.2 peer-as 65212
set protocols bgp group INTERCONNECT neighbor 192.168.192.3 peer-as 65213
```

2. Configure the routing options.

```
set routing-options static route 192.168.0.0/16 discard
```

3. Configure the policy options.

```
set policy-options policy-statement INTERCONNECT-EXPORT term Tenant_Aggregate from protocol static
set policy-options policy-statement INTERCONNECT-EXPORT term Tenant_Aggregate from route-filter 192.168.0.0/16 exact
set policy-options policy-statement INTERCONNECT-EXPORT term Tenant_Aggregate then accept
set policy-options policy-statement INTERCONNECT-EXPORT term Advertise_Loopback from protocol direct
set policy-options policy-statement INTERCONNECT-EXPORT term Advertise_Loopback from route-filter 192.168.255.1/32 exact
set policy-options policy-statement INTERCONNECT-EXPORT term Advertise_Loopback then accept
set policy-options policy-statement INTERCONNECT-EXPORT term Reject_All then reject

set policy-options policy-statement INTERCONNECT-IMPORT term Tenant_Routes from route-filter 192.168.0.0/16 longer
set policy-options policy-statement INTERCONNECT-IMPORT term Tenant_Routes then accept
set policy-options policy-statement INTERCONNECT-IMPORT term DEFAULT then reject
```

Configure Spine 1

Step-by-Step Procedure

1. Configure the EBGP peerings in the JNPR_1 routing instance.

```
set routing-instances JNPR_1_VRF protocols bgp group INTERCONNECT type external
set routing-instances JNPR_1_VRF protocols bgp group INTERCONNECT import Interconnect_JNPR_1-IMPORT
set routing-instances JNPR_1_VRF protocols bgp group INTERCONNECT family inet unicast
set routing-instances JNPR_1_VRF protocols bgp group INTERCONNECT authentication-key "$ABC123"
set routing-instances JNPR_1_VRF protocols bgp group INTERCONNECT export Interconnect_JNPR_1-EXPORT
set routing-instances JNPR_1_VRF protocols bgp group INTERCONNECT local-as 65113
set routing-instances JNPR_1_VRF protocols bgp group INTERCONNECT multipath multiple-as
set routing-instances JNPR_1_VRF protocols bgp group INTERCONNECT bfd-liveness-detection
```

```
minimum-interval 1000
set routing-instances JNPR_1_VRF protocols bgp group INTERCONNECT bfd-liveness-detection multiplier 3
set routing-instances JNPR_1_VRF protocols bgp group INTERCONNECT neighbor 192.168.191.1 peer-as 65200
```

2. Configure the EBGP peerings in the JNPR_2 routing instance.

```
set routing-instances JNPR_2_VRF protocols bgp group INTERCONNECT type external
set routing-instances JNPR_2_VRF protocols bgp group INTERCONNECT import Interconnect_JNPR_2-IMPORT
set routing-instances JNPR_2_VRF protocols bgp group INTERCONNECT family inet unicast
set routing-instances JNPR_2_VRF protocols bgp group INTERCONNECT authentication-key "$ABC123"
set routing-instances JNPR_2_VRF protocols bgp group INTERCONNECT export Interconnect_JNPR_2-EXPORT
set routing-instances JNPR_2_VRF protocols bgp group INTERCONNECT local-as 65213
set routing-instances JNPR_2_VRF protocols bgp group INTERCONNECT multipath multiple-as
set routing-instances JNPR_2_VRF protocols bgp group INTERCONNECT bfd-liveness-detection minimum-interval 1000
set routing-instances JNPR_2_VRF protocols bgp group INTERCONNECT bfd-liveness-detection multiplier 3
set routing-instances JNPR_2_VRF protocols bgp group INTERCONNECT neighbor 192.168.192.1 peer-as 65200
```

3. Configure the import and export policies for interconnect with the SRX device.

```
set policy-options policy-statement Interconnect_JNPR_1-EXPORT term Tenant_Routes from route-filter 192.168.0.0/16 orlonger
set policy-options policy-statement Interconnect_JNPR_1-EXPORT term Tenant_Routes then accept
set policy-options policy-statement Interconnect_JNPR_1-EXPORT term DEFAULT then reject
set policy-options policy-statement Interconnect_JNPR_1-IMPORT term Tenant_Routes from route-filter 192.168.0.0/16 orlonger
set policy-options policy-statement Interconnect_JNPR_1-IMPORT term Tenant_Routes then accept
set policy-options policy-statement Interconnect_JNPR_1-IMPORT term DEFAULT then reject

set policy-options policy-statement Interconnect_JNPR_2-EXPORT term Tenant_Routes from route-filter 192.168.0.0/16 orlonger
set policy-options policy-statement Interconnect_JNPR_2-EXPORT term Tenant_Routes then accept
set policy-options policy-statement Interconnect_JNPR_2-EXPORT term DEFAULT then reject
set policy-options policy-statement Interconnect_JNPR_2-IMPORT term Tenant_Routes from route-filter 192.168.0.0/16 orlonger
set policy-options policy-statement Interconnect_JNPR_2-IMPORT term Tenant_Routes then accept
set policy-options policy-statement Interconnect_JNPR_2-IMPORT term DEFAULT then reject
```

Configure Spine 2

Step-by-Step Procedure

1. Configure the EBGP peerings in the JNPR_1 routing instance.

```
set routing-instances JNPR_1_VRF protocols bgp group INTERCONNECT type external
set routing-instances JNPR_1_VRF protocols bgp group INTERCONNECT import Interconnect_JNPR_1-
IMPORT
set routing-instances JNPR_1_VRF protocols bgp group INTERCONNECT family inet unicast
set routing-instances JNPR_1_VRF protocols bgp group INTERCONNECT authentication-key "$ABC123"
set routing-instances JNPR_1_VRF protocols bgp group INTERCONNECT export Interconnect_JNPR_1-
EXPORT
set routing-instances JNPR_1_VRF protocols bgp group INTERCONNECT local-as 65112
set routing-instances JNPR_1_VRF protocols bgp group INTERCONNECT multipath multiple-as
set routing-instances JNPR_1_VRF protocols bgp group INTERCONNECT bfd-liveness-detection
minimum-interval 1000
set routing-instances JNPR_1_VRF protocols bgp group INTERCONNECT bfd-liveness-detection
multiplier 3
set routing-instances JNPR_1_VRF protocols bgp group INTERCONNECT neighbor 192.168.191.1 peer-
as 65200
```

2. Configure the EBGP peerings in the JNPR_2 routing instance.

```
set routing-instances JNPR_2_VRF protocols bgp group INTERCONNECT type external
set routing-instances JNPR_2_VRF protocols bgp group INTERCONNECT import Interconnect_JNPR_2-
IMPORT
set routing-instances JNPR_2_VRF protocols bgp group INTERCONNECT family inet unicast
set routing-instances JNPR_2_VRF protocols bgp group INTERCONNECT authentication-key "$ABC123"
set routing-instances JNPR_2_VRF protocols bgp group INTERCONNECT export Interconnect_JNPR_2-
EXPORT
set routing-instances JNPR_2_VRF protocols bgp group INTERCONNECT local-as 65212
set routing-instances JNPR_2_VRF protocols bgp group INTERCONNECT multipath multiple-as
set routing-instances JNPR_2_VRF protocols bgp group INTERCONNECT bfd-liveness-detection
minimum-interval 1000
set routing-instances JNPR_2_VRF protocols bgp group INTERCONNECT bfd-liveness-detection
multiplier 3
```

```
set routing-instances JNPR_2_VRF protocols bgp group INTERCONNECT neighbor 192.168.192.1 peer-
as 65200
```

3. Configure the import and export policies for interconnect with the SRX device.

```
set policy-options policy-statement Interconnect_JNPR_1-EXPORT term Tenant_Routes from route-
filter 192.168.0.0/16 orlonger
set policy-options policy-statement Interconnect_JNPR_1-EXPORT term Tenant_Routes then accept
set policy-options policy-statement Interconnect_JNPR_1-EXPORT term DEFAULT then reject
set policy-options policy-statement Interconnect_JNPR_1-IMPORT term Tenant_Routes from route-
filter 192.168.0.0/16 orlonger
set policy-options policy-statement Interconnect_JNPR_1-IMPORT term Tenant_Routes then accept
set policy-options policy-statement Interconnect_JNPR_1-IMPORT term DEFAULT then reject

set policy-options policy-statement Interconnect_JNPR_2-EXPORT term Tenant_Routes from route-
filter 192.168.0.0/16 orlonger
set policy-options policy-statement Interconnect_JNPR_2-EXPORT term Tenant_Routes then accept
set policy-options policy-statement Interconnect_JNPR_2-EXPORT term DEFAULT then reject
set policy-options policy-statement Interconnect_JNPR_2-IMPORT term Tenant_Routes from route-
filter 192.168.0.0/16 orlonger
set policy-options policy-statement Interconnect_JNPR_2-IMPORT term Tenant_Routes then accept
set policy-options policy-statement Interconnect_JNPR_2-IMPORT term DEFAULT then reject
```

Configure the SRX Series Firewall Security Policies

Step-by-Step Procedure

1. Configure the security policies in Zone 1 for JNPR_1.

```
set security policies from-zone JNPR_1-Zone to-zone JNPR_2-Zone policy Allow_All match source-
address any
set security policies from-zone JNPR_1-Zone to-zone JNPR_2-Zone policy Allow_All match
destination-address any
set security policies from-zone JNPR_1-Zone to-zone JNPR_2-Zone policy Allow_All match
application any
set security policies from-zone JNPR_1-Zone to-zone JNPR_2-Zone policy Allow_All then permit
```

2. Configure the security policies in Zone 1 for JNPR_2.

```
set security policies from-zone JNPR_2-Zone to-zone JNPR_1-Zone policy Allow_All match source-
address any
set security policies from-zone JNPR_2-Zone to-zone JNPR_1-Zone policy Allow_All match
destination-address any
set security policies from-zone JNPR_2-Zone to-zone JNPR_1-Zone policy Allow_All match
application any
set security policies from-zone JNPR_2-Zone to-zone JNPR_1-Zone policy Allow_All then permit
```

Verify BGP on the SRX Chassis Cluster

Step-by-Step Procedure

1. Ensure that all BGP peering sessions with the spine switches are established.

```
user@srx> show bgp summary
Groups: 1 Peers: 4 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0
                      26         14          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/
Received/Accepted/Damped...
192.168.191.2         65112        113        106       0      73       47:34 Establ
  inet.0: 4/7/7/0
192.168.191.3         65113        110        107       0      41       47:35 Establ
  inet.0: 4/7/7/0
192.168.192.2         65212        111        106       0      71       47:35 Establ
  inet.0: 3/6/6/0
192.168.192.3         65213        109        106       0      34       47:35 Establ
  inet.0: 3/6/6/0
```

2. Verify the SRX Series Firewall received the BGP routes from the JNPR_1 tenant.

```
user@srx> show route receive-protocol bgp 192.168.191.2
inet.0: 18 destinations, 35 routes (18 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
  192.168.191.0/28        192.168.191.2                           65112 I
  192.168.191.1/32        192.168.191.2                           65112 I
  192.168.201.0/24        192.168.191.2                           65112 I
* 192.168.202.42/32       192.168.191.2                           65112 I
  192.168.202.0/24        192.168.191.2                           65112 I
* 192.168.251.12/32       192.168.191.2                           65112 I
  192.168.251.13/32       192.168.191.2                           65112 65100 I
```

```
user@srx> show route receive-protocol bgp 192.168.191.3
inet.0: 18 destinations, 35 routes (18 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
  192.168.191.0/28        192.168.191.3                           65113 I
  192.168.191.1/32        192.168.191.3                           65113 I
* 192.168.201.0/24        192.168.191.3                           65113 I
  192.168.202.42/32       192.168.191.3                           65113 I
* 192.168.202.0/24        192.168.191.3                           65113 I
  192.168.251.12/32       192.168.191.3                           65113 65100 I
* 192.168.251.13/32       192.168.191.3                           65113 I
```

3. Verify the SRX Series Firewall received the BGP routes from the JNPR_2 tenant.

```
user@srx> show route receive-protocol bgp 192.168.192.2
inet.0: 18 destinations, 35 routes (18 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
  192.168.192.0/28        192.168.192.2                           65212 I
  192.168.192.1/32        192.168.192.2                           65212 I
  192.168.211.0/24        192.168.192.2                           65212 I
  192.168.212.0/24        192.168.192.2                           65212 I
* 192.168.252.12/32       192.168.192.2                           65212 I
  192.168.252.13/32       192.168.192.2                           65212 65100 I
```

```
user@srx> show route receive-protocol bgp 192.168.192.3
inet.0: 18 destinations, 35 routes (18 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
  192.168.192.0/28        192.168.192.3                           65213 I
  192.168.192.1/32        192.168.192.3                           65213 I
* 192.168.211.0/24        192.168.192.3                           65213 I
* 192.168.212.0/24        192.168.192.3                           65213 I
  192.168.252.12/32       192.168.192.3                           65213 65100 I
* 192.168.252.13/32       192.168.192.3                           65213 I
```

4. Verify that the SRX chassis cluster is advertising a summary route to the spine devices.

```
user@srx> show route advertising-protocol bgp 192.168.191.2
inet.0: 18 destinations, 35 routes (18 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 192.168.0.0/16          Self                                    I
* 192.168.255.1/32        Self                                    I
```

```
user@srx> show route advertising-protocol bgp 192.168.191.3
inet.0: 18 destinations, 35 routes (18 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 192.168.0.0/16          Self                                    I
* 192.168.255.1/32        Self                                    I
```

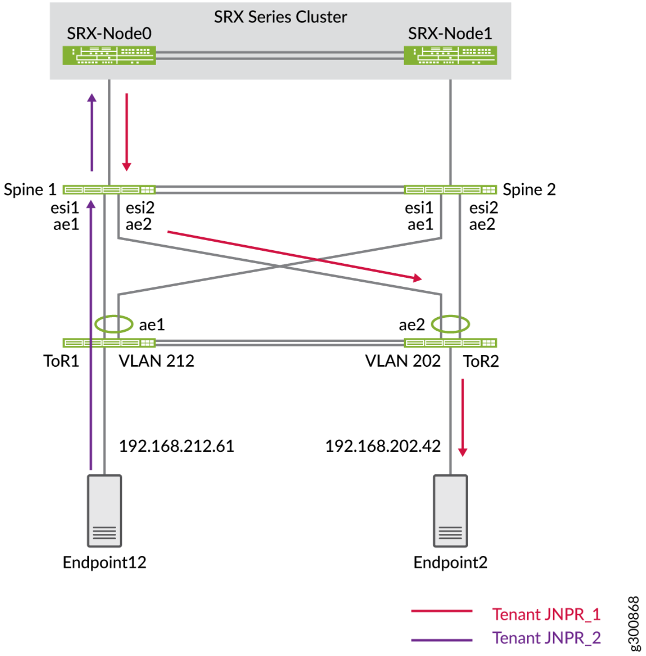
```
user@srx> show route advertising-protocol bgp 192.168.192.2
inet.0: 18 destinations, 35 routes (18 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 192.168.0.0/16          Self                                    I
* 192.168.255.1/32        Self                                    I
```

```
user@srx> show route advertising-protocol bgp 192.168.192.3
inet.0: 18 destinations, 35 routes (18 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 192.168.0.0/16          Self                                    I
* 192.168.255.1/32        Self                                    I
```

5. Verify inter-tenant traffic through the SRX chassis cluster.

   In this example, Endpoint12 is part of VLAN 212 and tenant JNPR_2. Endpoint12 is pinging Endpoint2, which is part of VLAN 201 and tenant JNPR_1, as shown in Figure 19 on page 70. Since this is inter-tenant traffic, this traffic goes through the active member of the SRX chassis cluster. SRX-Node0 is the active member of the SRX chassis cluster and SRX-Node1 is the passive member.

Figure 19: Inter-Tenant Traffic Through the SRX Cluster



Confirm that the flow table on the SRX Series Firewall shows this traffic traversing the SRX chassis cluster.

```
user@srx> show security flow session destination-prefix 192.168.202.42
node0:
--------------------------------------------------------------------------


Session ID: 15548, Policy name: Allow_All/7, State: Active, Timeout: 2, Valid
  In: 192.168.212.61/623 --> 192.168.202.42/8204;icmp, Conn Tag: 0x0, If: reth1.992, Pkts: 1,
Bytes: 84,
  Out: 192.168.202.42/8204 --> 192.168.212.61/623;icmp, Conn Tag: 0x0, If: reth1.991, Pkts:
1, Bytes: 84,

Session ID: 15551, Policy name: Allow_All/7, State: Active, Timeout: 2, Valid
  In: 192.168.212.61/624 --> 192.168.202.42/8204;icmp, Conn Tag: 0x0, If: reth1.992, Pkts: 1,
Bytes: 84,
  Out: 192.168.202.42/8204 --> 192.168.212.61/624;icmp, Conn Tag: 0x0, If: reth1.991, Pkts:
1, Bytes: 84,

Session ID: 15555, Policy name: Allow_All/7, State: Active, Timeout: 4, Valid
  In: 192.168.212.61/625 --> 192.168.202.42/8204;icmp, Conn Tag: 0x0, If: reth1.992, Pkts: 1,
Bytes: 84,
  Out: 192.168.202.42/8204 --> 192.168.212.61/625;icmp, Conn Tag: 0x0, If: reth1.991, Pkts:
1, Bytes: 84,
Total sessions: 3
```

You have configured advanced security for your data center and confirmed that inter-tenant traffic is routed through the SRX chassis cluster.

Configure Data Center Interconnect (DCI)

IN THIS SECTION

Requirements

- The devices that you configured in "How to Configure a Collapsed Spine with EVPN Multihoming" on page 12 and "Configure Advanced Security for Inter-Tenant Traffic" on page 54.

Overview

IN THIS SECTION

Now that you have configured a collapsed spine architecture for both data centers and added advanced security to DC1, it is time to connect DC1 and DC2 using Data Center Interconnect (DCI).

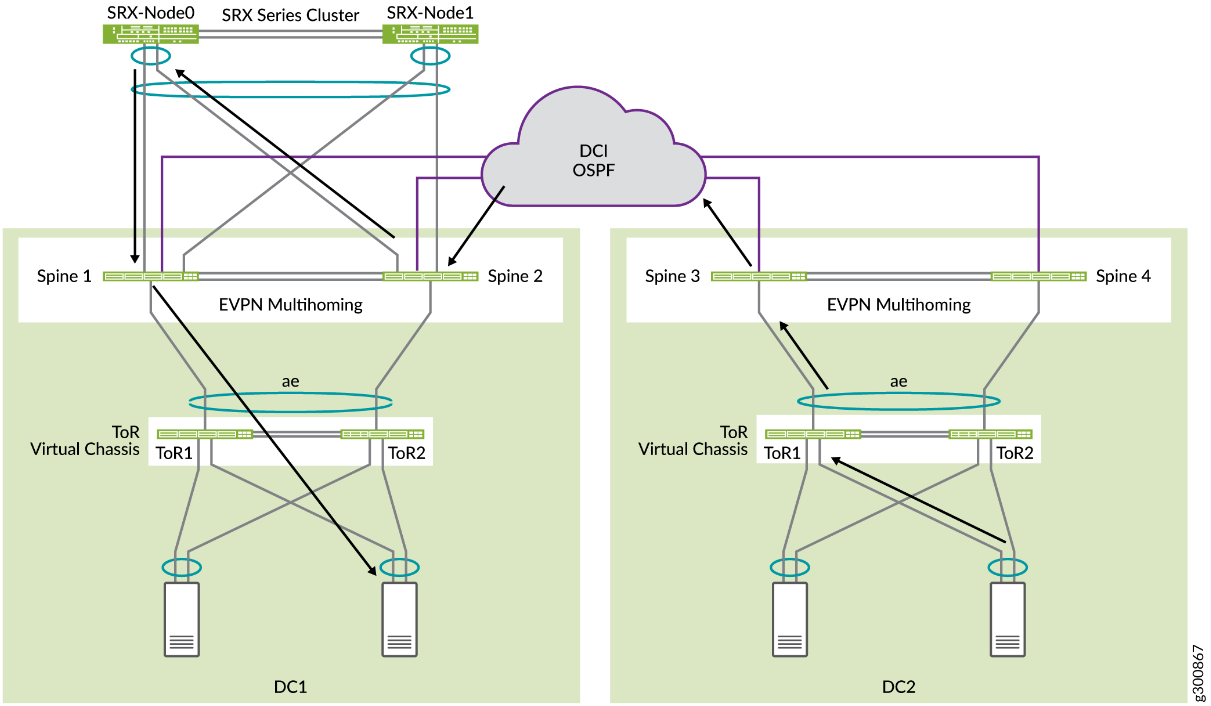
Topology

In this example, there is no need to stretch Layer 2 between data centers. Inter-data center communication is routed through the SRX chassis cluster in DC1, as shown in Figure 20 on page 72.

The spine switches each have a WAN routing instance and are connected to the WAN between data centers. The spine switches hand off the Layer 3 routes to the WAN router (not shown in this figure).

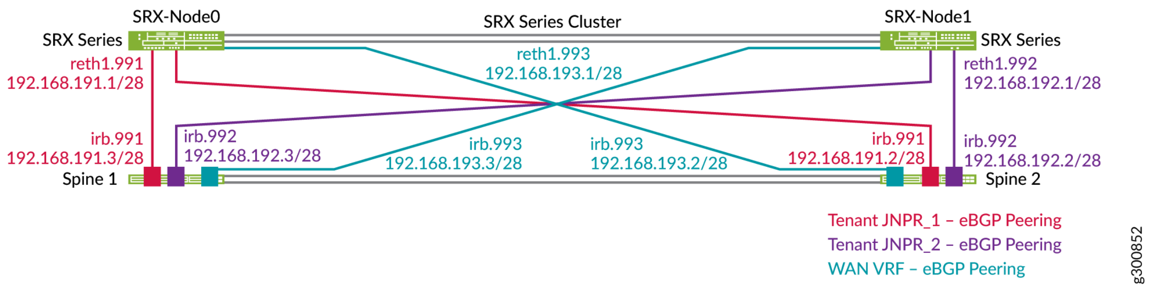
The SRX chassic cluster is advertising a 192.168.0.0/16 subnet. The DC2 spine switches Spine 3 and Spine 4 are advertising the two subnets 192.168.221.0/24 and 192.168.222.0/24.

Figure 20: Data Center Interconnect Topology



Each SRX Series Firewall is configured with three zones that correspond to the JNPR_1, JNPR_2, and WAN routing instances. All inter-tenant traffic between JNPR_1 and JNPR_2 is routed through the SRX chassis cluster. All traffic between DC1 and DC2 is routed through the SRX chassis cluster using the WAN routing instance. Each SRX Series Firewall has individual EBGP peering with Spine 1 and Spine 2 in each of the routing instances. Figure 21 on page 73 shows the EBGP peering between the spine switches and the SRX chassis cluster in DC1.

Figure 21: SRX Chassis Cluster EBGP Peering Topology



Configuration

IN THIS SECTION

Configure the SRX Device

Step-by-Step Procedure

Each SRX Series Firewall must be divided into three zones that correspond to the three routing instances: JNPR_1, JNPR_2, and WAN. You already created the JNPR_1 zone and the JNPR_2 zone in "Configure Advanced Security for Inter-Tenant Traffic" on page 54.

1. Add a new sub interface on Reth1 for the WAN interconnect.

```
set interfaces reth1 unit 993 description "DC1 Spine Interconnect for WAN VRF"
set interfaces reth1 unit 993 vlan-id 993
set interfaces reth1 unit 993 family inet address 192.168.193.1/28
```

2. Configure the WAN security zone.

```
set security zones security-zone WAN-Zone host-inbound-traffic system-services ping
set security zones security-zone WAN-Zone host-inbound-traffic protocols bgp
set security zones security-zone WAN-Zone interfaces reth1.993
```

3. Configure EBGP for the WAN security zone.

```
set protocols bgp group INTERCONNECT neighbor 192.168.193.2 peer-as 65312
set protocols bgp group INTERCONNECT neighbor 192.168.193.3 peer-as 65313
```

4. Configure the security policies. For simplicity, the security policies in this example are open. In your setup, modify the security policies as necessary.

```
set security address-book global address 192.168.221.0/24 192.168.221.0/24
set security address-book global address 192.168.222.0/24 192.168.222.0/24
```

```
set security policies from-zone WAN-Zone to-zone JNPR_1-Zone policy ALLOW_ALL match source-
address 192.168.221.0/24
set security policies from-zone WAN-Zone to-zone JNPR_1-Zone policy ALLOW_ALL match source-
address 192.168.222.0/24
set security policies from-zone WAN-Zone to-zone JNPR_1-Zone policy ALLOW_ALL match
destination-address any
set security policies from-zone WAN-Zone to-zone JNPR_1-Zone policy ALLOW_ALL match
application any
set security policies from-zone WAN-Zone to-zone JNPR_1-Zone policy ALLOW_ALL then permit
```

```
set security policies from-zone WAN-Zone to-zone JNPR_2-Zone policy ALLOW_ALL match source-
address 192.168.221.0/24
set security policies from-zone WAN-Zone to-zone JNPR_2-Zone policy ALLOW_ALL match source-
address 192.168.222.0/24
set security policies from-zone WAN-Zone to-zone JNPR_2-Zone policy ALLOW_ALL match
destination-address any
set security policies from-zone WAN-Zone to-zone JNPR_2-Zone policy ALLOW_ALL match
```

```
application any
set security policies from-zone WAN-Zone to-zone JNPR_2-Zone policy ALLOW_ALL then permit
```

```
set security policies from-zone JNPR_1-Zone to-zone WAN-Zone policy ALLOW_ALL match source-
address any
set security policies from-zone JNPR_1-Zone to-zone WAN-Zone policy ALLOW_ALL match
destination-address 192.168.222.0/24
set security policies from-zone JNPR_1-Zone to-zone WAN-Zone policy ALLOW_ALL match
destination-address 192.168.221.0/24
set security policies from-zone JNPR_1-Zone to-zone WAN-Zone policy ALLOW_ALL match
application any
set security policies from-zone JNPR_1-Zone to-zone WAN-Zone policy ALLOW_ALL then permit
```

```
set security policies from-zone JNPR_2-Zone to-zone WAN-Zone policy ALLOW_ALL match source-
address any
set security policies from-zone JNPR_2-Zone to-zone WAN-Zone policy ALLOW_ALL match
destination-address 192.168.222.0/24
set security policies from-zone JNPR_2-Zone to-zone WAN-Zone policy ALLOW_ALL match
destination-address 192.168.221.0/24
set security policies from-zone JNPR_2-Zone to-zone WAN-Zone policy ALLOW_ALL match
application any
set security policies from-zone JNPR_2-Zone to-zone WAN-Zone policy ALLOW_ALL then permit
```

Configure Spine Switches

Step-by-Step Procedure

1. Configure the routing instances and irb interface on Spine 1.

```
set interfaces irb unit 993 family inet address 192.168.193.3/28
set routing-instances WAN_VRF description "VRF for tenant WAN"
set routing-instances WAN_VRF instance-type vrf
set routing-instances WAN_VRF interface et-0/0/48.0
set routing-instances WAN_VRF interface irb.993
set routing-instances WAN_VRF interface lo0.103
set routing-instances WAN_VRF route-distinguisher 192.168.253.13:103
set routing-instances WAN_VRF vrf-target target:3:65001
set routing-instances WAN_VRF vrf-table-label
set routing-instances WAN_VRF routing-options auto-export
set routing-instances WAN_VRF routing-options multipath
set routing-instances WAN_VRF protocols bgp group INTERCONNECT type external
set routing-instances WAN_VRF protocols bgp group INTERCONNECT import Interconnect_WAN-IMPORT
set routing-instances WAN_VRF protocols bgp group INTERCONNECT family inet unicast
set routing-instances WAN_VRF protocols bgp group INTERCONNECT authentication-key "$ABC123"
set routing-instances WAN_VRF protocols bgp group INTERCONNECT export Interconnect_WAN-EXPORT
set routing-instances WAN_VRF protocols bgp group INTERCONNECT local-as 65313
set routing-instances WAN_VRF protocols bgp group INTERCONNECT multipath multiple-as
set routing-instances WAN_VRF protocols bgp group INTERCONNECT bfd-liveness-detection minimum-
interval 1000
set routing-instances WAN_VRF protocols bgp group INTERCONNECT bfd-liveness-detection
multiplier 3
set routing-instances WAN_VRF protocols bgp group INTERCONNECT neighbor 192.168.193.1 peer-as
65200

set routing-instances WAN_VRF protocols bgp group WAN_UNDERLAY type external
set routing-instances WAN_VRF protocols bgp group WAN_UNDERLAY description "Connection to
EBGP WAN_UNDERLAY"
set routing-instances WAN_VRF protocols bgp group WAN_UNDERLAY family inet unicast
set routing-instances WAN_VRF protocols bgp group WAN_UNDERLAY authentication-key "$ABC123"
set routing-instances WAN_VRF protocols bgp group WAN_UNDERLAY local-as 65313
set routing-instances WAN_VRF protocols bgp group WAN_UNDERLAY multipath multiple-as
set routing-instances WAN_VRF protocols bgp group WAN_UNDERLAY bfd-liveness-detection minimum-
interval 350
set routing-instances WAN_VRF protocols bgp group WAN_UNDERLAY bfd-liveness-detection
multiplier 3
set routing-instances WAN_VRF protocols bgp group WAN_UNDERLAY neighbor 192.168.100.2 peer-as
65300
```

2. Configure the routing instances on Spine 2.

```
set interfaces irb unit 993 family inet address 192.168.193.2/28
set routing-instances WAN_VRF description "VRF for tenant WAN"
set routing-instances WAN_VRF instance-type vrf
set routing-instances WAN_VRF interface et-0/0/48.0
set routing-instances WAN_VRF interface irb.993
set routing-instances WAN_VRF interface lo0.103
set routing-instances WAN_VRF route-distinguisher 192.168.253.12:103
set routing-instances WAN_VRF vrf-target target:3:65001
set routing-instances WAN_VRF vrf-table-label
set routing-instances WAN_VRF routing-options auto-export
set routing-instances WAN_VRF routing-options multipath
set routing-instances WAN_VRF protocols bgp group INTERCONNECT type external
set routing-instances WAN_VRF protocols bgp group INTERCONNECT import Interconnect_WAN-IMPORT
set routing-instances WAN_VRF protocols bgp group INTERCONNECT family inet unicast
set routing-instances WAN_VRF protocols bgp group INTERCONNECT authentication-key "$ABC123"
set routing-instances WAN_VRF protocols bgp group INTERCONNECT export Interconnect_WAN-EXPORT
set routing-instances WAN_VRF protocols bgp group INTERCONNECT local-as 65312
set routing-instances WAN_VRF protocols bgp group INTERCONNECT multipath multiple-as
set routing-instances WAN_VRF protocols bgp group INTERCONNECT bfd-liveness-detection minimum-
interval 1000
set routing-instances WAN_VRF protocols bgp group INTERCONNECT bfd-liveness-detection
multiplier 3
set routing-instances WAN_VRF protocols bgp group INTERCONNECT neighbor 192.168.193.1 peer-as
65200

set routing-instances WAN_VRF protocols bgp group WAN_UNDERLAY type external
set routing-instances WAN_VRF protocols bgp group WAN_UNDERLAY description "Connection to
EBGP WAN_UNDERLAY"
set routing-instances WAN_VRF protocols bgp group WAN_UNDERLAY family inet unicast
set routing-instances WAN_VRF protocols bgp group WAN_UNDERLAY authentication-key "$ABC123"
set routing-instances WAN_VRF protocols bgp group WAN_UNDERLAY local-as 65312
set routing-instances WAN_VRF protocols bgp group WAN_UNDERLAY multipath multiple-as
set routing-instances WAN_VRF protocols bgp group WAN_UNDERLAY bfd-liveness-detection minimum-
interval 350
set routing-instances WAN_VRF protocols bgp group WAN_UNDERLAY bfd-liveness-detection
multiplier 3
set routing-instances WAN_VRF protocols bgp group WAN_UNDERLAY neighbor 192.168.100.0 peer-as
65300
```

3. Configure EBGP on Spine 3.

```
set protocols bgp group WAN_UNDERLAY type external
set protocols bgp group WAN_UNDERLAY description "Connection to EBGP WAN_UNDERLAY"
set protocols bgp group WAN_UNDERLAY family inet unicast
set protocols bgp group WAN_UNDERLAY authentication-key "$ABC123"
set protocols bgp group WAN_UNDERLAY export WAN_EXPORT
set protocols bgp group WAN_UNDERLAY local-as 65322
set protocols bgp group WAN_UNDERLAY multipath multiple-as
set protocols bgp group WAN_UNDERLAY bfd-liveness-detection minimum-interval 350
set protocols bgp group WAN_UNDERLAY bfd-liveness-detection multiplier 3
set protocols bgp group WAN_UNDERLAY neighbor 192.168.100.10 peer-as 65300
set policy-options policy-statement WAN_EXPORT term DIRECT_ROUTES from protocol direct
set policy-options policy-statement WAN_EXPORT term DIRECT_ROUTES from route-filter
192.168.221.0/24 exact
set policy-options policy-statement WAN_EXPORT term DIRECT_ROUTES from route-filter
192.168.222.0/24 exact
set policy-options policy-statement WAN_EXPORT term DIRECT_ROUTES then accept
```

4. Configure EBGP on Spine 4.

```
set protocols bgp group WAN_UNDERLAY type external
set protocols bgp group WAN_UNDERLAY description "Connection to EBGP WAN_UNDERLAY"
set protocols bgp group WAN_UNDERLAY family inet unicast
set protocols bgp group WAN_UNDERLAY authentication-key "$ABC123"
set protocols bgp group WAN_UNDERLAY export WAN_EXPORT
set protocols bgp group WAN_UNDERLAY local-as 65323
set protocols bgp group WAN_UNDERLAY multipath multiple-as
set protocols bgp group WAN_UNDERLAY bfd-liveness-detection minimum-interval 350
set protocols bgp group WAN_UNDERLAY bfd-liveness-detection multiplier 3
set protocols bgp group WAN_UNDERLAY neighbor 192.168.100.16 peer-as 65300

set policy-options policy-statement WAN_EXPORT term DIRECT_ROUTES from protocol direct
set policy-options policy-statement WAN_EXPORT term DIRECT_ROUTES from route-filter
192.168.221.0/24 exact
set policy-options policy-statement WAN_EXPORT term DIRECT_ROUTES from route-filter
192.168.222.0/24 exact
set policy-options policy-statement WAN_EXPORT term DIRECT_ROUTES then accept
```

Verify DCI Routes

Step-by-Step Procedure

1. Verify the routes on the SRX chassis cluster. The SRX should learn all the specific routes for the different subnets.

```
user@srx> show route
inet.0: 31 destinations, 37 routes (31 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.201.0/24   *[BGP/170] 00:59:11, localpref 100
                      AS path: 65113 I, validation-state: unverified
```

```
                    > to 192.168.191.3 via reth1.991
192.168.201.10/32  *[BGP/170] 00:00:07, localpref 100
                      AS path: 65113 I, validation-state: unverified
                    > to 192.168.191.3 via reth1.991
192.168.201.81/32  *[BGP/170] 00:00:07, localpref 100
                      AS path: 65113 I, validation-state: unverified
                    > to 192.168.191.3 via reth1.991
192.168.202.0/24   *[BGP/170] 00:59:11, localpref 100
                      AS path: 65113 I, validation-state: unverified
                    > to 192.168.191.3 via reth1.991
192.168.202.61/32  *[BGP/170] 00:59:11, localpref 100
                      AS path: 65113 I, validation-state: unverified
                    > to 192.168.191.3 via reth1.991
192.168.202.62/32  *[BGP/170] 00:59:11, localpref 100
                      AS path: 65113 I, validation-state: unverified
                    > to 192.168.191.3 via reth1.991
192.168.203.0/24   *[BGP/170] 00:59:11, localpref 100
                      AS path: 65113 I, validation-state: unverified
                    > to 192.168.191.3 via reth1.991
192.168.203.61/32  *[BGP/170] 00:15:09, localpref 100
                      AS path: 65113 I, validation-state: unverified
                    > to 192.168.191.3 via reth1.991
192.168.211.0/24   *[BGP/170] 00:34:09, localpref 100
                      AS path: 65213 I, validation-state: unverified
                    > to 192.168.192.3 via reth1.992
192.168.212.0/24   *[BGP/170] 00:34:09, localpref 100
                      AS path: 65213 I, validation-state: unverified
                    > to 192.168.192.3 via reth1.992
192.168.221.0/24   *[BGP/170] 00:25:07, localpref 100
                      AS path: 65313 65300 65322 I, validation-state: unverified
                    > to 192.168.193.3 via reth1.993
192.168.222.0/24   *[BGP/170] 00:25:07, localpref 100
                      AS path: 65313 65300 65322 I, validation-state: unverified
                    > to 192.168.193.3 via reth1.993
```

2. Verify the routes on Spine 1 and Spine 2. The SRX cluster advertises the 192.168.0.0/16 summary route to the spine devices on all the VRFs. All inter-VRF traffic and DCI traffic goes through the SRX chassis cluster.

```
user@spine1> show route 192.168.0.0
JNPR_1_VRF.inet.0: 19 destinations, 23 routes (19 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
192.168.0.0/16     *[BGP/170] 01:05:15, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.191.1 via irb.991

JNPR_2_VRF.inet.0: 13 destinations, 16 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.0.0/16     *[BGP/170] 00:40:12, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.192.1 via irb.992

WAN_VRF.inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.0.0/16     *[BGP/170] 01:04:59, localpref 100
                      AS path: 65200 I, validation-state: unverified
                    >  to 192.168.193.1 via irb.993
```

3. Verify the routes on Spine 3 and Spine 4. The DC2 spine devices receive the aggregate route from the WAN VRFs on the DC1 spine devices. All traffic between the two data centers is routed through the SRX chassis cluster.

```
user@spine3> show route 192.168.0.0
inet.0: 24 destinations, 26 routes (21 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.0.0/16      [BGP ] 00:11:47
                      AS path: 65300 65313 65200 I, validation-state: unverified
                    >  to 192.168.100.10 via et-0/0/30.0
```

You have connected your collapsed spine data center networks with DCI.

Optional Configurations: DHCP Relay and Multicast

IN THIS SECTION

Use these examples to configure optional DHCP relay and multicast forwarding on your collapsed spine data center architecture.

Configure DHCP Relay (Optional)

IN THIS SECTION

Requirements

- DHCP server.
- The devices that you configured in "How to Configure a Collapsed Spine with EVPN Multihoming" on page 12.

Overview

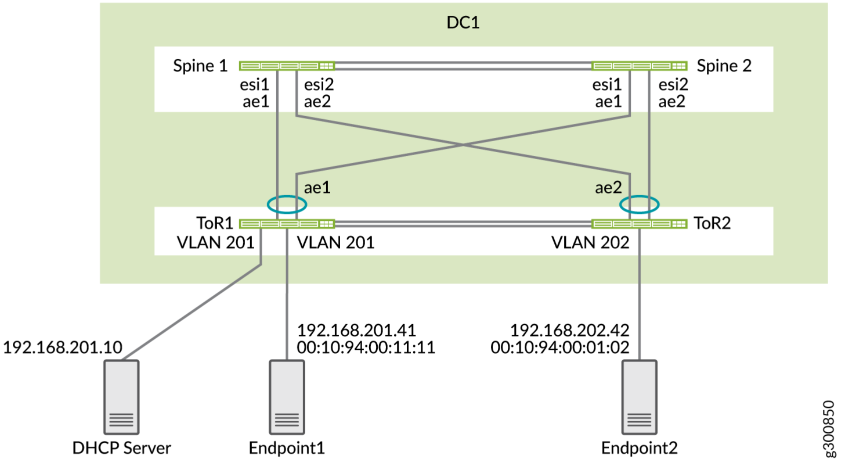
IN THIS SECTION

Use this section to configure the spine switches to relay the DHCP requests to the DHCP server. Enable DHCP relay in a routing instance with the `forward-only` option. The `forward-only` option ensures that DHCP packets are forwarded on the switch but that no DHCP server client bindings are created.

Topology

The DHCP server can be located anywhere in the data center or in a different data center. In this case, the DHCP server is connected to one of the ToR switches in DC1 and the IP address of the DHCP server is 192.168.201.10. The DHCP relay topology is shown in Figure 22 on page 82.

Figure 22: DHCP Relay Topology



Configuration

IN THIS SECTION

Configure Spine 1

Step-by-Step Procedure

1. Configure the DHCP relay on the first routing instance.

```
set routing-instances JNPR_1_VRF forwarding-options dhcp-relay forward-only
set routing-instances JNPR_1_VRF forwarding-options dhcp-relay server-group Server_Group1 192.168.201.10
set routing-instances JNPR_1_VRF forwarding-options dhcp-relay group Relay_Group1 active-server-group Server_Group1
set routing-instances JNPR_1_VRF forwarding-options dhcp-relay group Relay_Group1 route-suppression destination
set routing-instances JNPR_1_VRF forwarding-options dhcp-relay group Relay_Group1 interface irb.201
set routing-instances JNPR_1_VRF forwarding-options dhcp-relay group Relay_Group1 interface irb.202
set routing-instances JNPR_1_VRF forwarding-options dhcp-relay group Relay_Group1 interface irb.203
```

2. Configure the DHCP relay on the second routing instance.

```
set routing-instances JNPR_2_VRF forwarding-options dhcp-relay forward-only
set routing-instances JNPR_2_VRF forwarding-options dhcp-relay server-group Server_Group1 192.168.201.10
set routing-instances JNPR_2_VRF forwarding-options dhcp-relay group Relay_Group1 active-server-group Server_Group1
set routing-instances JNPR_2_VRF forwarding-options dhcp-relay group Relay_Group1 route-suppression destination
set routing-instances JNPR_2_VRF forwarding-options dhcp-relay group Relay_Group1 interface irb.211
set routing-instances JNPR_2_VRF forwarding-options dhcp-relay group Relay_Group1 interface irb.212
set routing-instances JNPR_2_VRF forwarding-options dhcp-relay group Relay_Group1 interface irb.213
```

3. Verify the DHCP relay on Spine 1.

```
user@spine1> show dhcp relay statistics routing-instance JNPR_1_VRF
Packets dropped:
    Total                      50741
    Invalid server address     0
    dhcp-service total         50738

Messages received:
    BOOTREQUEST                3
    DHCPDECLINE                0
    DHCPDISCOVER               1
    DHCPINFORM                 0
    DHCPRELEASE                0
    DHCPREQUEST                2
    DHCPLEASEACTIVE            0
    DHCPLEASEUNASSIGNED        0
    DHCPLEASEUNKNOWN           0
    DHCPLEASEQUERYDONE         0

Messages sent:
    BOOTREPLY                  2
    DHCPOFFER                  0
    DHCPACK                    2
    DHCPNAK                    0
    DHCPFORCERENEW             0
    DHCPLEASEQUERY             0
    DHCPBULKLEASEQUERY         0
```

Configure Spine 2

Step-by-Step Procedure

1. Configure the DHCP relay on the first routing instance.

```
set routing-instances JNPR_1_VRF forwarding-options dhcp-relay forward-only
set routing-instances JNPR_1_VRF forwarding-options dhcp-relay forward-only-replies
set routing-instances JNPR_1_VRF forwarding-options dhcp-relay server-group Server_Group1 192.168.201.10
set routing-instances JNPR_1_VRF forwarding-options dhcp-relay group Relay_Group1 active-server-group Server_Group1
set routing-instances JNPR_1_VRF forwarding-options dhcp-relay group Relay_Group1 route-suppression destination
set routing-instances JNPR_1_VRF forwarding-options dhcp-relay group Relay_Group1 interface irb.201
set routing-instances JNPR_1_VRF forwarding-options dhcp-relay group Relay_Group1 interface
irb.202
set routing-instances JNPR_1_VRF forwarding-options dhcp-relay group Relay_Group1 interface
irb.203
```

2. Configure the DHCP relay on the second routing instance.

```
set routing-instances JNPR_2_VRF forwarding-options dhcp-relay forward-only
set routing-instances JNPR_2_VRF forwarding-options dhcp-relay forward-only-replies
set routing-instances JNPR_2_VRF forwarding-options dhcp-relay server-group Server_Group1
192.168.201.10
set routing-instances JNPR_2_VRF forwarding-options dhcp-relay group Relay_Group1 active-
server-group Server_Group1
set routing-instances JNPR_2_VRF forwarding-options dhcp-relay group Relay_Group1 route-
suppression destination
set routing-instances JNPR_2_VRF forwarding-options dhcp-relay group Relay_Group1 interface
irb.211
set routing-instances JNPR_2_VRF forwarding-options dhcp-relay group Relay_Group1 interface
irb.212
set routing-instances JNPR_2_VRF forwarding-options dhcp-relay group Relay_Group1 interface
irb.213
```

3. Verify the DHCP relay on Spine 2.

```
user@spine2> show dhcp relay statistics routing-instance JNPR_1_VRF
Packets dropped:
    Total                       50741
    Invalid server address      0
    dhcp-service total          50738

Messages received:
    BOOTREQUEST                 3
    DHCPDECLINE                 0
    DHCPDISCOVER                1
    DHCPINFORM                  0
    DHCPRELEASE                 0
    DHCPREQUEST                 2
    DHCPLEASEACTIVE             0
    DHCPLEASEUNASSIGNED         0
    DHCPLEASEUNKNOWN            0
    DHCPLEASEQUERYDONE          0
```

```
Messages sent:
    BOOTREPLY                2
    DHCPOFFER                0
    DHCPACK                  2
    DHCPNAK                  0
    DHCPFORCERENEW           0
    DHCPLEASEQUERY           0
    DHCPBULKLEASEQUERY       0
```

Configure Multicast for Intra-VNI Traffic (Optional)

IN THIS SECTION

Requirements

- The devices that you configured in "How to Configure a Collapsed Spine with EVPN Multihoming" on page 12.

Overview

IN THIS SECTION

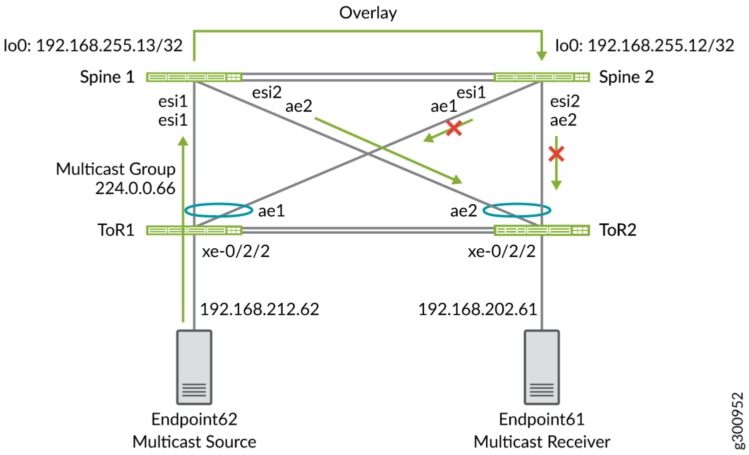
Use this section to configure your collapsed spine architecture to allow intra-VNI multicast traffic. The multicast source and receivers are part of the same VLAN.

NOTE: The collapsed spine architecture with QFX5120 spine switches does not support inter-VNI multicast traffic.

Topology

This section includes the configuration for two multicast groups. The first is multicast group 224.0.0.66. As shown in Figure 23 on page 87, the multicast source is Endpoint 62. The multicast receiver, Endpoint 61, is connected to ToR 2. Both the source and receiver are in the same VLAN, so traffic between them is intra-VNI multicast traffic.

Figure 23: Multicast Group 224.0.0.66 Topology



Configuration

IN THIS SECTION

Configure Devices

Step-by-Step Procedure

1. Enable IGMP snooping for all VLANs on both spine switches.

   ```
   set protocols igmp-snooping vlan VLAN-201 immediate-leave
   set protocols igmp-snooping vlan VLAN-202 immediate-leave
   set protocols igmp-snooping vlan VLAN-211 immediate-leave
   set protocols igmp-snooping vlan VLAN-212 immediate-leave
   ```

2. Configure the ToR switches with IGMP snooping for all VLANs.

   ```
   set protocols igmp-snooping vlan VLAN-201 immediate-leave
   set protocols igmp-snooping vlan VLAN-201 interface ae1.0 multicast-router-interface
   set protocols igmp-snooping vlan VLAN-202 immediate-leave
   set protocols igmp-snooping vlan VLAN-202 interface ae1.0 multicast-router-interface
   set protocols igmp-snooping vlan VLAN-211 immediate-leave
   set protocols igmp-snooping vlan VLAN-211 interface ae1.0 multicast-router-interface
   set protocols igmp-snooping vlan VLAN-212 immediate-leave
   set protocols igmp-snooping vlan VLAN-212 interface ae1.0 multicast-router-interface
   ```

Verification for Multicast Group 224.0.0.66

Step-by-Step Procedure

1. Verify IGMP snooping membership on ToR 2.

   ```
   user@tor2> show igmp snooping membership
   Instance: default-switch

   Vlan: VLAN-201

   Learning-Domain: default
   Interface: ae1.0, Groups: 0

   Vlan: VLAN-202

   Learning-Domain: default
   ```

```
Interface: ae1.0, Groups: 0

Learning-Domain: default
Interface: xe-0/2/2.0, Groups: 1
    Group: 225.0.0.66
        Group mode: Exclude
        Source: 0.0.0.0
        Last reported by: 192.168.202.61
        Group timeout:     177 Type: Dynamic

Vlan: VLAN-211

Learning-Domain: default
Interface: ae1.0, Groups: 0

Vlan: VLAN-212

Learning-Domain: default
Interface: ae1.0, Groups: 0
```

2. Verify IGMP snooping membership on Spine 1.

```
user@spine1> show igmp snooping evpn membership detail
Instance: default-switch

Vlan: VLAN-201, EVPN-Core-NH: 524301

Learning-Domain: default
Interface: ae2.0, Groups: 0
Interface: ae1.0, Groups: 0

Vlan: VLAN-202, EVPN-Core-NH: 524303

Learning-Domain: default
Interface: ae2.0, Groups: 1
    Group: 225.0.0.66
        Group mode: Exclude
        Source: 0.0.0.0
        Type: Local
Interface: ae1.0, Groups: 0

Vlan: VLAN-211, EVPN-Core-NH: 524307
```

```
Learning-Domain: default
Interface: ae2.0, Groups: 0
Interface: ae1.0, Groups: 0

Vlan: VLAN-212, EVPN-Core-NH: 524298

Learning-Domain: default
Interface: ae2.0, Groups: 0
Interface: ae1.0, Groups: 0
```

3. Verify IGMP snooping membership on Spine 2.

```
user@spine2> show igmp snooping evpn membership detail
Instance: default-switch

Vlan: VLAN-201, EVPN-Core-NH: 524298

Learning-Domain: default
Interface: ae2.0, Groups: 0
Interface: ae1.0, Groups: 0

Vlan: VLAN-202, EVPN-Core-NH: 524300

Learning-Domain: default
Interface: ae2.0, Groups: 1
    Group: 225.0.0.66
        Group mode: Exclude
        Source: 0.0.0.0
        Type: Remote
Interface: ae1.0, Groups: 0

Vlan: VLAN-211, EVPN-Core-NH: 524304

Learning-Domain: default
Interface: ae2.0, Groups: 0
Interface: ae1.0, Groups: 0

Vlan: VLAN-212, EVPN-Core-NH: 524306

Learning-Domain: default
```

```
Interface: ae2.0, Groups: 0
Interface: ae1.0, Groups: 0
```

4. Verify the designated forwarder on Spine 1. The output shows that Spine 1 is the designated forwarder for all Ethernet segments.

```
user@spine1> show evpn instance designated-forwarder
Instance: default-switch
  Number of ethernet segments: 12
    ESI: 00:00:00:00:00:00:00:00:01:01
      Designated forwarder: 192.168.255.13
    ESI: 00:00:00:00:00:00:00:00:01:02
      Designated forwarder: 192.168.255.13
    ESI: 00:00:00:00:00:00:00:00:01:11
      Designated forwarder: 192.168.255.13
    ESI: 00:00:00:00:00:00:00:00:01:12
      Designated forwarder: 192.168.255.13
```

5. Verify the multicast traffic flow on Spine 1.

   Based on LAG hashing, ToR 1 sends the multicast traffic to Spine 1. The traffic reaches Spine 1 on the AE1 interface. Spine 1 forwards this traffic through AE2 based on IGMP group membership.

```
user@spine1> monitor interface traffic detail
Interface    Link  Input packets        (pps)     Output packets        (pps)
ae1           Up     6742041146    (2946565)             3924          (1)
ae2           Up           2536          (0)       6741411465    (2948357)
```

6. Verify the multicast traffic flow on Spine 2.

   Based on LAG hashing, ToR 1 does not send the multicast traffic to Spine 2. Spine 2 receives this traffic from Spine 1 through the overlay, but it drops this traffic and does not forward it to AE2.

```
user@spine2> monitor interface traffic detail
Interface    Link  Input packets        (pps)     Output packets        (pps)
ae1           Up           2547          (0)             3762          (1)
ae2           Up           2462          (0)             3769          (1)
```

Verification for Multicast Group 224.0.0.65

Step-by-Step Procedure

1. Verify the designated forwarder on Spine 1. The output shows that Spine 1 is still the designated forwarder for all the Ethernet segments.

```
user@spine1> show evpn instance designated-forwarder
Instance: default-switch
  Number of ethernet segments: 12
    ESI: 00:00:00:00:00:00:00:00:01:01
      Designated forwarder: 192.168.255.13
    ESI: 00:00:00:00:00:00:00:00:01:02
      Designated forwarder: 192.168.255.13
    ESI: 00:00:00:00:00:00:00:00:01:11
      Designated forwarder: 192.168.255.13
    ESI: 00:00:00:00:00:00:00:00:01:12
      Designated forwarder: 192.168.255.13
```

2. Verify the multicast traffic flow on Spine 1.

   Based on LAG hashing, ToR 1 does not send the multicast traffic to Spine 1, so there is no incoming traffic for this multicast group on Spine 1. Spine 1 receives the multicast stream from Spine 2 through the overlay. Spine 1 drops the traffic and does not forward it to AE2 because Spine 1 and Spine 2 are part of the same Ethernet segments for AE1 and AE2.

```
user@spine1> monitor interface traffic detail
Interface     Link  Input packets        (pps)     Output packets        (pps)
et-0/0/51      Up           4750     (2947862)              4741         (3)
ae1            Up           1027           (1)              1543         (1)
ae2            Up            991           (0)              1537         (1)
```

3. Verify the multicast traffic flow on Spine 2.

   Based on LAG hashing, ToR 1 sends the multicast traffic to Spine 2. Spine 2 is not the designated forwarder for the Ethernet segments, but Spine 2 still forwards this traffic to receivers on AE2 based on the local bias rules for multicast forwarding. See Overview of Selective Multicast Forwarding for more information about the EVPN multicast forwarding rules.

```
user@spine2> monitor interface traffic detail
Interface     Link  Input packets        (pps)     Output packets        (pps)
```

```
et-0/0/51     Up            4603          (3)        3458201844    (2948641)
ae1           Up      2985999842    (2948821)              1542          (1)
ae2           Up            1009          (0)        2986000009    (2947024)
```

You have successfully configured multicast traffic forwarding on your network.