JUNIPER NETWORKS | Engineering Simplicity

Network Configuration Example

BGP Unnumbered EVPN Fabric

IN THIS GUIDE

About the BGP Unnumbered EVPN Fabric NCE

IN THIS SECTION

Use Case	Use the unnumbered BGP automatic peering feature to quickly deploy an IPv6 underlay for an EVPN-VXLAN data center (DC) fabric.
Audience	Network administrator, engineer, operators, and architects who want to understand how to deploy a BGP unnumbered Clos fabric underlay for a DC fabric.

Knowledge Level	General familiarity with EVPN-VXLAN data center network architectures and underlay and overlay routing. See the related topics section for background information about EVPN-VXLAN technology.
Benefits	• Provides IPv6 based E-BGP peer auto-discovery without explicit neighbor configuration. • Eliminates the need to manage and to configure IPv6 addresses on fabric interfaces, BGP peer definitions, routing policy, and AS number configurations. • Reduces configuration efforts and is less prone to provisioning errors.
Products Used	• Junos OS 21.2R1 or later, and on Junos OS Evolved release 21.3R1 or later. • QFX Series switches as leaf and spine devices. • This NCE was validated on vQFX platforms running Junos OS Release 21.4R1. For the full list of supported devices and OS versions, see Supported Juniper Devices.

This guide demonstrates how to deploy and verify the BGP unnumbered peering (also referred to BGP auto discovery or BGP auto-peering). Juniper Networks supports BGP unnumbered peering starting in Junos OS Release 21.1R1. This feature allows BGP to auto-discover and to create peer neighbor sessions using the link-local IPv6 addresses of directly connected neighbors.

The BGP unnumbered peering solution uses Junos OS support for the following RFC's:

- RFC 5549: Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop
- RFC 4861: Neighbor Discovery for IP version 6 (IPv6)
- RFC 2462: IPv6 Stateless Address Autoconfiguration

Overview

Today many enterprises and hyper-scale data centers use BGP as the underlay routing protocol. Unlike traditional IGPs such as OSPF and IS-IS, BGP typically requires that you explicitly configure peering, autonomous system (AS) numbers, and routing policies to control route exchanges.

Many operators are still relatively inexperienced with IPv6. Using BGP unnumbered peering, which dynamically discovers IPV6 neighbors, reduces the burden of manually configuring an IPv6 underlay in an EVPN-VXLAN DC fabric. Junos OS builds on the baseline IPv6 functionality by supporting BGP group configuration. BGP group configuration provides support for dynamic peering parameters (such as allowed remote AS numbers) used to support an unnumbered IPv6 fabric.

Use Case: Manually Configuring an EVPN-VXLAN Fabric

You can configure an EVPN-VXLAN DC fabric by manually configuring the fabric or by using the BGP unnumbered peering feature. This use case describes the complexity of manually configuration the fabric and why using the BGP unnumbered peering feature provides a much easier solution.

Consider a simple two-tier data center. This modest size fabric consists of four spine devices and 32 leaf devices. Each spine device has 32 links which attach to the leaf devices and each leaf has two fabric links, one for each spine device.

In this manual configuration, you first need to need to assign the IP addresses for the network. For this fabric you'll need to configure 4*32=128 IPv6 IP addresses. Each network requires two host address assignments.

Next, you configure the BGP peers and their associated AS numbers. For each end of every fabric link, you need one BGP peering session. In our example fabric, this calculation equates to a total of 4*32*2=256 BGP peer definitions, each of which requires a unique peering IP and remote AS number.

Manually defining 256 BGP peerings can be cumbersome is also prone to error. In a complex fabric, a simple misconfiguration can be difficult to isolate. Let's say that the fabric supports 128 leaf devices. You must now configure 4*128=512 IP IPv6 networks. It is clear from the math that the complexity of manually provisioning a large fabric quickly becomes a burden. Also, for IPv4 fabrics, an often overlooked factor is the large number of IPv4 addresses consumed by the underlay. In many networks, IPv4 addressing space is at a premium.

In contrast, BGP unnumbered peering requires no routable IP network assignments on underlay links. All BGP peering in the underlay uses only link-local IP's. Using link-local IP's means less configuration, less complexity, smaller routing tables, and IP address preservation.

Table 1 shows the configuration required for a simple two spine and two leaf EVPN-VXLAN fabric. Specifically, it compares the configuration of two underlay EBGP peers on a leaf device using manual peering versus unnumbered peering.

When comparing the configurations, consider not only the number of configuration statements required but also their relative complexity. As show in the table below, the manual configuration (IPv4) requires that you configuring the IP addresses, the remote peer IP address, and AS numbers for the remote peers. In contrast, with BGP unnumbered peering, you only need to define the interface names. No routable IP address assignments are required in the underlay. In addition, BGP unnumbered peering automatically configures the BGP neighbor IP and the remote AS number.

Table 1: Differences Between Manual Peering and Unnumbered Peering

Manual Configuration (IPv4)	BGP Unnumbered Peering
set interfaces xe-0/0/0 unit 0 family inet address 10.0.1.2/30 set interfaces xe-0/0/1 unit 0 family inet address 10.0.1.6/30 set protocols bgp group manual_underlay family inet unicast set protocols bgp group manual_underlay local-as 65510 set protocols bgp group manual_underlay neighbor 10.0.1.1 peer-as 65001 set protocols bgp group manual_underlay neighbor 10.0.1.5 peer-as 65002	set interfaces xe-0/0/0 unit 0 family inet6 set interfaces xe-0/0/1 unit 0 family inet6 set policy-options as-list as-list members [65000-6510] set protocols bgp group bgp_unnum family inet6 unicast set protocols bgp group bgp_unnum local-as 65003 set protocols bgp group bgp_unnum dynamic-neighbor FABRIC peer-bgp_unnumovery family inet6 ipv6-nd set protocols bgp group bgp_unnum dynamic-neighbor FABRIC peer-bgp_unnumovery interface xe-0/0/0 set protocols bgp group bgp_unnum dynamic-neighbor FABRIC peer-bgp_unnumovery interface xe-0/0/1 set protocols bgp group bgp_unnum peer-as-list as-list

Table 1: Differences Between Manual Peering and Unnumbered Peering *(Continued)*

Manual Configuration (IPv4)	BGP Unnumbered Peering
Disadvantages • Not configuration group friendly; IP addresses vary by interface and peer. • Routable address assignments are required on all fabric interfaces. • Requires manually configuring IP addresses, the remote peer IP address, and AS number. • This fabric is based on IPv4. Even though the IPv4 underlay supports both IPv4 and IPv6 workloads in the underlay, many operators are not familiar with IPv6.	**Advantages** • Configuration group friendly; easy to apply family inet6 and enable router advertisements. • No address assignments to underlay links; conserves IP addresses and reduces complexity. • Further reduces complexity by using automatic discovery. • Automatically configures the BGP peering IP and the remote AS number. • Eliminates the need to assign IPv6 addresses. • Supports both IPv4 and IPv6 workloads in the overlay.

NOTE: You can easily use a configuration group to apply the `inet6` family. Using configuration groups greatly reduces the time it takes to configure a large fabric. We show you a sample of a configuration group later in this NCE.

Next, we'll show you how to configure a BGP unnumbered peering on an VXLAN-EVPN fabric.

Configure BGP Unnumbered EVPN Fabric

SUMMARY

Quickly configure the BGP unnumbered peering feature.

IN THIS SECTION

Overview

Starting with Junos OS Release 21.1R1, Juniper supports BGP unnumbered (auto-discovered) peering. This feature automatically configures BGP peering and related route exchange in an EVPN-VXLAN fabric underlay. The underlay's E-BGP peering uses dynamically discovered link-local IPv6 addresses of directly connected fabric devices.

When configuring BGP unnumbered peering, you only need to configure a minimum number configuration statements. These configuration statements are used to quickly bootstrap an IPv6 based underlay. This underlay supports the EVPN-VXLAN overlay. Although the underlay uses native IPv6, it also supports IPv4 routes with IPv6 next-hops (RFC5549) . This means that the underlay supports both IPv4 and IPv6 workloads and their related virtual networks (VNs) in the EVPN-VXLAN overlay. A complete discussion of IPv6 stateless auto-configuration is beyond the scope of this document.

The following list highlights the key features of BGP unnumbered peering:

- Automatically configures stateless link-local IPv6 addresses on all IPv6 enabled interfaces.
- Supports a list of allowed AS numbers to simplify peering to remote fabric devices.
- Uses IPv6 Router Advertisements (RAs) that provides dynamic discovery of directly attached neighbors.
- Uses IPv6 neighbor discovery to resolve the neighbor's link-local IP to the corresponding MAC address to facilitate link level communications.
- The local end uses the discovered peer link-local and MAC addresses to send a BGP open message to directly attached neighbors. This open message contains the local peer's AS number. The remote peer matches this against its list of allowed AS numbers to decide if the session is allowed. Likewise, the local peer matches on the remote peer's AS number, as returned in the remote peer's open message.
- Provides you with a simple BGP policy that advertises all directly connected networks (at a minimum, the loopback address of each fabric device must be advertised).
- Uses the default E-BGP policy to re-advertise the routes learned from other fabric devices.
- Uses the BGP AS path length is to prevent loops and to provide optimum route selection for ECMP load balancing.
- Because the underlay provides loopback reachability, you can easily add an EVPN-VXLAN overlay.

NOTE: BGP unnumbered peering only supports EBGP. Multihop EBGP and IBGP are not supported.

Requirements

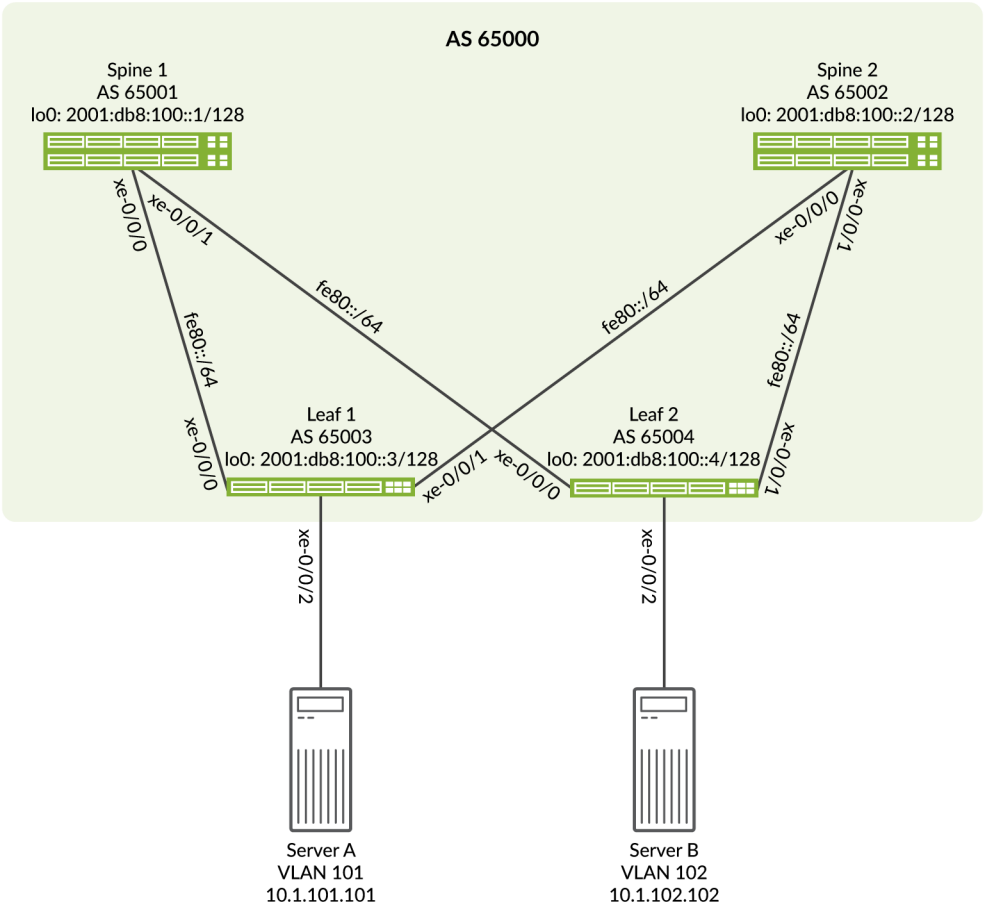
The requirements for staging this NCE are:

- Junos OS or Junos OS Evolved release 21.1R1 or higher. For the full list of supported devices and OS versions, see Supported Juniper Devices.

- •

Topology

The following topology shows a a simple two spine and two leaf topology. Refer to this topology when performing the "Step-by-Step Configuration" on page 7 in the next section.. Some points to note about the topology include:



- The fabric underlay is pure IPv6. This includes the device loopback addresses.
- The attached workloads are IPv4 based. This is to demonstrate support for IPv4 over IPv6 (RFC 5549). IPv6 workloads are also supported in the overlay. You may assume these are bare metal servers (BMS) that are preconfigured with the IP addressing shown. The VLAN membership is used to map these workloads into overlay VNs that are isolated by VXLAN encapsulation. The access interfaces are untagged in this example.
- You expect to have routed connectivity between the server workloads when you add an overlay. The overlay can be either Centrally-routed bridging (CRB) or Edge-routed bridging (ERB) as desired. The BGP unnumbered underlay supported either type of overlay. See EVPN Primer for information on EVPN-VXLAN overlay architecture options.

Step-by-Step Configuration

IN THIS SECTION

This section provides step-by-step instructions on how to configure BGP unnumbered peering. Because the configuration is similar on all the fabric devices, the configuration for the Leaf 1 device is only shown here.

To get you up and running quickly, we've provided configurations for the fabric devices in"Quick Configurations" on page 11.

Step-by-Step Configuration: Leaf 1

1. Enable the `inet6` family on all fabric interfaces. These fabric interfaces attach the leaf to the spine device. The `inet6` family provides support for IPv6 stateless auto-configuration and neighbor discovery. To support IPv4 workloads you must also add the `inet` family.

```
set interfaces xe-0/0/0 unit 0 family inet6
set interfaces xe-0/0/0 unit 0 family inet
set interfaces xe-0/0/1 unit 0 family inet6
set interfaces xe-0/0/1 unit 0 family inet
```

NOTE: If you have a large number of fabric interfaces, consider using a configuration group. A configuration group applies the `inet6` family to the first ten 10GE interfaces. The configuration group does not create the interface, For this configuration to work, the interface must already be present in the configuration (for example , an interface description or an interface parameter such as an IPv4 address).

```
root@leaf1# show groups v6-unnum
interfaces {
    "<xe-0/0/[0-10]>" {
        unit 0 {
            family inet;
            family inet6;
        }
    }
}
```

```
root@leaf1# show apply-groups
apply-groups v6-unnum;
```

2. Create the loopback interface and configure the IPv6 address. The loopback address is used to support BGP peering when you add the EVPN-VXLAN overlay.

```
set interfaces lo0 unit 0 family inet6 address 2001:db8:100::3/128
```

3. Configure a policy that specifies the list of BGP AS numbers you want to allow for dynamic BGP peering.

```
set policy-options as-list a-list members [65000-65100]
```

NOTE: Junos OS support for an AS number list is a key component of the BGP unnumbered peering solution. The AS number list simplifies dynamic BGP peering because it eliminates the need to explicitly list the AS number associated with each directly connected peer.

4. Configure a per-packet load balancing policy. A load balancing policy allows multiple equal-cost next-hops to be installed in the forwarding table. This provides rapid fail-over to alternative equal-cost fabric hops in the vent of a link failure.

```
set policy-options policy-statement load-balancing-policy then load-balance per-packet
```

5. Apply the per-packet load balancing policy to the forwarding table.

```
set routing-options forwarding-table export load-balancing-policy
```

6. Configure a policy to advertise direct routes. Because link-local subnets are not exported, in this example, this policy advertises only the loopback address. You'll use this same loopback address later when you configure BGP peering in the overlay.

```
set policy-options policy-statement DIRECT-RTS from protocol direct
set policy-options policy-statement DIRECT-RTS then accept
```

7. Configure the Router ID (RID). Because this is an example of a native IPv6 fabric you must ensure there is an IPv4 formatted router ID available. In most cases, the RID is automatically derived from the loopback address, but in this example, the loopback is IPv6 only.

```
set routing-options router-id 10.0.0.4
```

8. Configure router-advertisement on all fabric interfaces. When RA is enabled the interface periodically sends RA messages. RA messages are used to discover the remote neighbor's link-local IP, which in turn kicks off neighbor discovery and the rest of the dynamic peering process.

```
set protocols router-advertisement interface xe-0/0/0
set protocols router-advertisement interface xe-0/0/1
```

9. Configure a BGP group to support unnumbered peering. A policy that exports direct routes is applied to this group. You must include all fabric interfaces in this group and you must enable them for `peer-auto-discovery`. This group is linked to the policy that defines the AS numbers allowed for dynamic peering.

   You enable load balancing over multiple paths and multiple AS numbers. Recall that in this fabric each node uses a unique AS number. Multipath load balancing (ECMP) to multiple AS numbers enables fast fail-over by allowing the RIB to install multiple next hops that point to these different AS numbers.

```
set protocols bgp group auto-disc family inet6 unicast
set protocols bgp group auto-disc export DIRECT-RTS
set protocols bgp group auto-disc local-as 65003
set protocols bgp group auto-disc dynamic-neighbor FABRIC peer-auto-discovery family inet6 ipv6-nd
set protocols bgp group auto-disc dynamic-neighbor FABRIC peer-auto-discovery interface xe-0/0/0
set protocols bgp group auto-disc dynamic-neighbor FABRIC peer-auto-discovery interface xe-0/0/1
set protocols bgp group auto-disc peer-as-list a-list
set protocols bgp group auto-disc multipath multiple-as
```

Results

Recall that all fabric devices have similar configurations. For brevity, only the configuration delta for leaf 1 is shown.

```
user@leaf-1> show configuration
system {
    host-name leaf1;
}
interfaces {
    xe-0/0/0 {
```

```
        unit 0 {
            family inet;
            family inet6;
        }
    }
    xe-0/0/1 {
        unit 0 {
            family inet;
            family inet6;
        }
    }
    lo0 {
        unit 0 {
            family inet6 {
                address 2001:db8:100::3/128;
            }
        }
    }
}
policy-options {
    policy-statement DIRECT-RTS {
        from protocol direct;
        then accept;
    }
    policy-statement load-balancing-policy {
        then load-ballance per-packet;
    }
    as-list a-list members 65000-65100;
}
routing-options {
    router-id 10.0.0.3;
    forwarding-table {
        export load-balancing-policy;
    }
}
protocols {
    router-advertisement {
        interface xe-0/0/0.0;
        interface xe-0/0/1.0;
    }
    bgp {
        group auto-disc {
            family inet6 {
                unicast;
            }
            export DIRECT-RTS;
            local-as 65003;
```

```
            multipath;
            dynamic-neighbor FABRIC {
                peer-auto-discovery {
                    family inet6 {
                        ipv6-nd;
                    }
                    interface xe-0/0/0.0;
                    interface xe-0/0/1.0;
                }
            }
            peer-as-list a-list;
        }
    }
}
```

Quick Configurations

IN THIS SECTION

To get you up and running quickly, we've provided you with quick configurations for each node in the topology. Make sure that you edit these configurations to match your fabric specifics and paste them into the corresponding fabric node.

CLI Quick Configuration

NOTE: The device configurations omit the management interface, static routes, system logging, system services, and user login information. These parts of the configuration vary by location and are not directly related to BGP unnumbered peering functionality.

Edit the following commands as needed for the specifics of your environment and paste them into the related fabric device's terminal window when in configuration mode at the `[edit]` hierarchy:

The quick configuration for leaf 1:

```
set system host-name leaf1
set interfaces xe-0/0/0 unit 0 family inet
set interfaces xe-0/0/1 unit 0 family inet
set interfaces xe-0/0/0 unit 0 family inet6
set interfaces xe-0/0/1 unit 0 family inet6
set interfaces lo0 unit 0 family inet6 address 2001:db8:100::3/128

set policy-options as-list a-list members [65000-65100]
set policy-options policy-statement DIRECT-RTS from protocol direct
set policy-options policy-statement DIRECT-RTS then accept
set policy-options policy-statement load-balancing-policy then load-balance per-packet

set protocols router-advertisement interface xe-0/0/0
set protocols router-advertisement interface xe-0/0/1
set protocols bgp group auto-disc family inet6 unicast
set protocols bgp group auto-disc export DIRECT-RTS
set protocols bgp group auto-disc local-as 65003
set protocols bgp group auto-disc dynamic-neighbor FABRIC peer-auto-discovery family inet6 ipv6-nd
set protocols bgp group auto-disc dynamic-neighbor FABRIC peer-auto-discovery interface xe-0/0/0
set protocols bgp group auto-disc dynamic-neighbor FABRIC peer-auto-discovery interface xe-0/0/1
set protocols bgp group auto-disc peer-as-list a-list

set protocols bgp group auto-disc multipath multiple-as

set routing-options forwarding-table export load-balancing-policy
set routing-options router-id 10.0.0.3
```

The quick configuration for leaf 2:

```
set system host-name leaf2
set interfaces xe-0/0/0 unit 0 family inet
set interfaces xe-0/0/1 unit 0 family inet
set interfaces xe-0/0/0 unit 0 family inet6
set interfaces xe-0/0/1 unit 0 family inet6
set interfaces lo0 unit 0 family inet6 address 2001:db8:100::4/128

set policy-options as-list a-list members [65000-65100]
set policy-options policy-statement DIRECT-RTS from protocol direct
set policy-options policy-statement DIRECT-RTS then accept
set policy-options policy-statement load-balancing-policy then load-balance per-packet

set protocols router-advertisement interface xe-0/0/0
set protocols router-advertisement interface xe-0/0/1
```

```
set protocols bgp group auto-disc family inet6 unicast
set protocols bgp group auto-disc export DIRECT-RTS
set protocols bgp group auto-disc local-as 65004
set protocols bgp group auto-disc dynamic-neighbor FABRIC peer-auto-discovery family inet6 ipv6-nd
set protocols bgp group auto-disc dynamic-neighbor FABRIC peer-auto-discovery interface xe-0/0/0
set protocols bgp group auto-disc dynamic-neighbor FABRIC peer-auto-discovery interface xe-0/0/1
set protocols bgp group auto-disc peer-as-list a-list
set protocols bgp group auto-disc multipath multiple-as

set routing-options forwarding-table export load-balancing-policy
set routing-options router-id 10.0.0.4
```

The quick configuration for spine 1:

```
set system host-name spine1
set interfaces xe-0/0/0 unit 0 family inet
set interfaces xe-0/0/1 unit 0 family inet
set interfaces xe-0/0/0 unit 0 family inet6
set interfaces xe-0/0/1 unit 0 family inet6
set interfaces lo0 unit 0 family inet6 address 2001:db8:100::1/128

set policy-options as-list a-list members [65000-65100]
set policy-options policy-statement DIRECT-RTS from protocol direct
set policy-options policy-statement DIRECT-RTS then accept
set policy-options policy-statement load-balancing-policy then load-balance per-packet

set protocols router-advertisement interface xe-0/0/0
set protocols router-advertisement interface xe-0/0/1
set protocols bgp group auto-disc family inet6 unicast
set protocols bgp group auto-disc export DIRECT-RTS
set protocols bgp group auto-disc local-as 65001
set protocols bgp group auto-disc dynamic-neighbor FABRIC peer-auto-discovery family inet6 ipv6-nd
set protocols bgp group auto-disc dynamic-neighbor FABRIC peer-auto-discovery interface xe-0/0/0
set protocols bgp group auto-disc dynamic-neighbor FABRIC peer-auto-discovery interface xe-0/0/1
set protocols bgp group auto-disc peer-as-list a-list
set protocols bgp group auto-disc multipath multiple-as

set routing-options forwarding-table export load-balancing-policy
set routing-options router-id 10.0.0.1
```

The quick configuration for spine 2:

```
set system host-name spine2
set interfaces xe-0/0/0 unit 0 family inet
set interfaces xe-0/0/1 unit 0 family inet
```

```
set interfaces xe-0/0/0 unit 0 family inet6
set interfaces xe-0/0/1 unit 0 family inet6
set interfaces lo0 unit 0 family inet6 address 2001:db8:100::2/128

set policy-options as-list a-list members [65000-65100]
set policy-options policy-statement DIRECT-RTS from protocol direct
set policy-options policy-statement DIRECT-RTS then accept
set policy-options policy-statement load-balancing-policy then load-balance per-packet

set protocols router-advertisement interface xe-0/0/0
set protocols router-advertisement interface xe-0/0/1
set protocols bgp group auto-disc family inet6 unicast
set protocols bgp group auto-disc export DIRECT-RTS
set protocols bgp group auto-disc local-as 65002
set protocols bgp group auto-disc dynamic-neighbor FABRIC peer-auto-discovery family inet6 ipv6-nd
set protocols bgp group auto-disc dynamic-neighbor FABRIC peer-auto-discovery interface xe-0/0/0
set protocols bgp group auto-disc dynamic-neighbor FABRIC peer-auto-discovery interface xe-0/0/1
set protocols bgp group auto-disc peer-as-list a-list
set protocols bgp group auto-disc multipath multiple-as

set routing-options forwarding-table export load-balancing-policy
set routing-options router-id 10.0.0.2
```

Be sure to commit the configuration changes on all devices. Congratulations on your new IPv6 underlay!

NOTE: The configuration of the server devices is not relevant until you add the EVPN-VXLAN overlay. Until the overlay is added, these devices are isolated and unable to ping any other fabric or server devices. Part of adding the overlay involves configuring the access port parameters, such as VLAN ID and tagged vs. untagged, to be compatible with the attached device.

For now, it's sufficient to assume the server devices are configured with IPv4 addressing as shown in the topology diagram and that they are configured for untagged (access interface) operation.

In the next section, we show you how to verify proper operation of the BGP unnumbered underlay.

Validate BGP Unnumbered EVPN Fabric

SUMMARY

Follow these steps to verify the unnumbered underlay with automatic BGP peering is working properly.

IN THIS SECTION

Use these commands and sample output to confirm the proper operation of your unnumbered underlay. The configuration and operation of all nodes are similar. Below we only show commands and output for leaf 1. The same commands apply to all nodes, and similar output is expected for all devices.

Overall, the main verification task is to confirm all nodes have expected BGP sessions established using link-local addresses, and that the fabric devices are properly exchanging loopback routes. We take a structured bottom-up approach that confirms each aspect needed to successfully establish BGP sessions.

Validate Fabric Interfaces

IN THIS SECTION

Purpose

Confirm fabric interfaces are up and operational.

Action

```
user@leaf1> show interfaces xe-0/0/0Physical interface: xe-0/0/0, Enabled, Physical link is Up
  Interface index: 650, SNMP ifIndex: 516
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Speed: 10Gbps, Duplex: Full-Duplex, BPDU Error: None,
  Loop Detect PDU Error: None, Ethernet-Switching Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Disabled, Media type: Fiber
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: 02:05:86:24:88:03, Hardware address: 02:05:86:24:88:03
  Last flapped   : 2022-03-14 10:56:00 PDT (6d 23:47 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None
  PCS statistics                      Seconds
    Bit errors                             0
    Errored blocks                         0
  Ethernet FEC statistics              Errors
    FEC Corrected Errors                    0
    FEC Uncorrected Errors                  0
    FEC Corrected Errors Rate               0
    FEC Uncorrected Errors Rate             0
  Interface transmit statistics: Disabled

  Logical interface xe-0/0/0.0 (Index 555) (SNMP ifIndex 540)
    Flags: Up SNMP-Traps 0x4004000 Encapsulation: ENET2
    Input packets : 47384
    Output packets: 48579
    Protocol inet, MTU: 1500
    Max nh cache: 75000, New hold nh limit: 75000, Curr nh cnt: 0, Curr new hold cnt: 0, NH drop cnt: 0
      Flags: Sendbcast-pkt-to-re
    Protocol inet6, MTU: 1500
    Max nh cache: 75000, New hold nh limit: 75000, Curr nh cnt: 1, Curr new hold cnt: 0, NH drop cnt: 0
      Flags: Is-Primary
      Addresses, Flags: Is-Preferred
        Destination: fe80::/64, Local: fe80::205:86ff:fe24:8803
```

```
user@leaf1> show interfaces terse
Interface               Admin Link Proto    Local                 Remote
gr-0/0/0                up    up
pfe-0/0/0               up    up
pfe-0/0/0.16383         up    up   inet
```

```
                                    inet6
pfh-0/0/0               up    up
pfh-0/0/0.16383         up    up    inet
pfh-0/0/0.16384         up    up    inet
xe-0/0/0.0              up    up    inet
                                    inet6    fe80::205:86ff:fe24:8803/64
xe-0/0/1                up    up
xe-0/0/1.0              up    up    inet
                                    inet6    fe80::205:86ff:fe24:8807/64
xe-0/0/2                up    up
xe-0/0/2.16386          up    up
. . .
jsrv                    up    up
jsrv.1                  up    up    inet     128.0.0.127/2
lo0                     up    up
lo0.0                   up    up    inet
                                    inet6    2001:db8:100::3
                                             fe80::205:860f:fcc1:6f00
lo0.16385               up    up    inet
lsi                     up    up
mtun                    up    up
pimd                    up    up
pime                    up    up
pip0                    up    up
tap                     up    up
vme                     up    down
vtep                    up    up
```

Meaning

The output shows that the leaf 1's fabric interfaces are operational. Also seen is the lack of an explicit IPv4 or IPv6 address. Only the `inet` and `inet6` families are configured on the interface. As a result, only IPv6 link-local IPv6 addresses are present as part of IPv6 stateless address configuration. We also note the loopback address has the expected IPv6 address assignment.

The output of the `show interfaces terse` CLI command makes it easy to verify the state and configuration of all interfaces, including the loopback address.

Validate Router Advertisements

IN THIS SECTION

Purpose

Confirm all fabric interfaces are sending and receiving IPv6 router advertisements (RAs).

Action

```
user@leaf1>  show ipv6 router-advertisement
Interface: xe-0/0/0.0
  Advertisements sent: 1512, last sent 00:01:05 ago
  Solicits sent: 1, last sent 6d 23:49:07 ago
  Solicits received: 3, last received 4d 18:18:57 ago
  Advertisements received: 503
  Solicited router advertisement unicast: Disable
  IPv6 RA Preference: DEFAULT/MEDIUM
  Advertisement from fe80::205:86ff:fe0c:dd03, heard 4d 18:15:17 ago
    Managed: 0
    Other configuration: 0
    Reachable time: 0 ms
    Default lifetime: 0 sec
    Retransmit timer: 0 ms
    Current hop limit: 64
Interface: xe-0/0/1.0
  Advertisements sent: 1523, last sent 00:02:05 ago
  Solicits sent: 1, last sent 6d 23:49:07 ago
  Solicits received: 0
  Advertisements received: 1515
  Solicited router advertisement unicast: Disable
  IPv6 RA Preference: DEFAULT/MEDIUM
  Advertisement from fe80::205:86ff:fec6:b503, heard 00:03:47 ago
    Managed: 0
    Other configuration: 0
    Reachable time: 0 ms
    Default lifetime: 1800 sec
```

```
    Retransmit timer: 0 ms
    Current hop limit: 64
```

Meaning

The output confirms that leaf 1 is sending, and receiving RAs, to and from both spine devices. The RAs correctly report the sending end's link-local address.

Verify IPv6 Neighbor Discovery

IN THIS SECTION

Purpose

Confirm the fabric devices have learned the MAC to link-local address bindings of all directly attached IPv6 neighbors using IPv6 ND.

Action

```
user@leaf1> show ipv6 neighbors
IPv6 Address                           Linklayer Address  State       Exp   Rtr  Secure
Interface
fe80::205:86ff:fe0c:dd03                02:05:86:0c:dd:03  reachable   19    yes  no
xe-0/0/0.0
fe80::205:86ff:fec6:b503                02:05:86:c6:b5:03  reachable   34    yes  no
xe-0/0/1.0
Total entries: 2
```

Meaning

The output confirms that leaf 1 has successfully resolved the MAC to link-local address of both its neighbors, that is, the two spine devices.

Verify Link-Local Connectivity

IN THIS SECTION

Purpose

Confirm that you can ping a neighbor using its link-local address.

NOTE: Because all IPv6 link-locals share the same fe80::/64 prefix you must disambiguate a link-local ping by qualifying it with the corresponding egress interface.

Action

```
user@leaf1> ping fe80::205:86ff:fe0c:dd03 interface xe-0/0/0
PING6(56=40+8+8 bytes) fe80::205:86ff:fec1:6f03 --> fe80::205:86ff:fe0c:dd03
16 bytes from fe80::205:86ff:fe0c:dd03, icmp_seq=0 hlim=64 time=117.229 ms
16 bytes from fe80::205:86ff:fe0c:dd03, icmp_seq=1 hlim=64 time=114.074 ms
^C
--- fe80::205:86ff:fe0c:dd03 ping6 statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/std-dev = 114.074/115.651/117.229/1.577 ms
```

Meaning

The ping from leaf 1 to spine 1 succeeds. This result confirms IPv6 connectivity using link-local addresses.

Verify BGP Peering

IN THIS SECTION

Purpose

Confirm that all fabric devices have established BGP peering sessions to the directly connected neighbors.

Action

```
user@leaf1> show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 1 Peers: 2 Down peers: 0
Auto-discovered peers: 2
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet6.0
                       4          4          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/Received/Accepted/
Damped...
fe80::205:86ff:fe0c:dd03%xe-0/0/0.0       65001          6          5       0       1          50 Establ
  inet6.0: 2/2/2/0
fe80::205:86ff:fec6:b503%xe-0/0/1.0       65002          6          6       0       1          51 Establ
  inet6.0: 2/2/2/0
```

Meaning

As expected, the output shows that the leaf 1 device has two BGP sessions established. The display confirms that IPv6 routes are exchanged. More on that later. If desired, show details about a BGP neighbor:

```
user@leaf1>  show bgp neighbor
Peer: fe80::205:86ff:fe0c:dd03%xe-0/0/0.0+56258 AS 65001 Local: fe80::205:86ff:fec1:6f03%xe-0/0/0.0+179 AS 65003
  Group: auto-disc             Routing-Instance: master
  Forwarding routing-instance: master
  Type: External    State: Established    Flags: <Sync PeerAsList AutoDiscoveredNdp>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ DIRECT-RTS ]
  Options: <AddressFamily Multipath LocalAS Refresh>
  Options: <MultipathAs>
  Options: <GracefulShutdownRcv>
  Address families configured: inet6-unicast
  Holdtime: 90 Preference: 170
  Graceful Shutdown Receiver local-preference: 0
  Local AS: 65003 Local System AS: 0
  Number of flaps: 1
  Last flap event: TransportError
  Peer ID: 10.0.0.1        Local ID: 10.0.0.3          Active Holdtime: 90
```

```
  Keepalive Interval: 30         Group index: 0    Peer index: 1    SNMP index: 2
  I/O Session Thread: bgpio-0 State: Enabled
  BFD: disabled, down
  Local Interface: xe-0/0/0.0
  NLRI for restart configured on peer: inet6-unicast
  NLRI advertised by peer: inet6-unicast
  NLRI for this session: inet6-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  Restart flag received from the peer: Notification
  NLRI that restart is negotiated for: inet6-unicast
  NLRI of received end-of-rib markers: inet6-unicast
  NLRI of all end-of-rib markers sent: inet6-unicast
  Peer does not support LLGR Restarter functionality
  Peer supports 4 byte AS extension (peer-as 65001)
  Peer does not support Addpath
  NLRI(s) enabled for color nexthop resolution: inet6-unicast
  Table inet6.0 Bit: 20000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              2
    Received prefixes:            2
    Accepted prefixes:            2
    Suppressed due to damping:    0
    Advertised prefixes:          2
  Last traffic (seconds): Received 1    Sent 27   Checked 56
  Input messages:  Total 7      Updates 3       Refreshes 0     Octets 342
  Output messages: Total 5      Updates 2       Refreshes 0     Octets 260
  Output Queue[1]: 0            (inet6.0, inet6-unicast)
. . .
```

Verify BGP Route Exchange

IN THIS SECTION

- Purpose | 23

Purpose

Confirm that all nodes are advertising their loopback address while learning the loopback address of other nodes.

Action

```
user@leaf1> show route protocol bgp
inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)

inet6.0: 8 destinations, 11 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2001:db8:100::1/128*[BGP/170] 00:33:43, localpref 100
                      AS path: 65001 I, validation-state: unverified
                    >  to fe80::205:86ff:fe0c:dd03 via xe-0/0/0.0
                    [BGP/170] 00:33:46, localpref 100
                      AS path: 65002 65004 65001 I, validation-state: unverified
                    >  to fe80::205:86ff:fec6:b503 via xe-0/0/1.0
2001:db8:100::2/128*[BGP/170] 00:33:46, localpref 100
                      AS path: 65002 I, validation-state: unverified
                    >  to fe80::205:86ff:fec6:b503 via xe-0/0/1.0
                    [BGP/170] 00:33:43, localpref 100
                      AS path: 65001 65004 65002 I, validation-state: unverified
                    >  to fe80::205:86ff:fe0c:dd03 via xe-0/0/0.0
2001:db8:100::4/128*[BGP/170] 00:33:46, localpref 100
                      AS path: 65002 65004 I, validation-state: unverified
                    >  to fe80::205:86ff:fec6:b503 via xe-0/0/1.0
                    [BGP/170] 00:33:43, localpref 100
                      AS path: 65001 65004 I, validation-state: unverified
                    >  to fe80::205:86ff:fe0c:dd03 via xe-0/0/0.0
```

Meaning

As expected, the output confirms that leaf 1 has learned the loopback addresses of all other fabric devices. Note that for leaf 2, it shows two equal costs paths. One path through spine 1, the other through spine 2.

If needed, you can help isolate routing problems by displaying the specific routes advertised to, or received from, a given BGP neighbor. Note that as with a ping to a link-local address, you must qualify the peer's link-local IP with the corresponding egress interface:

```
user@leaf1> show route advertising-protocol bgp fe80::205:86ff:fe0c:dd03%xe-0/0/0.0
inet6.0: 8 destinations, 10 routes (8 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
```

```
* 2001:db8:100::2/128     Self                                  65002 I
* 2001:db8:100::3/128     Self                                  I

user@leaf1> show route receive-protocol bgp fe80::205:86ff:fe0c:dd03%xe-0/0/0.0
inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)

inet6.0: 8 destinations, 10 routes (8 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 2001:db8:100::1/128     fe80::205:86ff:fe0c:dd03               65001 I
* 2001:db8:100::4/128     fe80::205:86ff:fe0c:dd03               65001 65004 I
```

Verify ECMP Load Balancing

IN THIS SECTION

Purpose

Confirm that the fabric supports ECMP load balancing. Display details about the routes leaf 1 use to reach leaf 2. You expect two equal costs routes, one through each spine device. In the below section you confirm ECMP in both the routing and forwarding tables (RIB and FIB).

Action

```
user@leaf1>  show route 2001:db8:100::4 detail
inet6.0: 8 destinations, 11 routes (8 active, 0 holddown, 0 hidden)
2001:db8:100::4/128 (2 entries, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Router, Next hop index: 0
                Address: 0xd057cc8
                Next-hop reference count: 2
                Source: fe80::205:86ff:fe0c:dd03%xe-0/0/0.0
                Next hop: fe80::205:86ff:fe0c:dd03 via xe-0/0/0.0, selected
                Session Id: 0
                Next hop: fe80::205:86ff:fec6:b503 via xe-0/0/1.0
                Session Id: 0
                State: <Active Ext>
                Peer AS: 65001
```

```
                Age: 1:39:21
                Validation State: unverified
                Task: BGP_0_65003.fe80::205:86ff:fe0c:dd03
                Announcement bits (3): 0-KRT 1-BGP_Multi_Path 2-BGP_RT_Background
                AS path: 65001 65004 I
                Accepted Multipath
                Localpref: 100
                Router ID: 10.0.0.1
                Thread: junos-main
        BGP     Preference: 170/-101
                Next hop type: Router, Next hop index: 1731
                Address: 0xd22cc50
                Next-hop reference count: 5
                Source: fe80::205:86ff:fec6:b503%xe-0/0/1.0
                Next hop: fe80::205:86ff:fec6:b503 via xe-0/0/1.0, selected
                Session Id: 321
                State: <Ext>
                Inactive reason: Active preferred
                Peer AS: 65002
                Age: 1:39:21
                Validation State: unverified
                Task: BGP_0_65003.fe80::205:86ff:fec6:b503
                AS path: 65002 65004 I
                Accepted MultipathContrib
                Localpref: 100
                Router ID: 10.0.0.2
                Thread: junos-mai
```

```
user@leaf1> show route forwarding-table destination 2001:db8:100::4
Routing table: default.inet6
Internet6:
Destination        Type RtRef Next hop           Type Index    NhRef Netif
2001:db8:100::4/128 user     0                    ulst   131070     2
                              fe80::205:86ff:fe0c:dd03
                                                   ucst     1730     6 xe-0/0/0.0
                              fe80::205:86ff:fec6:b503
                                                   ucst     1731     6 xe-0/0/1.0
```

Meaning

The output confirms that both paths between leaf 1 and leaf 2 are actively used to forward traffic and are part of a multipath route.

Verify Fabric Forwarding

IN THIS SECTION

Purpose

Confirm that leaf 1 and leaf 2 have connectivity over the underlay.

Action

```
user@leaf1> traceroute no-resolve 2001:db8:100::4
traceroute6 to 2001:db8:100::4 (2001:db8:100::4) from 2001:db8:100::3, 64 hops max, 12 byte packets
 1  2001:db8:100::1  220.185 ms  210.200 ms  203.652 ms
 2  2001:db8:100::4  213.774 ms  246.773 ms  186.533 ms
```

Meaning

The output confirms that leaf 1 and leaf 2 have underlay connectivity. The path shows that this flow went through spine 1, as indicated by its loopback address as the first hop.

Summary of BGP Unnumbered EVPN Fabric NCE

Congratulations! You configured and validated a BGP unnumbered fabric based on IPv6 and eBGP. This underlay is now ready to support your choice of EVPN-VXLAN overlay. Table 2 on page 26 details your accomplishments.

Table 2: Summary of Accomplishments

Configured BGP unnumbered peering for an EVPN-VXLAN fabric.	In this step, you configured the fabric devices to support IPv6 unnumbered peering. Only minimal interaction is needed to add the `inet6` family on the fabric interfaces and to define a BGP peer group that supports automatic peering. Junos OS support for a routing policy that defined the allowed range of automatic peering AS numbers, and configuration groups, greatly reduces the configuration size and complexity needed to deploy large fabrics.

Validated the operation of a BGP unnumbered underlay	You confirmed link-local addressing, IPv6 router advertisements and neighbor discovery. You verified dynamic IPv6 BGP peering to link-local addresses. You also confirmed the expected loopback route exchange needed to support an EVPN-VXLAN overlay.

Next, you'll want to add the EVPN-VXLAN overlay. Refer to Configure IBGP for the Overlay for details on adding an overlay.

RELATED DOCUMENTATION

- EVPN-VXLAN Technology and Architectures
- EVPN User Guide
- vLab Sandbox: IP Fabric with EVPN-VXLAN