Configuring VXLAN BGP EVPN

This chapter contains the following sections:

- About VXLAN BGP EVPN, on page 1
- Guidelines and Limitations for VXLAN BGP EVPN, on page 3
- About VXLAN EVPN with Downstream VNI, on page 8
- Guidelines and Limitations for VXLAN EVPN with Downstream VNI, on page 10
- Configuring VXLAN BGP EVPN, on page 12
- Configuring ND Suppression, on page 62

About VXLAN BGP EVPN

About RD Auto

The auto-derived Route Distinguisher (rd auto) is based on the Type 1 encoding format as described in IETF RFC 4364 section 4.2 https://tools.ietf.org/html/rfc4364#section-4.2. The Type 1 encoding allows a 4-byte administrative field and a 2-byte numbering field. Within Cisco NX-OS, the auto derived RD is constructed with the IP address of the BGP Router ID as the 4-byte administrative field (RID) and the internal VRF identifier for the 2-byte numbering field (VRF ID).

The 2-byte numbering field is always derived from the VRF, but results in a different numbering scheme depending on its use for the IP-VRF or the MAC-VRF:

- The 2-byte numbering field for the IP-VRF uses the internal VRF ID starting at 1 and increments. VRF IDs 1 and 2 are reserved for the default VRF and the management VRF respectively. The first custom defined IP VRF uses VRF ID 3.
- The 2-byte numbering field for the MAC-VRF uses the VLAN ID + 32767, which results in 32768 for VLAN ID 1 and incrementing.

Example auto-derived Route Distinguisher (RD)

- IP-VRF with BGP Router ID 192.0.2.1 and VRF ID 6 - RD 192.0.2.1:6
- MAC-VRF with BGP Router ID 192.0.2.1 and VLAN 20 - RD 192.0.2.1:32787

About Route-Target Auto

The auto-derived Route-Target (route-target import/export/both auto) is based on the Type 0 encoding format as described in IETF RFC 4364 section 4.2 (https://tools.ietf.org/html/rfc4364#section-4.2). IETF RFC 4364 section 4.2 describes the Route Distinguisher format and IETF RFC 4364 section 4.3.1 refers that it is desirable to use a similar format for the Route-Targets. The Type 0 encoding allows a 2-byte administrative field and a 4-byte numbering field. Within Cisco NX-OS, the auto derived Route-Target is constructed with the Autonomous System Number (ASN) as the 2-byte administrative field and the Service Identifier (VNI) for the 4-byte numbering field.

2-byte ASN

The Type 0 encoding allows a 2-byte administrative field and a 4-byte numbering field. Within Cisco NX-OS, the auto-derived Route-Target is constructed with the Autonomous System Number (ASN) as the 2-byte administrative filed and the Service Identifier (VNI) for the 4-byte numbering field.

Examples of an auto derived Route-Target (RT):

- IP-VRF within ASN 65001 and L3VNI 50001 - Route-Target 65001:50001
- MAC-VRF within ASN 65001 and L2VNI 30001 - Route-Target 65001:30001

For Multi-AS environments, the Route-Targets must either be statically defined or rewritten to match the ASN portion of the Route-Targets.

https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus9000/sw/7-x/command_references/configuration_commands/b_N9K_Config_Commands_703i7x/b_N9K_Config_Commands_703i7x_chapter_010010.html#wp4498893710

4-byte ASN

The Type 0 encoding allows a 2-byte administrative field and a 4-byte numbering field. Within Cisco NX-OS, the auto-derived Route-Target is constructed with the Autonomous System Number (ASN) as the 2-byte administrative filed and the Service Identifier (VNI) for the 4-byte numbering field. With the ASN demand of 4-byte length and the VNI requiring 24-bit (3-bytes), the Sub-Field length within the Extended Community is exhausted (2-byte Type and 6-byte Sub-Field). As a result of the length and format constraint and the importance of the Service Identifiers (VNI) uniqueness, the 4-byte ASN is represented in a 2-byte ASN named AS_TRANS, as described in IETF RFC 6793 section 9 (https://tools.ietf.org/html/rfc6793#section-9). The 2-byte ASN 23456 is registered by the IANA (https://www.iana.org/assignments/iana-as-numbers-special-registry/iana-as-numbers-special-registry.xhtml) as AS_TRANS, a special purpose AS number that aliases 4-byte ASNs.

Example auto derived Route-Target (RT) with 4-byte ASN (AS_TRANS):

- IP-VRF within ASN 65656 and L3VNI 50001 - Route-Target 23456:50001
- MAC-VRF within ASN 65656 and L2VNI 30001 - Route-Target 23456:30001

Note Beginning with Cisco NX-OS Release 9.2(1), auto derived Route-Target for 4-byte ASN is supported.

Guidelines and Limitations for VXLAN BGP EVPN

VXLAN BGP EVPN has the following guidelines and limitations:

- The following guidelines and limitations apply to VXLAN/VTEP using BGP EVPN:
  - SPAN source or destination is supported on any port.

  For more information, see the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide, Release 9.3(x).
- When SVI is enabled on a VTEP (flood and learn, or EVPN) regardless of ARP suppression, make sure that ARP-ETHER TCAM is carved using the **hardware access-list tcam region arp-ether 256 double-wide** command. This requirement does not apply to Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2/FX3 and 9300-GX platform switches and Cisco Nexus 9500 platform switches with 9700-EX/FX line cards.
- For the Cisco Nexus 9504 and 9508 with R-series line cards, VXLAN EVPN (Layer 2 and Layer 3) is only supported with the 9636C-RX and 96136YC-R line cards.
- VXLAN is not supported on N9K-C92348GC-X switches.
- You can configure EVPN over segment routing or MPLS. See the Cisco Nexus 9000 Series NX-OS Label Switching Configuration Guide, Release 9.3(x) for more information.
- You can use MPLS tunnel encapsulation using the new CLI encapsulation mpls command. You can configure the label allocation mode for the EVPN address family. See the Cisco Nexus 9000 Series NX-OS Label Switching Configuration Guide, Release 9.3(x) for more information.
- In a VXLAN EVPN setup that has 2K VNI scale configuration, the control plane down time may take more than 200 seconds. To avoid potential BGP flap, extend the graceful restart time to 300 seconds.
- The command "clear ip arp <interface> vrf <vrf-name> force-delete" on specific interface normally deletes entries from ARP belonging to that interface and will relearn on traffic. However, when ARP for same IP is resolved on all ECMP paths, force-deleting ARP entry belonging to one of the ECMP interface will result in automatic relearning of that entry unless that link is down.
- IP unnumbered in EVPN underlay supports ECMP. Multiple IP unnumbered links are connected back to back between same switches. ARP will be resolved on all connected interfaces, thus providing ECMP.
- Beginning with Cisco NX-OS Release 10.2(2)F, the following scale limits are enhanced — Layer 2 VNIs, Extended Layer 2 VNIs, Layer 3 VNIs, SVI with Distributed Anycast Gateway, IPv4 and IPv6 host routes in internet-peering mode and the ECMP paths. For the VXLAN scale limit information, see the Cisco Nexus 9000 Series NX-OS Verified Scalability Guide, Release 10.2(2)F.
- Beginning with Cisco NX-OS Release 10.2(1q)F, VXLAN EVPN is supported on Cisco Nexus N9KC9332D-GX2B platform switches.
- Beginning with Cisco NX-OS Release 10.2(3)F, VXLAN EVPN is supported on Cisco Nexus 9364D-GX2A, and 9348D-GX2A platform switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, VXLAN EVPN is supported on Cisco Nexus 9348GC-FX3, 9348GC-FX3PH, and 9332D-H2R switches.
- Beginning with Cisco NX-OS Release 10.4(2)F, VXLAN EVPN is supported on Cisco Nexus 93400LD-H1 switches.

- Beginning with Cisco NX-OS Release 10.4(3)F, VXLAN EVPN is supported on Cisco Nexus 9364C-H1 switches.
- Starting from Cisco NX-OS Release 9.3(5), new VXLAN uplink capabilities are introduced:
  - A physical interface in default VRF is supported as VXLAN uplink.
  - A parent interface in default VRF, carrying subinterfaces with VRF and dot1q tags, is supported as VXLAN uplink.
  - A subinterface in any VRF and/or with dot1q tag remains not supported as VXLAN uplink.
  - An SVI in any VRF remains not supported as VXLAN uplink.
  - In vPC with physical peer-link, a SVI can be leveraged as backup underlay, default VRF only between the vPC members (infra-VLAN, system nve infra-vlans).
  - On a vPC pair, shutting down NVE or NVE loopback on one of the vPC nodes is not a supported configuration. This means that traffic failover on one-side NVE shut or one-side loopback shut is not supported.
  - FEX host interfaces remain not supported as VXLAN uplink and cannot have VTEPs connected (BUD node).
- During the vPC Border Gateway boot up process the NVE source loopback interface undergoes the hold down timer twice instead of just once. This is a day-1 and expected behavior.
- The value of the delay timer on NVE interface must be configured to a value that is less than the multi-site delay-restore timer.
- You need to configure the VXLAN uplink with **ip unreachables** in order to enable Path maximum transmission unit (MTU) discovery (PMTUD) in a VXLAN set up. PMTUD prevents fragmentation in the path between two endpoints by dynamically determining the lowest MTU along the path from the packet's source to its destination.
- In a VXLAN EVPN setup, border nodes must be configured with unique route distinguishers, preferably using the **auto rd** command. Not using unique route distinguishers across all border nodes is not supported. The use of unique route distinguishers is strongly recommended for all VTEPs of a fabric.
- ARP suppression is only supported for a VNI if the VTEP hosts the First-Hop Gateway (Distributed Anycast Gateway) for this VNI. The VTEP and the SVI for this VLAN have to be properly configured for the distributed Anycast Gateway operation, for example, global Anycast Gateway MAC address configured and Anycast Gateway feature with the virtual IP address on the SVI.
- The ARP suppression setting must match across the entire fabric. For a specific VNID, all VTEPs must be either configured or not configured.
- Mobility Sequence number of a locally originated type-2 route (MAC/MAC-IP) can be mismatched between vPC peers, with one vTEP having a sequence number K while other vTEP in the same complex can have the same route with sequence number 0. This does not cause any functional impact and the traffic is not impacted even after the host moves.
- DHCP snooping (Dynamic Host Configuration Protocol snooping) is not supported on VXLAN VLANs.
- RACLs are not supported on VXLAN uplink interfaces. VACLs are not supported on VXLAN de-capsulated traffic in egress direction; this applies for the inner traffic coming from network (VXLAN) towards the access (Ethernet).

As a best practice, always use PACLs/VACLs for the access (Ethernet) to the network (VXLAN) direction. See the Cisco Nexus 9000 Series NX-OS Security Configuration Guide, Release 9.3(x) for other guidelines and limitations for the VXLAN ACL feature.

- The Cisco Nexus 9000 QoS buffer-boost feature is not applicable for VXLAN traffic.
- For VXLAN BGP EVPN fabrics with EBGP, the following recommendations are applicable:
  - It is recommended to use loopbacks for the EBGP EVPN peering sessions (overlay control-plane).
  - It is a best practice to use the physical interfaces for EBGP IPv4/IPv6 peering sessions (underlay).
- Bind the NVE source-interface to a dedicated loopback interface and do not share this loopback with any function or peerings of Layer-3 protocols. A best practice is to use a dedicated loopback address for the VXLAN VTEP function.
- You must bind NVE to a loopback address that is separate from other loopback addresses that are required by Layer 3 protocols. NVE and other Layer 3 protocols using the same loopback is not supported.
- The NVE source-interface loopback is required to be present in the default VRF.
- Only EBGP peering between a VTEP and external nodes (Edge Router, Core Router or VNF) is supported.
  - EBGP peering from the VTEP to the external node using a physical interface or subinterfaces is recommended and it is a best practice (external connectivity).
  - The EBGP peering from the VTEP to the external node can be in the default VRF or in a tenant VRF (external connectivity).
  - The EBGP peering from the VTEP to a external node over VXLAN must be in a tenant VRF and must use the update-source of a loopback interface (peering over VXLAN).
  - Using an SVI for EBGP peering on a from the VTEP to the External Node requires the VLAN to be local (not VXLAN extended).
- When configuring VXLAN BGP EVPN, only the "System Routing Mode: Default" is applicable for the following hardware platforms:
  - Cisco Nexus 9300 platform switches
  - Cisco Nexus 9300-EX platform switches
  - Cisco Nexus 9300-FX/FX2/FX3 platform switches
  - Cisco Nexus 9300-GX/GX2/H2R/H1 platform switches
  - Cisco Nexus 9500 platform switches with X9500 line cards
  - Cisco Nexus 9500 platform switches with X9700-EX and X9700-FX line cards
- Changing the “System Routing Mode” requires a reload of the switch.
- Cisco Nexus 9516 platform is not supported for VXLAN EVPN.
- VXLAN is supported on Cisco Nexus 9500 platform switches with the following line cards:
  - 9500-R
  - 9564PX

  - 9564TX
  - 9536PQ
  - 9700-EX
  - 9700-FX
- Cisco Nexus 9500 platform switches with 9700-EX or -FX line cards support 1G, 10G, 25G, 40G, 100G and 400G for VXLAN uplinks.
- Cisco Nexus 9200 and 9300-EX/FX/FX2/FX3 and -GX support 1G, 10G, 25G, 40G, 100G and 400G for VXLAN uplinks.
- Beginning with Cisco NX-OS Release 10.2(3)F, Cisco Nexus 9300-GX2 platform switches support 10G, 25G, 40G, 100G and 400G for VXLAN uplinks.
- Beginning with Cisco NX-OS Release 10.4(1)F, Cisco Nexus 9332D-H2R switches support 10G, 25G, 40G, 100G and 400G for VXLAN uplinks.
- Beginning with Cisco NX-OS Release 10.4(2)F, Cisco Nexus 93400LD-H1 switches support 10G, 25G, 40G, 100G and 400G for VXLAN uplinks.
- Beginning with Cisco NX-OS Release 10.4(3)F, Cisco Nexus 9364C-H1 switches support 10G, 25G, 40G, 100G and 400G for VXLAN uplinks.
- The Cisco Nexus 9000 platform switches use standards conforming UDP port number 4789 for VXLAN encapsulation. This value is not configurable.
- The Cisco Nexus 9200 platform switches with Application Spine Engine (ASE2) have throughput constrains for packet sizes of 99-122 bytes; packet drops might be experienced.
- The VXLAN network identifier (VNID) 16777215 is reserved and should explicitly not be configured.
- Non-Disruptive In Service Software Upgrade (ND-ISSU) is supported on Nexus 9300 with VXLAN enabled. Exception is ND-ISSU support for Cisco Nexus 9300-FX3 and 9300-GX platform switch.
- Gateway functionality for VXLAN to MPLS (LDP), VXLAN to MPLS-SR (Segment Routing) and VXLAN to SRv6 can be operated on the same Cisco Nexus 9000 Series platform.
  - VXLAN to MPLS (LDP) Gateway is supported on the Cisco Nexus 3600-R and the Cisco Nexus 9500 with R-Series line cards.
  - VXLAN to MPLS-SR Gateway is supported on the Cisco Nexus 9300-FX2/FX3/GX and Cisco Nexus 9500 with R-Series line cards.
  - Beginning with Cisco NX-OS Release 10.2(3)F, VXLAN to MPLS-SR Gateway is supported on the Cisco Nexus 9300-GX2 platform switches.
  - Beginning with Cisco NX-OS Release 10.4(1)F, VXLAN to MPLS-SR Gateway is supported on the Cisco Nexus 9332D-H2R switches.
  - Beginning with Cisco NX-OS Release 10.4(2)F, VXLAN to MPLS-SR Gateway is supported on the Cisco Nexus 93400LD-H1 switches.
  - Beginning with Cisco NX-OS Release 10.4(3)F, VXLAN to MPLS-SR Gateway is supported on the Cisco Nexus 9364C-H1 switches.
  - VXLAN to SRv6 is supported on the Cisco Nexus 9300-GX platform.

- Beginning with Cisco NX-OS Release 10.2(3)F, VXLAN to SRv6 is supported on the Cisco Nexus 9300-GX2 platform switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, VXLAN to SRv6 is supported on the Cisco Nexus 9332D-H2R switches.
- Beginning with Cisco NX-OS Release 10.4(2)F, VXLAN to SRv6 is supported on the Cisco Nexus 93400LD-H1 switches.
- Beginning with Cisco NX-OS Release 10.4(3)F, VXLAN to SRv6 is supported on the Cisco Nexus 9364C-H1 switches.
- Beginning with Cisco NX-OS Release 10.2(3)F, VXLAN and GRE co-existence is supported on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2 switches, and N9K-C93108TC-FX3P, N9K-C93180YC-FX3, N9K-X9716D-GX switches. Only GRE RX path (decapsulation) is supported. GRE TX path (encapsulation) is not supported.
- Beginning with Cisco NX-OS Release 10.4(1)F, VXLAN and GRE co-existence is supported on Cisco Nexus 9332D-H2R switches. Only GRE RX path (decapsulation) is supported. GRE TX path (encapsulation) is not supported.
- Beginning with Cisco NX-OS Release 10.4(2)F, VXLAN and GRE co-existence is supported on Cisco Nexus 93400LD-H1 switches. Only GRE RX path (decapsulation) is supported. GRE TX path (encapsulation) is not supported.
- Beginning with Cisco NX-OS Release 10.4(3)F, VXLAN and GRE co-existence is supported on Cisco Nexus 9364C-H1 switches. Only GRE RX path (decapsulation) is supported. GRE TX path (encapsulation) is not supported.
- Multiple Tunnel Encapsulations (VXLAN, GRE and/or MPLS, static label or segment routing) can not co-exist on the same Cisco Nexus 9000 Series switch with Network Forwarding Engine (NFE).

- Resilient hashing is supported on the following switch platform with a VXLAN VTEP configured:
  - Cisco Nexus 9300-EX/FX/FX2/FX3/GX support ECMP resilient hashing.
  - Cisco Nexus 9300 with ALE uplink ports does not support resilient hashing.

Note Resilient hashing is disabled by default.

- Beginning with Cisco NX-OS Release 10.2(3)F, the ECMP resilient hashing is supported on the Cisco Nexus 9300-GX2 platform switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, the ECMP resilient hashing is supported on the Cisco Nexus 9300-H2R platform switches.
- Beginning with Cisco NX-OS Release 10.4(2)F, the ECMP resilient hashing is supported on the Cisco Nexus 93400LD-H1 platform switches.
- Beginning with Cisco NX-OS Release 10.4(3)F, the ECMP resilient hashing is supported on the Cisco Nexus 9364C-H1 switches.
- It is recommended to use the **vpc orphan-ports suspend** command for single attached and/or routed devices on a Cisco Nexus 9000 platform switch acting as vPC VTEP.

- Beginning with Cisco NX-OS Release 10.3(2)F, Static MAC for BGP EVPN is supported on Cisco Nexus 9300-EX/FX/FXP/FX2/FX3/GX/GX2 series switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, Static MAC for BGP EVPN is supported on Cisco Nexus 9300-H2R series switches.
- Beginning with Cisco NX-OS Release 10.4(2)F, Static MAC for BGP EVPN is supported on Cisco Nexus 93400LD-H1 series switches.
- Beginning with Cisco NX-OS Release 10.4(3)F, Static MAC for BGP EVPN is supported on Cisco Nexus 9364C-H1 switches.
- The **mac address-table static mac-address vlan vlan-id {[drop | interface {type slot/port} | port-channel number]}** command is supported on BGP EVPN.
- Cisco Nexus supports Type-6 EVPN routes (for IPv4) based on earlier version of **draft-ietf-bess-evpn-igmp-mld-proxy** draft, where SMET flag field is set as optional.
- Routing protocol adjacencies using Anycast Gateway SVIs is not supported.

Note For information about VXLAN BGP EVPN scalability, see the Cisco Nexus 9000 Series NX-OS Verified Scalability Guide.

About VXLAN EVPN with Downstream VNI

Cisco NX-OS Release 9.3(5) introduces VXLAN EVPN with downstream VNI. In earlier releases, the VNI configuration must be consistent across all nodes in the VXLAN EVPN network in order to enable communication between them.

VXLAN EVPN with downstream VNI provides the following solutions:

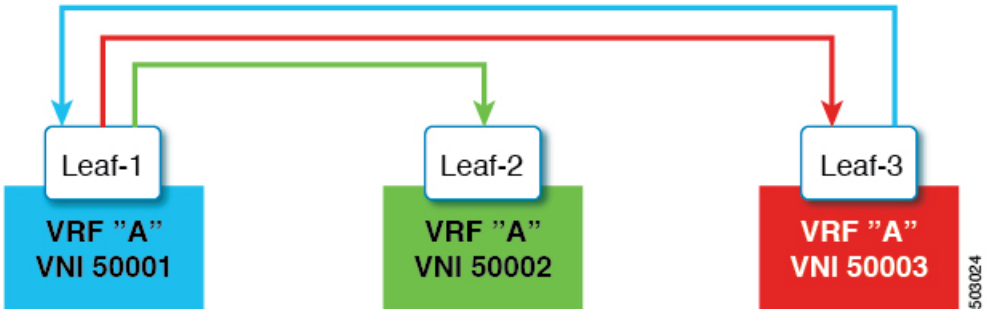
- Enables asymmetric VNI communication across nodes in a VXLAN EVPN network
- Provides customers access to a common shared service outside of their domain (tenant VRF)
- Supports communication between isolated VXLAN EVPN sites that have different sets of VNIs

Asymmetric VNIs

VXLAN EVPN with downstream VNI supports asymmetric VNI allocation.

The following figure shows an example of asymmetric VNIs. All three VTEPs have different VNIs configured for the same IP VRF or MAC VRF.

Figure 1: Asymmetric VNIs



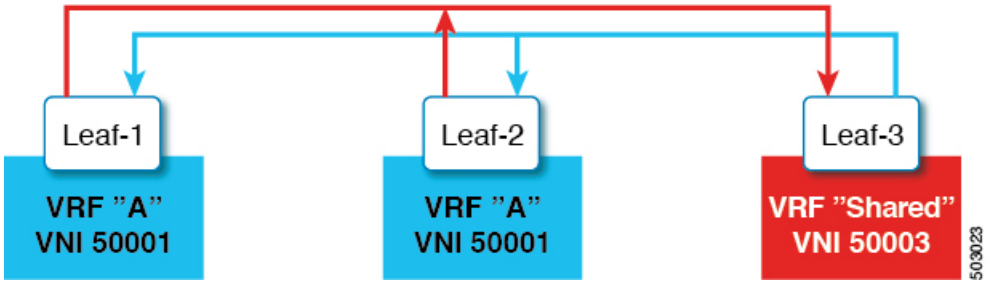
Shared Services VRFs

VXLAN EVPN with downstream VNI supports shared services VRFs. It does so by importing multiple L3VRFs into a single local L3VRF and supporting disparate values of downstream L3VNIs on a per-peer basis.

For example, a DNS server needs to serve multiple hosts in a data center regardless of the tenant VRFs on which the hosts sit. The DNS server is attached to a shared services VRF, which is attached to an L3VNI. To access this server from any of the tenant VRFs, the switches must import the routes from the shared services VRF to the tenant VRF, even though the L3VNI associated to the shared services VRF is different from the L3VNI associated to the tenant VRF.

In the following figure, Tenant VRF A in Leaf-1 can communicate with Tenant VRF A in Leaf-2. However, Tenant VRF A requires access to a shared service sitting behind Leaf-3.

Figure 2: Shared Services VRFs

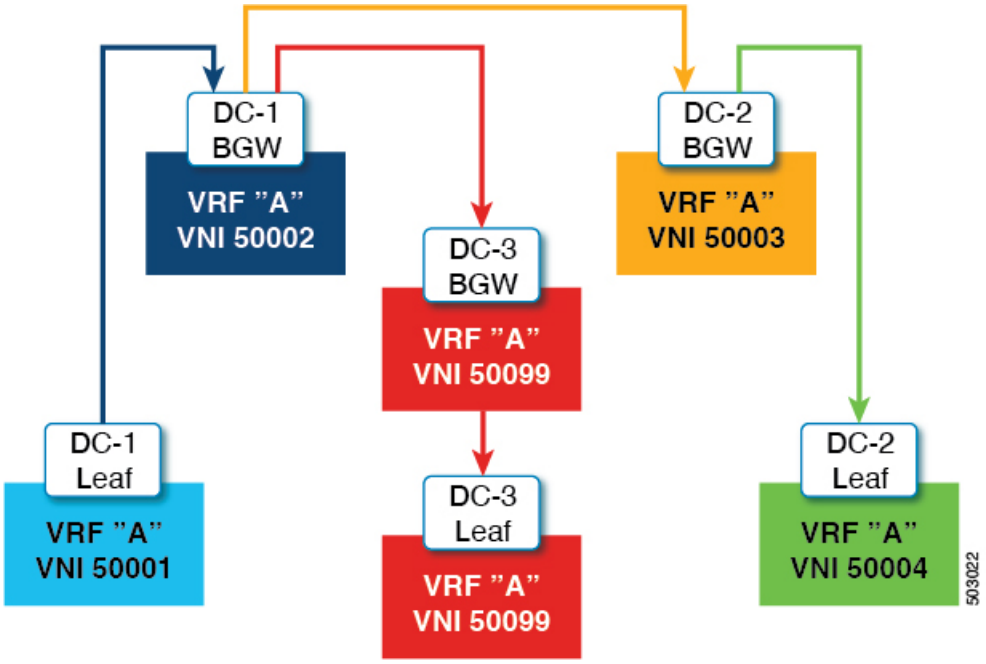


Multi-Site with Asymmetric VNIs

VXLAN EVPN with downstream VNI allows communication between sites that have different sets of VNIs. It does so by stitching the asymmetric VNIs at the border gateways.

In the following figure, DC-1 and DC-2 are asymmetric sites, and DC-3 is a symmetric site. Each site uses different VNIs within its site to communicate.

Figure 3: Multi-Site with Asymmetric VNIs



Guidelines and Limitations for VXLAN EVPN with Downstream VNI

VXLAN EVPN with downstream VNI has the following guidelines and limitations:

- Cisco Nexus 9332C, 9364C, 9300-EX, and 9300-FX/FX2/FXP platform switches and Cisco Nexus 9500 platform switches with -EX/FX line cards support VXLAN EVPN with downstream VNI.
- Beginning with Cisco NX-OS Release 9.3(7), Cisco Nexus 9300-GX platform switches support VXLAN EVPN with downstream VNI.
- Beginning with Cisco NX-OS Release 10.2(3)F, the VXLAN EVPN with downstream VNI is supported on the Cisco Nexus 9300-FX3/GX2 platform switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, the VXLAN EVPN with downstream VNI is supported on the Cisco Nexus 9332D-H2R switches.
- Beginning with Cisco NX-OS Release 10.4(2)F, the VXLAN EVPN with downstream VNI is supported on the Cisco Nexus 93400LD-H1 switches.
- Beginning with Cisco NX-OS Release 10.4(3)F, the VXLAN EVPN with downstream VNI is supported on the Cisco Nexus 9364C-H1 switches.
- VXLAN EVPN with downstream VNI is supported only on the IPv4 underlay.
- Downstream VNI is configured based on route-target export and import. The following conditions must be met to leverage Downstream VNI:

  - Downstream VNI requires the usage of different VRF (MAC-VRF or IP-VRF), each VRF must have a different VNI (Asymmetric VNI).
  - To import routes of a foreign VRF (MAC-VRF or IP-VRF) the appropriate route-target for the import into the local VRF must be configured.
  - The configuration of only auto-derived route-targets will not result in downstream VNI.
  - The export of VRF prefixes can be done by static or auto-derived route-target configuration.
  - The import of a foreign VRF's auto-derived route-target is supported.
  - The import of a foreign VRFs statically configured route-target is supported.

- Downstream VNI is supported for the following underlay constellations:
  - For downstream VNI with Layer-3 VNI, the underlay can be ingress replication or multicast based.
  - For downstream VNI with Layer-2 VNI, the underlay must be in ingress replication. Multicast based underlay is not supported with downstream VNI of Layer-2 VNIs.

- Downstream VNI requires to have consistent configuration:
  - All multi-site Border Gateway (BGW) in a site must have a consistent configuration.
  - All vPC members in a vPC domain must have consistent configuration.

- The usage of downstream VNI with multi-site requires all BGW across all sites to run at least Cisco NX-OS Release 9.3(5).
- For existing centralized VRF route leaking deployments, a brief traffic loss might occur during ISSU to Cisco NX-OS Release 9.3(5) or later.
- For successful downgrade from Cisco NX-OS Release 9.3(5) to a prior release, ensure that the asymmetric VNI configuration has been removed. Downstream VNI is not supported before Cisco NX-OS Release 9.3(5) and hence traffic forwarding would be impacted.
- Layer-3 VNIs (IP-VRF) can flexibly mapped between VNIs per peer.
  - VNI 50001 on VTEP1 can perform symmetric VNI with VNI 50001 and asymmetric VNI with VNI 50002 on VTEP2 at the same time.
  - VNI 50001 on VTEP1 can perform asymmetric VNI with VNI 50002 on VTEP2 and VNI 50003 on VTEP3.
  - VNI 50001 on VTEP1 can perform asymmetric VNI with VNI 50002 and VNI5003 on VTEP2 at the same time.

- Layer-2 VNIs (MAC-VRF) can only be mapped to one VNI per peer.
  - VNI 30001 on VTEP1 can perform asymmetric VNI with VNI 30002 on VTEP2 and VNI 30003 on VTEP3.
  - VNI 30001 on VTEP1 cannot perform asymmetric VNI with VNI 30002 and VNI 3003 on VTEP2 at the same time.

- iBGP sessions between vPC peer nodes in a VRF are not supported.

- BGP peering across VXLAN and Downstream VNI support the following constellations:
  - BGP peering between symmetric VNI is supported by using loopbacks.
  - BGP peering between asymmetric VNI is supported if the VNIs are in a direct message relationship. A loopback from VNI 50001 (on VTEP1) can peer with a loopback in VNI 50002 (on VTEP2).
  - BGP peering between asymmetric VNI is supported if the VNIs are in a direct message relationship but on different VTEPs. A loopback from VNI 50001 (on VTEP1) can peer with a loopback in VNI 50002 (on VTEP2 and VTEP3).
  - BGP peering between asymmetric VNI is not supported if the VNIs are in a 1:N relationship. A loopback in VNI 50001 (VTEP1) can't peer with a loopback in VNI 50002 (VTEP2) and VNI 50003 (VTEP3) at the same time.
- VXLAN consistency checker is not supported for VXLAN EVPN with downstream VNI.
- VXLAN EVPN with downstream VNI is currently not supported with the following feature combinations:
  - VXLAN static tunnels
  - TRM and TRM with Multi-Site
  - CloudSec VXLAN EVPN Tunnel Encryption
  - ESI-based multihoming
  - Seamless integration of EVPN with L3VPN (MPLS SR)
  - VXLAN policy-based routing (PBR)
- Make sure that you configure L2VNI SVI on Anycast BGW to enable DSVNI MAC-IP Layer 3 label translation in a multisite environment. The functionality of DSVNI is limited for reoriginated routes, which requires as association between L2VNI and VRF. You can associate using the VRF member command in L2VNI SVI.

Configuring VXLAN BGP EVPN

Enabling VXLAN

Enable VXLAN and the EVPN.

SUMMARY STEPS

1. **feature vn-segment**
2. **feature nv overlay**
3. **feature vn-segment-vlan-based**
4. **feature interface-vlan**
5. **nv overlay evpn**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**feature vn-segment**	Enable VLAN-based VXLAN
Step 2	**feature nv overlay**	Enable VXLAN
Step 3	**feature vn-segment-vlan-based**	Enable VN-Segment for VLANs.
Step 4	**feature interface-vlan**	Enable Switch Virtual Interface (SVI).
Step 5	**nv overlay evpn**	Enable the EVPN control plane for VXLAN.

Configuring VLAN and VXLAN VNI

Note Step 3 to Step 6 are optional for configuring the VLAN for VXLAN VNI and are only necessary in case of a custom route distinguisher or route-target requirement (not using auto derivation).

SUMMARY STEPS

1. **vlan** *number*
2. **vn-segment** *number*
3. **evpn**
4. **vni** *number* **l2**
5. **rd auto**
6. **route-target both** {**auto** | *rt*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**vlan** *number*	Specify VLAN.
Step 2	**vn-segment** *number*	Map VLAN to VXLAN VNI to configure Layer 2 VNI under VXLAN VLAN.
Step 3	**evpn**	Enter EVI (EVPN Virtual Instance) configuration mode.
Step 4	**vni** *number* **l2**	Specify the Service Instance (VNI) for the EVI.
Step 5	**rd auto**	Specify the MAC-VRF's route distinguisher (RD).
Step 6	**route-target both** {**auto** | *rt*}	Configure the route target (RT) for import and export of MAC prefixes. The RT is used for a per-MAC-VRF prefix import/export policy. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. **Note** Specifying the auto option is applicable only for IBGP.

	Command or Action	Purpose
		Manually configured route targets are required for EBGP and for asymmetric VNIs.

Configuring New L3VNI Mode

Guidelines and Limitations for New L3VNI Mode

New L3VNI mode has the following configuration guidelines and limitations:

- Beginning with Cisco NX-OS Release 10.2(3)F, the new L3VNI mode is supported on Cisco Nexus 9300-X Cloud Scale Switches.
- **interface vni** config is optional (not needed if the PBR/NAT feature is not required).
- VRF-VNI-L3 new configuration will implicitly create the L3VNI interface. By default, it will not show up in the show running command.

Note Ensure that VRF-VNI-L3 is configured before configuring **interface vni**.

- Following configuration are allowed on **interface vni**:
  - PBR/NAT
  - no interface vni
  - default interface vni (will remove PBR/NAT configuration if present)
- The **shut/no shut** command is not allowed on **interface vni**. Performing **shut/no shut** command on VRF performs shut/no shut on L3VNI.
- Performing **no feature nv overlay** with the new L3VNI configuration removes all vrf-vni-l3 config under VRF and cleanup the PBR/NAT configuration, if present. Any existing VRF configuration will not be removed.
- VNI Configuration has the following guidelines and limitations:
  - Both old and new L3VNI mode configuration can coexist on the same switch.
  - For the VPC/VMCT system, same VNI config mode should be consistent across peers.
  - Post upgrade, the old L3VNI configuration holds good.
  - Beginning with Cisco NX-OS Release 10.3(1)F, TRM support for the new L3VNI is provided on Cisco Nexus 9300-X Cloud Scale Switches.
  - Config-replace and rollback are supported.
  - ISSU (ND) is supported for the new L3VNI.
- PBR/NAT configuration on the new L3VNI has the following guidelines and limitations:
  - NAT configuration can be applied on the new **interface vni**.

- PBR encap side policy is still configured on encap node interface SVI as existing.
- PBR decap side policy for the new L3VNI now applies on **interface vni** for the corresponding L3VNI.
- PBR config syntax on the new L3VNI is similar to SVI interface.
- The **no interface vni** removes the PBR/NAT config first and then remove the **interface vni**.
- The **no interface vni** will only remove the CLI from config, as long as VRF-VNI-L3 config is still present, the **interface vni** is still present at the back-end.

- The following features are supported on the new L3VNI mode:
  - Leaf/VTEP features which use L3VNIs
    - VxLAN EVPN
      - IR and multicast.
      - IGMP Snooping
      - vPC
      - Distributed Anycast Gateway
    - MCT-less vPC
    - VxLAN Multisite
      - Cover all existing scenarios with Border Leaf, Border Spine and multi-site Border Gateway
      - Anycast BGW and vPC BGW
    - DSVNI
    - VxLAN NGOAM
  - VXLAN supported features: PBR, NAT, and QoS
  - VXLAN access features (QinVNI, SQinVNI, NIA, BUD-Node etc.)
  - 4K scale L2VNI for VXLAN Port VLAN-Mapping VXLAN feature.

- Migration of L3VNI configuration has the following guidelines and limitations:
  - To migrate the L3VNI configuration from old to new, perform the following steps:
    1. Remove the VLAN, vlan-vnsegment and SVI configuration..
    2. Retain Interface nve1 member-vni-associate configuration.
    3. Add new VRF-VNI-L3 configuration. For more information, refer to Configuring New L3VNI Mode, on page 16.
  - To migrate the L3VNI configuration from new to old, perform the following steps:
    1. Remove new VRF-VNI-L3 configuration.
    2. Create VLAN and vlan-vnsegment configuration.

3. Retain Interface nve1 member-vni-associate configuration.
4. Create SVI configuration for the L3VNI.
5. Add member-vni under VRF configuration.

- Upgrade and download have the following guidelines and limitations:
  - Upgrade:
    - Existing L3VNI configuration remains as is and stay functional.
    - You can configure additional L3VNIs with the new keyword **L3** without VLAN association.
    - You can choose to migrate the existing L3VNI config one by one to the new L3VNI without VLAN association.
    - If needed, you can revert from new L3VNI config to old L3VNI config (with VLAN association).
    - ND ISSU is supported for new L3VNI future releases.
  - Downgrade:
    - If the new L3 VNI is configured, check and disable the new L3VNI configuration before performing downgrade.
    - Downgrade will be allowed only after removing all new L3VNI configuration.

Configuring New L3VNI Mode

This procedure enables the new L3VNI mode on the switch:

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **vni** *number* **l3**
4. **member vni** *vni id* **associate-vrf**
5. (Optional) **{ip | ipv6} policy route-map** *map-name*
6. (Optional) **ip nat outside**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose
Step 2	**vrf context** *vrf-name* **Example:** `switch(config)# vrf context vxlan-501`	Configures the VRF.
Step 3	**vni** *number* **l3** **Example:** `switch(config)# vni 500001  L3`	Specifies the VNI. **L3** is the new keyword which indicates the new L3VNI mode.

Step 4	**member vni** *vni id* **associate-vrf**
	Example:
	`switch(config)# interface nve1`
	`switch(config-intf)# no shutdown`
	`switch(config-intf)# member vni 500001 associate-vrf`
		Associates L3VNI to VRF.

Step 5	(Optional) **{ip | ipv6} policy route-map** *map-name*
	Example:
	`switch(config)# interface vni 500001`
	Example:
	For IPv4
	`switch(config-intf)# ip policy route-map IPV4_PBR_Appgroup`
	Example:
	For IPv6
	`switch(config-intf)# ipv6 policy route-map IPV6_PBR_Appgroup`
		Assigns a route map for IPv4 or IPv6 policy-based routing to L3VNI interface.

Step 6	(Optional) **ip nat outside**
	Example:
	`switch(config)# interface vni 500001`
	`switch(config-intf)# ip nat outside`
		Assigns a route map for NAT to L3VNI interface.

Verifying New L3VNI Mode Configuration

To display the new L3VNI mode configuration information, perform the following task:

Command	Purpose
Show nve vni	Displays corresponding new l3vni state

Configuring VRF for VXLAN Routing

Configure the tenant VRF.

Note Step 3 to step 6 are optional for configuring the VRF for VXLAN Routing and are only necessary in case of a custom route distinguisher or route-target requirement (not using auto derivation).

SUMMARY STEPS

1. **vrf context** *vrf-name*
2. **vni** *number*
3. **rd auto**
4. **address-family** {**ipv4** | **ipv6**} **unicast**
5. **route-target both** {**auto** | *rt*}
6. **route-target both** {**auto** | *rt*} **evpn**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**vrf context** *vrf-name*	Configure the VRF.
Step 2	**vni** *number*	Specify the VNI.
Step 3	**rd auto**	Specify the IP-VRF's route distinguisher (RD).
Step 4	**address-family** {**ipv4**	**ipv6**} **unicast**	Configure the IPv4 or IPv6 unicast address family.
Step 5	**route-target both** {**auto**	*rt*}	Configure the route target (RT) for import and export of IPv4 or IPv6 prefixes. The RT is used for a per-IP-VRF prefix import/export policy. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. **Note** Specifying the **auto** option is applicable only for IBGP. Manually configured route targets are required for EBGP and for asymmetric VNIs.
Step 6	**route-target both** {**auto**	*rt*} **evpn**	Configure the route target (RT) for import and export of IPv4 or IPv6 prefixes. The RT is used for a per-VRF prefix import/export policy. If you enter an RT, the following formats are supported: ASN2:NN, ASN4:NN, or IPV4:NN. **Note** Specifying the **auto** option is applicable only for IBGP. Manually configured route targets are required for EBGP and for asymmetric VNIs.

Configuring VXLAN UDP Source Port

Configure the VXLAN UDP source port.

SUMMARY STEPS

1. [**no**] **vxlan udp src-port** [**high** | **rfc** | **low**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	[**no**] **vxlan udp src-port** [**high** | **rfc** | **low**]	Allows to select the VXLAN UDP source port number range for VXLAN encapsulated packets. **high**: This option sets the port number range to 0x8000-0xFFFF. **rfc**: Beginning with Cisco NX-OS Release 10.4(1)F, the **rfc** option is provided to set the port number range to 0xC000-0xFFFF. **Note** The **rfc** option is available only on Cisco Nexus 9332D-H2R, 9364C-H1, and 93400LD-H1 switches. **low**: Beginning with Cisco NX-OS Release 10.4(1)F, the **low** option is provided to set the port number range to default value (1024 to 32K-1). This is the default option. The **no** form of the **high** and **rfc** command is equivalent to the **low** command. **Note** The **low** option is available on all Cisco Nexus 9000 Series platform switches.

Configuring SVI for Core-facing VXLAN Routing

Configure the core-facing SVI VRF.

SUMMARY STEPS

1. **vlan** *number*
2. **vn-segment** *number*
3. **interface** *vlan-number*
4. **mtu** *vlan-number*
5. **vrf member** *vrf-name*
6. **no** {**ip** |**ipv6**} **redirects**
7. **ip forward**
8. **ipv6 address use-link-local-only**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**vlan** *number*	Specify VLAN.
Step 2	**vn-segment** *number*	Map VLAN to VXLAN VNI to configure Layer 3 VNI under VXLAN VLAN.
Step 3	**interface** *vlan-number*	Specify VLAN interface.
Step 4	**mtu** *vlan-number*	MTU size in bytes <68-9216>.
Step 5	**vrf member** *vrf-name*	Assign to VRF.
Step 6	**no** {**ip** |**ipv6**} **redirects**	Disable sending IP redirect messages for IPv4 and IPv6.
Step 7	**ip forward**	Enable IPv4 based lookup even when the interface VLAN has no IP address defined.
Step 8	**ipv6 address use-link-local-only**	Enable IPv6 forwarding. **Note** The IPv6 address use-link-local-only serves the same purpose as ip forward for IPv4. It enables the switch to perform an IP based lookup even when the interface VLAN has no IP address defined under it.

Configuring SVI for Host-Facing VXLAN Routing

Configure the SVI for hosts, acting as Distributed Default Gateway.

SUMMARY STEPS

1. **fabric forwarding anycast-gateway-mac** *address*
2. **vlan** *number*
3. **vn-segment** *number*
4. **interface** *vlan-number*
5. **vrf member** *vrf-name*
6. **ip address** *address*
7. **fabric forwarding mode anycast-gateway**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**fabric forwarding anycast-gateway-mac** *address*	Configure distributed gateway virtual MAC address. **Note** One virtual MAC per VTEP. **Note** All VTEPs should have the same virtual MAC address.

	Command or Action	Purpose
Step 2	**vlan** *number*	Specify VLAN.
Step 3	**vn-segment** *number*	Specify vn-segment.
Step 4	**interface** *vlan-number*	Specify VLAN interface.
Step 5	**vrf member** *vrf-name*	Assign to VRF.
Step 6	**ip address** *address*	Specify IP address.
Step 7	**fabric forwarding mode anycast-gateway**	Associate SVI with anycast gateway under VLAN configuration mode.

Configuring the NVE Interface and VNIs Using Multicast

SUMMARY STEPS

1. **interface** *nve-interface*
2. **source-interface loopback1**
3. **host-reachability protocol bgp**
4. **global mcast-group** *ip-address* **{L2 | L3}**
5. **member vni** *vni*
6. **mcast-group** *ip address*
7. **member vni** *vni* **associate-vrf**
8. **mcast-group** *address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**interface** *nve-interface*	Configure the NVE interface.
Step 2	**source-interface loopback1**	Binds the NVE source-interface to a dedicated loopback interface.
Step 3	**host-reachability protocol bgp**	This defines BGP as the mechanism for host reachability advertisement
Step 4	**global mcast-group** *ip-address* **{L2 | L3}**	Configures the mcast group globally (for all VNI) on a per-NVE interface basis. This applies and gets inherited s to all Layer 2 or Layer 3 VNIs. **Note** Layer3 macst group is only used for Tenant Routed Multicast (TRM).
Step 5	**member vni** *vni*	Add Layer 2 VNIs to the tunnel interface.

	Command or Action	Purpose
Step 6	**mcast-group** *ip address*	Configure the mcast group on a per-VNI basis. Add Layer 2 VNI specific mcast group and override the global set configuration. **Note** Instead of a mcast group, ingress replication can be configured.
Step 7	**member vni** *vni* **associate-vrf**	Add Layer-3 VNIs, one per tenant VRF, to the overlay. **Note** Required for VXLAN routing only.
Step 8	**mcast-group** *address*	Configure the mcast group on a per-VNI basis. Add Layer 3 VNI specific mcast group and override the global set configuration.

Configuring the Delay Timer on NVE Interface

Configuring the delay timer on NVE interface allows BGP to delay the fabric route advertisement to VRF peers and VRF peer routes to fabric so that there are no transient traffic drops seen when border leaf nodes come up after a switch reload. Configure this timer on NX-OS border leaf and AnyCast border gateway.

The value of the delay timer on NVE interface depends on the scale values of NVE peers, VNIs, routes, and so on. To find the timer value to be configured, find the time it took to program the last NVE peer after reload and add buffer time of 100 seconds to it. This buffer time also provides time for route-advertisement. Use the **show forwarding internal trace nve-peer-history** command to display the time stamp of each NVE peer installed.

Also, convergence will not be improved for fabric isolation on NX-OS border leaf even when this timer is configured.

SUMMARY STEPS

1. **configure terminal**
2. **interface nve** *nve-interface*
3. **fabric-ready time** *seconds*
4. **show nve interface nve1 detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface nve** *nve-interface*	Configures the NVE interface.
Step 3	**fabric-ready time** *seconds*	Specifies the delay timer value for NVE interface. The default value is 135 seconds.
Step 4	**show nve interface nve1 detail**	Displays the configured timer value.

Configuring VXLAN EVPN Ingress Replication

For VXLAN EVPN ingress replication, the VXLAN VTEP uses a list of IP addresses of other VTEPs in the network to send BUM (broadcast, unknown unicast and multicast) traffic. These IP addresses are exchanged between VTEPs through the BGP EVPN control plane.

Note VXLAN EVPN ingress replication is supported on:

- Cisco Nexus Series 9300 Series switches (7.0(3)I1(2) and later).
- Cisco Nexus Series 9500 Series switches (7.0(3)I2(1) and later).

Before you begin: The following are required before configuring VXLAN EVPN ingress replication (7.0(3)I1(2) and later):

- Enable VXLAN.
- Configure VLAN and VXLAN VNI.
- Configure BGP on the VTEP.
- Configure RD and Route Targets for VXLAN Bridging.

SUMMARY STEPS

1. **interface** *nve-interface*
2. **host-reachability protocol bgp**
3. **global ingress-replication protocol bgp**
4. **member vni** *vni* **associate-vrf**
5. **member vni** *vni*
6. **ingress-replication protocol bgp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**interface** *nve-interface*	Configure the NVE interface.
Step 2	**host-reachability protocol bgp**	This defines BGP as the mechanism for host reachability advertisement.
Step 3	**global ingress-replication protocol bgp**	Enables globally (for all VNI) the VTEP to exchange local and remote VTEP IP addresses on the VNI in order to create the ingress replication list. This enables sending and receiving BUM traffic for the VNI. **Note** Using ingress-replication protocol bgp avoids the need for any multicast configurations that might have been required for configuring the underlay.

	Command or Action	Purpose
Step 4	**member vni** *vni* **associate-vrf**	Add Layer-3 VNIs, one per tenant VRF, to the overlay. **Note** Required for VXLAN routing only.
Step 5	**member vni** *vni*	Add Layer 2 VNIs to the tunnel interface.
Step 6	**ingress-replication protocol bgp**	Enables the VTEP to exchange local and remote VTEP IP addresses on a oer VNI basis in order to create the ingress replication list. This enables sending and receiving BUM traffic for the VNI and override the global configuration. **Note** Instead of a ingress replication, mcast group can be configured. **Note** Using **ingress-replication protocol bgp** avoids the need for any multicast configurations that might have been required for configuring the underlay.

Configuring BGP on the VTEP

SUMMARY STEPS

1. **router bgp** *number*
2. **router-id** *address*
3. **neighbor** *address* **remote-as** *number*
4. **address-family l2vpn evpn**
5. (Optional) **Allowas-in**
6. **send-community extended**
7. **vrf** *vrf-name*
8. **address-family ipv4 unicast**
9. **advertise** *l2vpn* **evpn**
10. **maximum-paths path** {**ibgp**}
11. **address-family ipv6 unicast**
12. **advertise** *l2vpn* **evpn**
13. **maximum-paths path** {**ibgp**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**router bgp** *number*	Configure BGP.
Step 2	**router-id** *address*	Specify router address.
Step 3	**neighbor** *address* **remote-as** *number*	Define MPBGP neighbors. Under each neighbor define L2VPN EVPN.

	Command or Action	Purpose
Step 4	**address-family l2vpn evpn**	Configure address family Layer 2 VPN EVPN under the BGP neighbor. **Note** Address-family IPv4 EVPN for VXLAN host-based routing
Step 5	(Optional) **Allowas-in**	Only for EBGP deployment cases: Allows duplicate autonomous system (AS) numbers in the AS path. Configure this parameter on the leaf for eBGP when all leafs are using the same AS, but the spines have a different AS than leafs.
Step 6	**send-community extended**	Configures community for BGP neighbors.
Step 7	**vrf** *vrf-name*	Specify VRF.
Step 8	**address-family ipv4 unicast**	Configure the address family for IPv4.
Step 9	**advertise** *l2vpn* **evpn**	Enable advertising EVPN routes. **Note** Beginning with Cisco NX-OS Release 9.2(1), the **advertise l2vpn evpn** command no longer takes effect. To disable advertisement for a VRF toward the EVPN, disable the VNI in NVE by entering the **no member vni** *vni* **associate-vrf** command in interface nve1. The *vni* is the VNI associated with that particular VRF.
Step 10	**maximum-paths path** {**ibgp**}	Enable ECMP for EVPN transported IP Prefixes within the IPv6 address-family of the respective VRF.
Step 11	**address-family ipv6 unicast**	Configure the address family for IPv6.
Step 12	**advertise** *l2vpn* **evpn**	Enable advertising EVPN routes. **Note** To disable advertisement for a VRF toward the EVPN, disable the VNI in NVE by entering the **no member vni** *vni* **associate-vrf** command in interface nve1. The *vni* is the VNI associated with that particular VRF.
Step 13	**maximum-paths path** {**ibgp**}	Enable ECMP for EVPN transported IP Prefixes within the IPv6 address-family of the respective VRF.

Configuring iBGP for EVPN on the Spine

SUMMARY STEPS

1. **router bgp** *autonomous system number*
2. **neighbor** *address* **remote-as** *number*
3. **address-family l2vpn evpn**
4. **send-community extended**
5. **route-reflector-client**
6. **retain route-target all**
7. **address-family l2vpn evpn**
8. **disable-peer-as-check**
9. **route-map permitall out**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**router bgp** *autonomous system number*	Specify BGP.
Step 2	**neighbor** *address* **remote-as** *number*	Define neighbor.
Step 3	**address-family l2vpn evpn**	Configure address family Layer 2 VPN EVPN under the BGP neighbor.
Step 4	**send-community extended**	Configures community for BGP neighbors.
Step 5	**route-reflector-client**	Enable Spine as Route Reflector.
Step 6	**retain route-target all**	Configure retain route-target all under address-family Layer 2 VPN EVPN [global]. **Note** Required for eBGP. Allows the spine to retain and advertise all EVPN routes when there are no local VNI configured with matching import route targets.
Step 7	**address-family l2vpn evpn**	Configure address family Layer 2 VPN EVPN under the BGP neighbor.
Step 8	**disable-peer-as-check**	Disables checking the peer AS number during route advertisement. Configure this parameter on the spine for eBGP when all leafs are using the same AS but the spines have a different AS than leafs. **Note** Required for eBGP.
Step 9	**route-map permitall out**	Applies route-map to keep the next-hop unchanged. **Note** Required for eBGP.

Configuring eBGP for EVPN on the Spine

SUMMARY STEPS

1. **route-map NEXT-HOP-UNCH permit 10**
2. **set ip next-hop unchanged**
3. **router bgp** *autonomous system number*
4. **address-family l2vpn evpn**
5. **retain route-target all**
6. **neighbor** *address* **remote-as** *number*
7. **address-family l2vpn evpn**
8. **disable-peer-as-check**
9. **send-community extended**
10. **route-map NEXT-HOP-UNCH out**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**route-map NEXT-HOP-UNCH permit 10**	Configure route-map to keepthe next-hop unchanged for EVPN routes.
Step 2	**set ip next-hop unchanged**	Set next-hop address. **Note** When two next hops are enabled, next hop ordering is not maintained. If one of the next hops is a VXLAN next hop and the other next hop is local reachable via FIB/AM/Hmm, the local next hop reachable via FIB/AM/Hmm is always taken irrespective of the order. Directly/locally connected next hops are always given priority over remotely connected next hops.
Step 3	**router bgp** *autonomous system number*	Specify BGP.
Step 4	**address-family l2vpn evpn**	Configure address family Layer 2 VPN EVPN under the BGP neighbor.
Step 5	**retain route-target all**	Configure retain route-target all under address-family Layer 2 VPN EVPN [global]. **Note** Required for eBGP. Allows the spine to retain and advertise all EVPN routes when there are no local VNI configured with matching import route targets.
Step 6	**neighbor** *address* **remote-as** *number*	Define neighbor.

	Command or Action	Purpose
Step 7	**address-family l2vpn evpn**	Configure address family Layer 2 VPN EVPN under the BGP neighbor.
Step 8	**disable-peer-as-check**	Disables checking the peer AS number during route advertisement. Configure this parameter on the spine for eBGP when all leafs are using the same AS but the spines have a different AS than leafs.
Step 9	**send-community extended**	Configures community for BGP neighbors.
Step 10	**route-map NEXT-HOP-UNCH out**	Applies route-map to keep the next-hop unchanged.

Suppressing ARP

Suppressing ARP includes changing the size of the ACL ternary content addressable memory (TCAM) regions in the hardware.

Note For information on configuring ACL TCAM regions, see the *Configuring IP ACLs* chapter of the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.

SUMMARY STEPS

1. **hardware access-list tcam region arp-ether** *size* **double-wide**
2. **interface nve 1**
3. **global suppress-arp**
4. **member vni** *vni-id*
5. **suppress-arp**
6. **suppress-arp disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**hardware access-list tcam region arp-ether** *size* **double-wide**	Configure TCAM region to suppress ARP. *tcam-size* —TCAM size. The size has to be a multiple of 256. If the size is more than 256, it has to be a multiple of 512. **Note** Reload is required for the TCAM configuration to be in effect. **Note** Configuring the **hardware access-list tcam region arp-ether** *size* **double-wide** command is not required for Cisco Nexus 9200, 9300-EX, and 9300-FX/FX2/FX3 and 9300-GX platform switches.

	Command or Action	Purpose
Step 2	**interface nve 1**	Create the network virtualization endpoint (NVE) interface.
Step 3	**global suppress-arp**	Configure to suppress ARP globally for all Layer 2 VNI.within the NVE interface.
Step 4	**member vni** *vni-id*	Specify VNI ID.
Step 5	**suppress-arp**	Configure to suppress ARP under Layer 2 VNI and overrides the global set default.
Step 6	**suppress-arp disable**	Disables the global setting of the ARP suppression on a specific VNI.

Disabling VXLANs

SUMMARY STEPS

1. **configure terminal**
2. **no nv overlay evpn**
3. **no feature vn-segment-vlan-based**
4. **no feature nv overlay**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	**no nv overlay evpn**	Disables EVPN control plane.
Step 3	**no feature vn-segment-vlan-based**	Disables the global mode for all VXLAN bridge domains
Step 4	**no feature nv overlay**	Disables the VXLAN feature.
Step 5	(Optional) **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Duplicate Detection for IP and MAC Addresses

For IP addresses:

Cisco NX-OS supports duplicate detection for IP addresses. This enables the detection of duplicate IP addresses based on the number of moves in a given time-interval (seconds), if host appears simultaneously under two VTEP's.

Simultaneous availability of host under two VTEP's is detected by host mobility logic with 600 msec refresh timeout for IPv4 hosts and default refresh time out logic for IPv6 addresses (default is 3 seconds).

The default is 5 moves in 180 seconds. (Default number of moves is 5 moves. Default time-interval is 180 seconds.)

After the 5th move within 180 seconds, the switch starts a 30 second lock (hold down timer) before checking to see if the duplication still exists (an effort to prevent an increment of the sequence bit). This 30 second lock can occur 5 times within 24 hours (this means 5 moves in 180 seconds for 5 times) before the switch permanently locks or freezes the duplicate entry. (**show fabric forwarding ip local-host-db vrf abc**)

Wherever a host IP address is permanently frozen, a syslog message is written by HMM.

```
2021 Aug 26 01:08:26 leaf hmm: (vrf-name) [IPv4] Freezing potential duplicate host
20.2.0.30/32, reached recover count (5) threshold
```

The following are example commands to help the configuration of the number of VM moves in a specific time interval (seconds) for duplicate IP-detection:

Command	Description

`switch(config)# fabric forwarding ?`
`anycast-gateway-mac`
`dup-host-ip-addr-detection`
	Available sub-commands:
	• Anycast gateway MAC of the switch.
	• To detect duplicate host addresses in n seconds.

`switch(config)# fabric forwarding dup-host-ip-addr-detection ?` `<1-1000>`	The number of host moves allowed in n seconds. The range is 1 to 1000 moves; default is 5 moves.
`switch(config)# fabric forwarding dup-host-ip-addr-detection 100 ?` `<2-36000>`	The duplicate detection timeout in seconds for the number of host moves. The range is 2 to 36000 seconds; default is 180 seconds.
`switch(config)# fabric forwarding dup-host-ip-addr-detection 100 10`	Detects duplicate host addresses (limited to 100 moves) in a period of 10 seconds.

For MAC addresses:

Cisco NX-OS supports duplicate detection for MAC addresses. This enables the detection of duplicate MAC addresses based on the number of moves in a given time-interval (seconds).

The default is 5 moves in 180 seconds. (Default number of moves is 5 moves. Default time-interval is 180 seconds.)

After the 5th move within 180 seconds, the switch starts a 30 second lock (hold down timer) before checking to see if the duplication still exists (an effort to prevent an increment of the sequence bit). This 30 second lock can occur 3 times within 24 hours (this means 5 moves in 180 seconds for 3 times) before the switch permanently locks or freezes the duplicate entry. (**show l2rib internal permanently-frozen-list**)

Wherever a MAC address is permanently frozen, a syslog message with written by L2RIB.

```
2017 Jul  5 10:27:34 leaf %$ VDC-1 %$  %USER-2-SYSTEM_MSG: Unfreeze limit (3) hit, MAC
0000.0033.3333in topo: 200 is permanently frozen - l2rib
2017 Jul  5 10:27:34 leaf %$ VDC-1 %$  %USER-2-SYSTEM_MSG: Detected duplicate host
```

```
0000.0033.3333, topology 200, during Local update, with host located at remote VTEP 1.2.3.4,
 VNI 2 - l2rib
2017 Jul  5 10:27:34 leaf %$ VDC-1 %$  %USER-2-SYSTEM_MSG: Unfreeze limit (3) hit, MAC
0000.0033.3334in topo: 200 is permanently frozen - l2rib
2017 Jul  5 10:27:34 leaf %$ VDC-1 %$  %USER-2-SYSTEM_MSG: Detected duplicate host
0000.0033.3334, topology 200, during Local update, with host l
```

MAC address remains in permanently frozen list until both local and remote entry exists.

Unconfiguring below commands will not disable permanently frozen functionality rather will change the parameters to default values.

- **l2rib dup-host-mac-detection**
- **l2rib dup-host-recovery**

The following are example commands to help the configuration of the number of VM moves in a specific time interval (seconds) for duplicate MAC-detection:

Command	Description

`switch(config)# l2rib dup-host-mac-detection ?`
`<1-1000>`
`default`
	Available sub-commands for L2RIB:
	• The number of host moves allowed in n seconds. The range is 1 to 1000 moves.
	• Default setting (5 moves in 180 in seconds).

`switch(config)# l2rib dup-host-mac-detection 100 ?` `<2-36000>`	The duplicate detection timeout in seconds for the number of host moves. The range is 2 to 36000 seconds; default is 180 seconds.
`switch(config)# l2rib dup-host-mac-detection 100 10`	Detects duplicate host addresses (limited to 100 moves) in a period of 10 seconds.

Configuring Event History Size for L2RIB

To set the event history size for the L2RIB component follow these steps:

SUMMARY STEPS

1. **configure terminal**
2. **l2rib event-history** { **mac** | **mac-ip** | **loop-detection** } **size** { **default** | **medium** | **high** | **very-high** }
3. **l2rib event-history** { **fl** | **imet** | **dme-oper** } **size** { **default** | **medium** | **high** | **very-high** }
4. **clear l2rib event-history** { **mac** | **mac-ip** | **loop-detection** } **size** { **default** | **medium** | **high** | **very-high** }

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enter global configuration mode.
Step 2	**l2rib event-history** { **mac** | **mac-ip** | **loop-detection** } **size** { **default** | **medium** | **high** | **very-high** } **Example:** `switch(config)# l2rib event-history mac size low`	Sets the event history size for the L2RIB component.
Step 3	**l2rib event-history** { **fl** | **imet** | **dme-oper** } **size** { **default** | **medium** | **high** | **very-high** } **Example:** `switch(config)# l2rib event-history fl size very-high`	Generates the event logs for specified L2RIB objects: • **fl**: L2RIB VXLAN Flood-list • **imet**: L2RIB IMET • **dme-oper**: L2RIB DME OPER **Note** Ensure to enable buffer size to **very-high** in large scaling environment.
Step 4	**clear l2rib event-history** { **mac** | **mac-ip** | **loop-detection** } **size** { **default** | **medium** | **high** | **very-high** } **Example:** `switch(config)# clear l2rib event-history mac size low`	Clears the set event history size for the L2RIB component.

Verifying the VXLAN BGP EVPN Configuration

To display the VXLAN BGP EVPN configuration information, enter one of the following commands:

Command	Purpose
show nve vrf	Displays VRFs and associated VNIs
show bgp l2vpn evpn	Displays routing table information.
show ip arp suppression-cache [**detail** | **summary** | **vlan** *vlan* | **statistics**]	Displays ARP suppression information.
show vxlan interface	Displays VXLAN interface status.

Command	Purpose
show vxlan interface | count	Displays VXLAN VLAN logical port VP count. **Note** A VP is allocated on a per-port per-VLAN basis. The sum of all VPs across all VXLAN-enabled Layer 2 ports gives the total logical port VP count. For example, if there are 10 Layer 2 trunk interfaces, each with 10 VXLAN VLANs, then the total VXLAN VLAN logical port VP count is 10*10 = 100.
show l2route evpn mac [**all** | **evi** *evi* [**bgp** | **local** | **static** | **vxlan** | **arp**]]	Displays Layer 2 route information.
show l2route evpn fl all	Displays all fl routes.
show l2route evpn imet all	Displays all imet routes.
show l2route evpn mac-ip all **show l2route evpn mac-ip all detail**	Displays all MAC IP routes.
show l2route topology	Displays Layer 2 route topology.

Note Although the **show ip bgp** command is available for verifying a BGP configuration, as a best practice, it is preferable to use the **show bgp** command instead.

Verifying the VXLAN EVPN with Downstream VNI Configuration

To display the VXLAN EVPN with downstream VNI configuration information, enter one of the following commands:

Command	Purpose
show bgp evi *l2-evi*	Displays the VRF associated with an L2VNI.
show forwarding adjacency nve platform	Displays both symmetric and asymmetric NVE adjacencies with the corresponding DestInfoIndex.
show forwarding route vrf *vrf*	Displays the egress VNI or downstream VNI for each next-hop.
show ip route detail vrf *vrf*	Displays the egress VNI or downstream VNI for each next-hop.
show l2route evpn mac-ip all detail	Displays labeled next-hops that are present in the remote MAC routes.

Command	Purpose
show l2route evpn imet all detail	Displays the egress VNI associated with the remote peer.
show nve peers control-plane-vni peer-ip *ip-address*	Displays the egress VNI or downstream VNI for each NVE adjacency.

The following example shows sample output for the **show bgp evi** *l2-evi* command:

```
switch# show bgp evi 100
-----------------------------------------------
  L2VNI ID                      : 100 (L2-100)
  RD                            : 3.3.3.3:32867
  Secondary RD                  : 1:100
  Prefixes (local/total)        : 1/6
  Created                       : Jun 23 22:35:13.368170
  Last Oper Up/Down             : Jun 23 22:35:13.369005 / never
  Enabled                       : Yes
  Associated IP-VRF             : vni100
  Active Export RT list         :
        100:100
  Active Import RT list         :
        100:100
```

The following example shows sample output for the **show forwarding adjacency nve platform** command:

```
switch# show forwarding adjacency nve platform
slot  1
=======
IPv4 NVE adjacency information

next_hop:12.12.12.12   interface:nve1 (0x49000001)  table_id:1
  Peer_id:0x49080002  dst_addr:12.12.12.12  src_addr:13.13.13.13  RefCt:1  PBRCt:0
Flags:0x440800
cp : TRUE, DCI peer: FALSE is_anycast_ip FALSE dsvni peer: FALSE
  HH:0x7a13f  DstInfoIndex:0x3002
    tunnel init: unit-0:0x3 unit-1:0x0

next_hop:12.12.12.12   interface:nve1 (0x49000001)  table_id:1
  Peer_id:0x49080002  dst_addr:12.12.12.12  src_addr:13.13.13.13  RefCt:1  PBRCt:0
Flags:0x10440800
cp : TRUE, DCI peer: FALSE is_anycast_ip FALSE dsvni peer: TRUE
  HH:0x7a142  DstInfoIndex:0x3ffd
    tunnel init: unit-0:0x6 unit-1:0x0
...
```

The following example shows sample output for the **show forwarding route vrf** *vrf* command:

```
switch# show forwarding route vrf vrf1000

slot  1
=======

IPv4 routes for table vrf1000/base

--------------+------------------+--------------+-----------------+-----------------
Prefix        | Next-hop         | Interface    | Labels          | Partial Install
--------------+------------------+--------------+-----------------+-----------------
…..
10.1.1.11/32    12.12.12.12         nve1        dsvni: 301000
```

```
10.1.1.20/32    123.123.123.123     nve1        dsvni: 301000
10.1.1.21/32    30.30.30.30         nve1        dsvni: 301000
10.1.1.30/32    10.1.1.30           Vlan10
```

The following example shows sample output for the **show ip route detail vrf** *vrf* command:

```
switch# show ip route detail vrf default
IP Route Table for VRF "default"
  '*' denotes best ucast next-hop
  '**' denotes best mcast next-hop
  '[x/y]' denotes [preference/metric]
  '%<string>' in via output denotes VRF <string>

193.0.1.0/24, ubest/mbest: 4/0
      *via 30.1.0.2, Eth1/1, [100/0], 00:00:05, urib_dt6-client1 segid: 6544, tunnelid:
0x7b9 encap: VXLAN

      *via 30.1.1.2, Eth1/1, [100/0], 00:00:05, urib_dt6-client1 segid: 6545, (Asymmetric)
 tunnelid: 0x7ba encap: VXLAN

      *via 30.1.2.2, Eth1/1, [100/0], 00:00:05, urib_dt6-client1 segid: 6546, (Asymmetric)
 tunnelid: 0x7bb encap: VXLAN
```

The following example shows sample output for the **show l2route evpn mac-ip all detail** command:

```
switch# show l2route evpn mac-ip all
Flags -(Rmac):Router MAC (Stt):Static (L):Local (R):Remote (V):vPC link
(Dup):Duplicate (Spl):Split (Rcv):Recv(D):Del Pending (S):Stale (C):Clear
(Ps):Peer Sync (Ro):Re-Originated (Orp):Orphan
Topology Mac Address    Host IP   Prod   Flags Seq No  Next-Hops
-------- -------------- --------  ------ ----- ------- ----------------------------
5        0000.0005.1301 1.3.13.1  BGP    --     0      102.1.13.1 (Label: 2000005)
5        0000.0005.1401 1.3.14.1  BGP    --     0      102.1.145.1 (Label: 2000005)
```

The following example shows sample output for the **show l2route evpn imet all detail** command:

```
switch# show l2route evpn imet all

Flags- (F): Originated From Fabric, (W): Originated from WAN

Topology ID VNI         Prod  IP Addr     Flags
----------- ----------- ----- ----------- ---------
3           2000003     BGP   102.1.13.1    -
3           2000003     BGP   102.1.31.1    -
3           2000003     BGP   102.1.32.1    -
3           2000003     BGP   102.1.145.1   -
```

The following example shows sample output for the **show nve peers control-plane-vni** command. In this example, 3000003 is the downstream VNI.

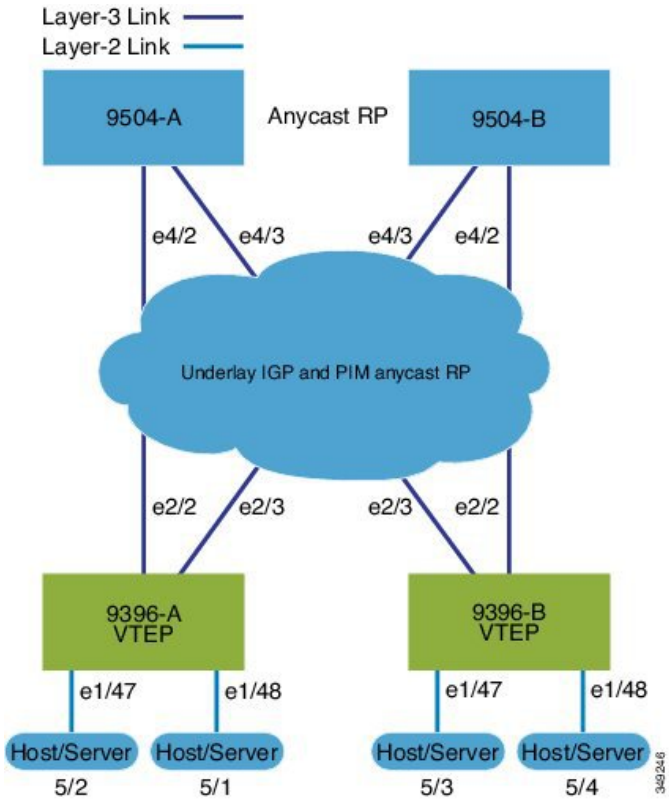
```
switch# show nve peers control-plane-vni peer-ip 203.1.1.1
Peer       VNI     Learn-Source Gateway-MAC     Peer-type  Egress-VNI SW-BD  State

---------  -----   ------------ --------------- ---------- ---------- -----
----------------------
203.1.1.1  2000003 BGP          f40f.1b6f.f8db   FAB        3000003      3005
peer-vni-add-complete
```

Example of VXLAN BGP EVPN (IBGP)

An example of a VXLAN BGP EVPN (IBGP):

Figure 4: VXLAN BGP EVPN Topology (IBGP)



IBGP between Spine and Leaf

- Spine (9504-A)
  - Enable the EVPN control plane

    ```
    nv overlay evpn
    ```

  - Enable the relevant protocols

    ```
    feature ospf
    feature bgp
    feature pim
    ```

  - Configure Loopback for local Router ID, PIM, and BGP

    ```
    interface loopback0
      ip address 10.1.1.1/32
      ip router ospf 1 area 0.0.0.0
      ip pim sparse-mode
    ```

  - Configure Loopback for local VTEP IP, and BGP

    ```
    interface loopback0
    ```

```
  ip address 10.1.1.1/32
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
```

- Configure Loopback for Anycast RP

```
interface loopback1
  ip address 100.1.1.1/32
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
```

- Configure Anycast RP

```
ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
ip pim anycast-rp 100.1.1.1 10.1.1.1
ip pim anycast-rp 100.1.1.1 20.1.1.1
```

- Enable OSPF for underlay routing

```
router ospf 1
```

- Configure interfaces for Spine-leaf interconnect

```
interface Ethernet4/2
  ip address 192.168.1.42/24
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface Ethernet4/3
  ip address 192.168.2.43/24
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
  no shutdown
```

- Configure BGP

```
router bgp 65535
router-id 10.1.1.1
  neighbor 30.1.1.1 remote-as 65535
    update-source loopback0
    address-family l2vpn evpn
      send-community both
      route-reflector-client
  neighbor 40.1.1.1 remote-as 65535
    update-source loopback0
    address-family l2vpn evpn
      send-community both
      route-reflector-client
```

- Spine (9504-B)
  - Enable the EVPN control plane

```
nv overlay evpn
```

  - Enable the relevnt Protocols

```
feature ospf
feature bgp
feature pim
```

- Configure Loopback for local Router ID, PIM, and BGP

```
interface loopback0
  ip address 20.1.1.1/32
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
```

- Configure Loopback for local VTEP IP, and BGP

```
interface loopback0
  ip address 20.1.1.1/32
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
```

- Configure Loopback for AnycastRP

```
interface loopback1
  ip address 100.1.1.1/32
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
```

- Configure Anycast RP

```
ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
ip pim anycast-rp 100.1.1.1 10.1.1.1
ip pim anycast-rp 100.1.1.1 20.1.1.1
```

- Enable OSPF for underlayrouting

```
router ospf 1
```

- Configure interfaces for Spine-leaf interconnect

```
interface Ethernet4/2
  ip address 192.168.3.42/24
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface Ethernet4/3
  ip address 192.168.4.43/24
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
  no shutdown
```

- Configure BGP

```
router bgp 65535
  router-id 20.1.1.1
  neighbor 30.1.1.1 remote-as 65535
    update-source loopback0
    address-family l2vpn evpn
      send-community both
      route-reflector client
  neighbor 40.1.1.1 remote-as 65535
    update-source loopback0
    address-family l2vpn evpn
```

```
        send-community both
        route-reflector client
```

- Leaf (9396-A)
  - Enable the EVPN control plane

```
nv overlay evpn
```

  - Enable the relevant protocols

```
feature ospf
feature bgp
feature pim
feature interface-vlan
```

  - Enable VXLAN with distributed anycast-gateway using BGP EVPN

```
feature vn-segment-vlan-based
feature nv overlay
fabric forwarding anycast-gateway-mac 0000.2222.3333
```

  - Enabling OSPF for underlay routing

```
router ospf 1
```

  - Configure Loopback for local Router ID, PIM, and BGP

```
interface loopback0
  ip address 30.1.1.1/32
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
```

  - Configure Loopback for local VTEP IP, and BGP

```
interface loopback0
  ip address 30.1.1.1/32
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
```

  - Configure interfaces for Spine-leaf interconnect

```
interface Ethernet2/2
  no switchport
  ip address 192.168.1.22/24
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface Ethernet2/3
  no switchport
  ip address 192.168.3.23/24
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
  shutdown
```

  - Configure route-map to Redistribute Host-SVI (Silent Host)

```
route-map HOST-SVI permit 10
  match tag 54321
```

- Configure PIM RP

```
ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
```

- Create VLANs

```
vlan 1001-1002
```

- Create overlay VRF VLAN and configure vn-segment

```
vlan 101
  vn-segment 900001
```

- Create overlay VRF VLAN and configure vn-segment

```
vlan 101
  vn-segment 900001
```

- Configure Core-facing SVI for VXLAN routing

```
interface vlan101
 no shutdown
  vrf member vxlan-900001
  ip forward
  no ip redirects
  ipv6 address use-link-local-only
  no ipv6 redirects
```

- Create VLAN and provide mapping to VXLAN

```
vlan 1001
  vn-segment 2001001
vlan 1002
  vn-segment 2001002
```

- Create VRF and configure VNI

```
vrf context vxlan-900001
  vni 900001
  rd auto
```

Note The **rd auto** and **route-target** commands are automatically configured unless one or more are entered as overrides.

```
\
address-family ipv4 unicast
      route-target both auto
      route-target both auto evpn
    address-family ipv6 unicast
```

```
      route-target both auto
      route-target both auto evpn
```

- Create server facing SVI and enable distributed anycast-gateway.

```
interface vlan1001
  no shutdown
  vrf member vxlan-900001
  ip address 4.1.1.1/24 tag 54321
  ipv6 address 4:1:0:1::1/64 tag 54321
  fabric forwarding mode anycast-gateway

interface vlan1002
  no shutdown
  vrf member vxlan-900001
  ip address 4.2.2.1/24 tag 54321
  ipv6 address 4:2:0:1::1/64 tag 54321
  fabric forwarding mode anycast-gateway
```

- Configure ACL TCAM region for ARP suppression

Note The **hardware access-list tcam region arp-ether 256 double-wide** command is not needed for Cisco Nexus 9300-EX and 9300-FX/FX2/FX3 and 9300-GX platform switches.

```
hardware access-list tcam region arp-ether 256 double-wide
```

-

Note You can choose either of the following two options for creating the NVE interface. Use Option 1 for a small number of VNIs. Use Option 2 to leverage the simplified configuration mode.

Create the network virtualization endpoint (NVE) interface

Option 1

```
interface nve1
no shutdown
  source-interface loopback1
  host-reachability protocol bgp
  member vni 900001 associate-vrf
  member vni 2001001
    mcast-group 239.0.0.1
  member vni 2001002
    mcast-group 239.0.0.1
```

Option 2

```
interface nve1
  source-interface loopback1
  host-reachability protocol bgp
  global mcast-group 239.0.0.1 L2
```

```
  member vni 2001001
  member vni 2001002
  member vni 2001007-2001010
```

- Configure interfaces for hosts/servers

```
interface Ethernet1/47
  switchport
  switchport access vlan 1002

interface Ethernet1/48
  switchport
  switchport access vlan 1001
```

- Configure BGP

```
router bgp 65535
  router-id 30.1.1.1
  neighbor 10.1.1.1 remote-as 65535
    update-source loopback0
    address-family l2vpn evpn
      send-community both
  neighbor 20.1.1.1 remote-as 65535
    update-source loopback0
    address-family l2vpn evpn
      send-community both
  vrf vxlan-900001
    address-family ipv4 unicast
      redistribute direct route-map HOST-SVI
    address-family ipv6 unicast
      redistribute direct route-map HOST-SVI
```

Note The following commands in EVPN mode do not need to be entered.

```
evpn
  vni 2001001 l2
  vni 2001002 l2
```

Note The **rd auto** and **route-target auto** commands are automatically configured unless one or more are entered as overrides.

```
rd auto
    route-target import auto
    route-target export auto
```

Note The **rd auto** and **route-target** commands are automatically configured unless you want to use them to override the **import** or **export** options.

Note The following commands in EVPN mode do not need to be entered.

```
evpn
  vni 2001001 l2
    rd auto
    route-target import auto
    route-target export auto
  vni 2001002 l2
    rd auto
    route-target import auto
    route-target export auto
```

- Leaf (9396-B)
  - Enable the EVPN control plane

```
nv overlay evpn
```

  - Enable the relevant protocols

```
feature ospf
feature bgp
feature pim
feature interface-vlan
```

  - Enable VxLAN with distributed anycast-gateway using BGP EVPN

```
feature vn-segment-vlan-based
feature nv overlay
fabric forwarding anycast-gateway-mac 0000.2222.3333
```

  - Enabling OSPF for underlayrouting

```
router ospf 1
```

  - Configure Loopback for local Router ID, PIM, and BGP

```
interface loopback0
  ip address 40.1.1.1/32
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
```

  - Configure Loopback for local VTEP IP, and BGP

```
interface loopback0
  ip address 40.1.1.1/32
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
```

  - Configure interfaces for Spine-leaf interconnect

```
interface Ethernet2/2
  no switchport
  ip address 192.168.3.22/24
  ip router ospf 1 area 0.0.0.0
```

```
  ip pim sparse-mode
  no shutdown

interface Ethernet2/3
  no switchport
  ip address 192.168.4.23/24
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
  shutdown
```

- Configure route-map to Redistribute Host-SVI (Silent Host)

```
route-map HOST-SVI permit 10
  match tag 54321
```

- Configure PIM RP

```
ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
```

- Create VLANs

```
vlan 1001-1002
```

- Create overlay VRF VLAN and configure vn-segment

```
vlan 101
  vn-segment 900001
```

- Configure Core-facing SVI for VXLAN routing

```
interface vlan101
 no shutdown
  vrf member vxlan-900001
  ip forward
  no ip redirects
  ipv6 address use-link-local-only
  no ipv6 redirects
```

- Create VLAN and provide mapping to VXLAN

```
vlan 1001
  vn-segment 2001001
vlan 1002
  vn-segment 2001002
```

- Create VRF and configure VNI

```
vrf context vxlan-900001
  vni 900001
  rd auto
```

Note The **rd auto** and **route-target** commands are automatically configured unless one or more are entered as overrides.

```
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
  address-family ipv6 unicast
    route-target both auto
    route-target both auto evpn
```

- Create server facing SVI and enable distributed anycast-gateway

```
interface vlan1001
  no shutdown
  vrf member vxlan-900001
  ip address 4.1.1.1/24
  ipv6 address 4:1:0:1::1/64
  fabric forwarding mode anycast-gateway

interface vlan1002
  no shutdown
  vrf member vxlan-900001
  ip address 4.2.2.1/24
  ipv6 address 4:2:0:1::1/64
  fabric forwarding mode anycast-gateway
```

- Configure ACL TCAM region for ARP suppression

Note The **hardware access-list tcam region arp-ether 256 double-wide** command is not needed for Cisco Nexus 9300-EX and 9300-FX/FX2/FX3 and 9300-GX platform switches.

```
hardware access-list tcam region arp-ether 256 double-wide
```

-

Note You can choose either of the following two command procedures for creating the NVE interfaces. Use Option 1 for a small number of VNIs. Use Option 2 to leverage the simplified configuration mode.

Create the network virtualization endpoint (NVE) interface

Option 1

```
interface nve1
  no shutdown
  source-interface loopback1
  host-reachability protocol bgp
  member vni 900001 associate-vrf
  member vni 2001001
    mcast-group 239.0.0.1
  member vni 2001002
    mcast-group 239.0.0.1
```

Option 2

```
interface nve1
  interface nve1
  source-interface loopback1
  host-reachability protocol bgp
  global mcast-group 239.0.0.1 L2
  member vni 2001001
  member vni 2001002
  member vni 2001007-2001010
```

- Configure interfaces for hosts/servers

```
interface Ethernet1/47
  switchport
  switchport access vlan 1002

interface Ethernet1/48
  switchport
  switchport access vlan 1001
```

- Configure BGP

```
router bgp 65535
  router-id 40.1.1.1
  neighbor 10.1.1.1 remote-as 65535
    update-source loopback0
    address-family l2vpn evpn
      send-community both
  neighbor 20.1.1.1 remote-as 65535
    update-source loopback0
    address-family l2vpn evpn
      send-community both
  vrf vxlan-900001
  vrf vxlan-900001
    address-family ipv4 unicast
      redistribute direct route-map HOST-SVI
    address-family ipv6 unicast
      redistribute direct route-map HOST-SVI
```

Note The following commands in EVPN mode do not need to be entered.

```
evpn
  vni 2001001 l2
  vni 2001002 l2
```

Note The **rd auto** and **route-target** commands are automatically configured unless one or more are entered as overrides.

```
rd auto
    route-target import auto
    route-target export auto
```

Note The following commands in EVPN mode do not need to be entered.

```
evpn
  vni 2001001 l2
    rd auto
    route-target import auto
    route-target export auto
  vni 2001002 l2
    rd auto
    route-target import auto
    route-target export auto
```

- Configure interface vlan on Border Gateway (BGW)

```
interface vlan101
  no shutdown
  vrf member evpn-tenant-3103101
  no ip redirects
  ip address 101.1.0.1/16
  ipv6 address cafe:101:1::1/48
  no ipv6 redirects
  fabric forwarding mode anycast-gateway
```

Note When you have IBGP session between BGWs and EBGP fabric is used, you need to configure the route-map to make VIP or VIP_R route advertisement with higher AS-PATH when local VIP or VIP_R is down (due to reload or fabric link flap). A sample route-map configuration is provided below. In this example 192.0.2.1 is VIP address and 198.51.100.1 is BGP VIP route's nexthop learned from same BGW site.

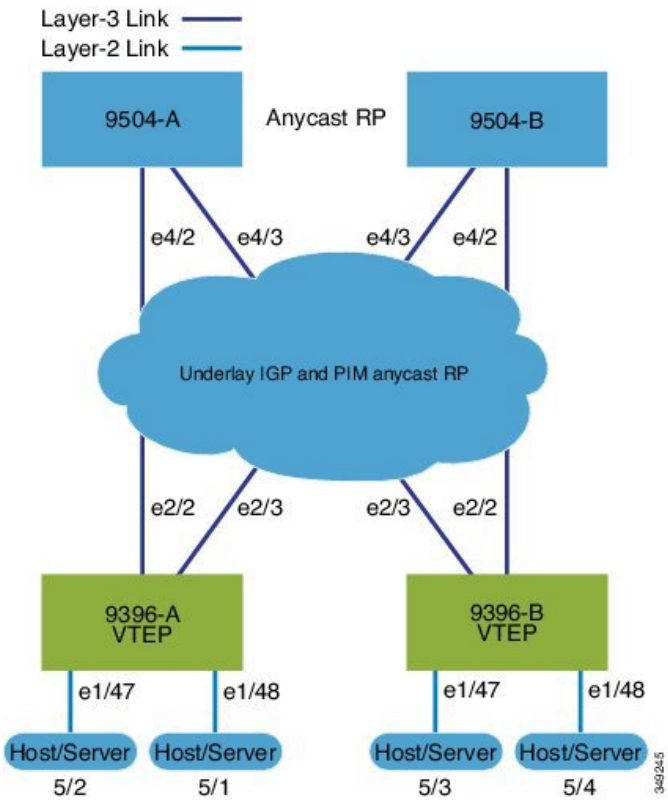
```
ip prefix-list vip_ip seq 5 permit 192.0.2.1/32
ip prefix-list vip_route_nh seq 5 permit 198.51.100.1/32

route-map vip_ip permit 5
  match ip address prefix-list vip_ip
  match ip next-hop prefix-list vip_route_nh
  set as-path prepend 5001 5001 5001
route-map vip_ip permit 10
```

Example of VXLAN BGP EVPN (EBGP)

An example of a VXLAN BGP EVPN (EBGP):

Figure 5: VXLAN BGP EVPN Topology (EBGP)



EBGP between Spine and Leaf

- Spine (9504-A)
  - Enable the EVPN control plane

    ```
    nv overlay evpn
    ```

  - Enable the relevant protocols

    ```
    feature bgp
    feature pim
    ```

  - Configure Loopback for local Router ID, PIM, and BGP

    ```
    interface loopback0
      ip address 10.1.1.1/32 tag 12345
      ip pim sparse-mode
    ```

  - Configure Loopback for Anycast RP

    ```
    interface loopback1
      ip address 100.1.1.1/32 tag 12345
      ip pim sparse-mode
    ```

  - Configure Anycast RP

    ```
    ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
    ```

```
ip pim anycast-rp 100.1.1.1 10.1.1.1
ip pim anycast-rp 100.1.1.1 20.1.1.1
```

- Configure route-map used by EBGP for Spine

```
route-map NEXT-HOP-UNCH permit 10
  set ip next-hop unchanged
```

- Configure route-map to Redistribute Loopback

```
route-map LOOPBACK permit 10
  match tag 12345
```

- Configure interfaces for Spine-leaf interconnect

```
interface Ethernet4/2
  ip address 192.168.1.42/24
  ip pim sparse-mode
  no shutdown

interface Ethernet4/3
  ip address 192.168.2.43/24
  ip pim sparse-mode
  no shutdown
```

- Configure the BGP overlay for the EVPN address family.

```
router bgp 100
  router-id 10.1.1.1
  address-family l2vpn evpn
    nexthop route-map NEXT-HOP-UNCH
    retain route-target all
  neighbor 30.1.1.1 remote-as 200
    update-source loopback0
    ebgp-multihop 3
    address-family l2vpn evpn
      send-community both
      disable-peer-as-check
      route-map NEXT-HOP-UNCH out
  neighbor 40.1.1.1 remote-as 200
    update-source loopback0
    ebgp-multihop 3
    address-family l2vpn evpn
      send-community both
      disable-peer-as-check
      route-map NEXT-HOP-UNCH out
```

- Configure BGP underlay for the IPv4 unicast address family.

```
 address-family ipv4 unicast
    redistribute direct route-map LOOPBACK
  neighbor 192.168.1.22 remote-as 200
    update-source ethernet4/2
    address-family ipv4 unicast
      allowas-in
      disable-peer-as-check
  neighbor 192.168.2.23 remote-as 200
    update-source ethernet4/3
    address-family ipv4 unicast
```

```
      allowas-in
      disable-peer-as-check
```

- Spine (9504-B)
  - Enable the EVPN control plane

```
nv overlay evpn
```

  - Enable the relevant protocols

```
feature bgp
feature pim
```

  - Configure Loopback for local Router ID, PIM, and BGP

```
interface loopback0
  ip address 20.1.1.1/32 tag 12345
  ip pim sparse-mode
```

  - Configure Loopback for AnycastRP

```
interface loopback1
  ip address 100.1.1.1/32 tag 12345
  ip pim sparse-mode
```

  - Configure Anycast RP

```
ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
ip pim anycast-rp 100.1.1.1 10.1.1.1
ip pim anycast-rp 100.1.1.1 20.1.1.1
```

  - Configure route-map used by EBGP for Spine

```
route-map NEXT-HOP-UNCH permit 10
  set ip next-hop unchanged
```

  - Configure route-map to Redistribute Loopback

```
route-map LOOPBACK permit 10
  match tag 12345
```

  - Configure interfaces for Spine-leaf interconnect

```
interface Ethernet4/2
  no switchport
  ip address 192.168.3.42/24
  ip router ospf 1 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface Ethernet4/3
  no switchport
  ip address 192.168.4.43/24
  ip router ospf 1 area 0.0.0.0
```

```
  ip pim sparse-mode
  shutdown
```

- Configure BGP overlay for the EVPN address family

```
router bgp 100
  router-id 20.1.1.1
  address-family l2vpn evpn
    nexthop route-map NEXT-HOP-UNCH
    retain route-target all
  neighbor 30.1.1.1 remote-as 200
    update-source loopback0
    ebgp-multihop 3
    address-family l2vpn evpn
      send-community both
      disable-peer-as-check
      route-map NEXT-HOP-UNCH out
  neighbor 40.1.1.1 remote-as 200
    update-source loopback0
    ebgp-multihop 3
    address-family l2vpn evpn
      send-community both
      disable-peer-as-check
      route-map NEXT-HOP-UNCH out
```

- Configure the BGP underlay for the IPv4 unicast address family.

```
  address-family ipv4 unicast
    redistribute direct route-map LOOPBACK
  neighbor 192.168.3.22 remote-as 200
    update-source ethernet4/2
    address-family ipv4 unicast
      allowas-in
      disable-peer-as-check
  neighbor 192.168.4.43 remote-as 200
    update-source ethernet4/3
    address-family ipv4 unicast
      allowas-in
      disable-peer-as-check
```

- Leaf (9396-A)
  - Enable the EVPN control plane.

    ```
    nv overlay evpn
    ```

  - Enable the relevant protocols.

    ```
    feature bgp
    feature pim
    feature interface-vlan
    ```

  - Enable VXLAN with distributed anycast-gateway using BGP EVPN.

    ```
    feature vn-segment-vlan-based
    feature nv overlay
    ```

```
fabric forwarding anycast-gateway-mac 0000.2222.3333
```

- Enabling OSPF for underlay routing.

```
router ospf 1
```

- Configure Loopback for local Router ID, PIM, and BGP.

```
interface loopback0
  ip address 30.1.1.1/32
  ip pim sparse-mode
```

- Configure Loopback for VTEP.

```
interface loopback1
  ip address 33.1.1.1/32
  ip pim sparse-mode
```

- Configure interfaces for Spine-leafi nterconnect.

```
interface Ethernet2/2
  no switchport
  ip address 192.168.1.22/24
  ip pim sparse-mode
  no shutdown

interface Ethernet2/3
  no switchport
  ip address 192.168.4.23/24
  ip pim sparse-mode
  shutdown
```

- Configure route-map to Redistribute Host-SVI (Silent Host).

```
route-map HOST-SVI permit 10
  match tag 54321
```

- Enable PIM RP.

```
ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
```

- Create VLANs.

```
vlan 1001-1002
```

- Create overlay VRF VLAN and configure vn-segment.

```
vlan 101
  vn-segment 900001
```

- Configure core-facing SVI for VXLAN routing.

```
interface vlan101
 no shutdown
  vrf member vxlan-900001
  ip forward
  no ip redirects
  ipv6 address use-link-local-only
  no ipv6 redirects
```

- Create VLAN and provide mapping toVXLAN.

```
vlan 1001
  vn-segment 2001001
vlan 1002
  vn-segment 2001002
```

- Create VRF and configure VNI

```
vrf context vxlan-900001
  vni 900001
  rd auto
```

Note The **rd auto** and **route-target** commands are automatically configured unless one or more are entered as overrides.

```
    address-family ipv4 unicast
      route-target both auto
      route-target both auto evpn
    address-family ipv6 unicast
      route-target both auto
      route-target both auto evpn
```

- Create server facing SVI and enable distributed anycast-gateway

```
interface vlan1001
  no shutdown
  vrf member vxlan-900001
  ip address 4.1.1.1/24 tag 54321
  ipv6 address 4:1:0:1::1/64 tag 54321
  fabric forwarding mode anycast-gateway

interface vlan1002
  no shutdown
  vrf member vxlan-900001
  ip address 4.2.2.1/24 tag 54321
  ipv6 address 4:2:0:1::1/64 tag 54321
  fabric forwarding mode anycast-gateway
```

- Configure ACL TCAM region for ARP suppression

Note The **hardware access-list tcam region arp-ether 256 double-wide** command is not needed for Cisco Nexus 9300-EX and 9300-FX/FX2/FX3 and 9300-GX platform switches.

```
hardware access-list tcam region arp-ether 256 double-wide
```

-

Note You can choose either of the following two options for creating the NVE interface. Use Option 1 for a small number of VNIs. Use Option 2 to leverage the simplified configuration mode.

Create the network virtualization endpoint (NVE) interface

Option 1

```
interface nve1
  no shutdown
  source-interface loopback1
  host-reachability protocol bgp
  member vni 900001 associate-vrf
  member vni 2001001
    mcast-group 239.0.0.1
  member vni 2001002
    mcast-group 239.0.0.1
```

Option 2

```
interface nve1
  source-interface loopback1
  host-reachability protocol bgp
  global mcast-group 239.0.0.1 L2
  member vni 2001001
  member vni 2001002
  member vni 2001007-2001010
```

- Configure interfaces for hosts/servers.

```
interface Ethernet1/47
  switchport
  switchport access vlan 1002

interface Ethernet1/48
  switchport
  switchport access vlan 1001
```

- Configure BGP underlay for the IPv4 unicast address family.

```
router bgp 200
  router-id 30.1.1.1
  address-family ipv4 unicast
    redistribute direct route-map LOOPBACK
  neighbor 192.168.1.42 remote-as 100
    update-source ethernet2/2
    address-family ipv4 unicast
      allowas-in
      disable-peer-as-check
```

```
  neighbor 192.168.4.43 remote-as 100
    update-source ethernet2/3
    address-family ipv4 unicast
      allowas-in
      disable-peer-as-check
```

- Configure BGP overlay for the EVPN address family.

```
  address-family l2vpn evpn
    nexthop route-map NEXT-HOP-UNCH
    retain route-target all
  neighbor 10.1.1.1 remote-as 100
    update-source loopback0
    ebgp-multihop 3
    address-family l2vpn evpn
      send-community both
      disable-peer-as-check
      route-map NEXT-HOP-UNCH out
  neighbor 20.1.1.1 remote-as 100
    update-source loopback0
    ebgp-multihop 3
    address-family l2vpn evpn
      send-community both
      disable-peer-as-check
      route-map NEXT-HOP-UNCH out
  vrf vxlan-900001
```

Note The following commands in EVPN mode do not need to be entered.

```
evpn
  vni 2001001 l2
  vni 2001002 l2
```

Note The **rd auto** and **route-target auto** commands are automatically configured unless one or more are entered as overrides.

```
rd auto
route-target import auto
route-target export auto
```

Note The following commands in EVPN mode do not need to be entered.

```
evpn
  vni 2001001 l2
    rd auto
    route-target import auto
    route-target export auto
  vni 2001002 l2
    rd auto
    route-target import auto
    route-target export auto
```

- Leaf (9396-B)
  - Enable the EVPN control plane.

```
nv overlay evpn
```

  - Enable the relevant protocols.

```
feature bgp
feature pim
feature interface-vlan
```

  - Enable VXLAN with distributed anycast-gateway using BGP EVPN.

```
feature vn-segment-vlan-based
feature nv overlay
fabric forwarding anycast-gateway-mac 0000.2222.3333
```

  - Enabling OSPF for underlay routing.

```
router ospf 1
```

  - Configure Loopback for local Router ID, PIM, and BGP.

```
interface loopback0
  ip address 40.1.1.1/32
  ip pim sparse-mode
```

  - Configure Loopback for VTEP.

```
interface loopback1
  ip address 44.1.1.1/32
  ip pim sparse-mode
```

  - Configure interfaces for Spine-leaf interconnect.

```
interface Ethernet2/2
  no switchport
  ip address 192.168.3.22/24
  ip pim sparse-mode
  no shutdown

interface Ethernet2/3
  no switchport
  ip address 192.168.2.23/24
  ip pim sparse-mode
  shutdown
```

  - Configure route-map to Redistribute Host-SVI (Silent Host).

```
route-map HOST-SVI permit 10
  match tag 54321
```

  - Enable PIM RP

```
ip pim rp-address 100.1.1.1 group-list 224.0.0.0/4
```

  - Create VLANs

```
vlan 1001-1002
```

- Create overlay VRF VLAN and configure vn-segment.

```
vlan 101
  vn-segment 900001
```

- Configure core-facing SVI for VXLAN routing.

```
interface vlan101
 no shutdown
  vrf member vxlan-900001
  ip forward
  no ip redirects
  ipv6 address use-link-local-only
  no ipv6 redirects
```

- Create VLAN and provide mapping to VXLAN.

```
vlan 1001
  vn-segment 2001001
vlan 1002
  vn-segment 2001002
```

- Create VRF and configure VNI

```
vrf context vxlan-900001
  vni 900001
  rd auto
```

Note The following commands are automatically configured unless one or more are entered as overrides.

```
    address-family ipv4 unicast
      route-target both auto
      route-target both auto evpn
    address-family ipv6 unicast
      route-target both auto
      route-target both auto evpn
```

- Create server facing SVI and enable distributed anycast-gateway.

```
interface vlan1001
  no shutdown
  vrf member vxlan-900001
  ip address 4.1.1.1/24 tag 54321
  ipv6 address 4:1:0:1::1/64 tag 54321
  fabric forwarding mode anycast-gateway

interface vlan1002
  no shutdown
  vrf member vxlan-900001
  ip address 4.2.2.1/24 tag 54321
  ipv6 address 4:2:0:1::1/64 tag 54321
  fabric forwarding mode anycast-gateway
```

- Configure ACL TCAM region for ARP suppression

Note The **hardware access-list tcam region arp-ether 256 double-wide** command is not needed for Cisco Nexus 9300-EX and 9300-FX/FX2/FX3 and 9300-GX platform switches.

```
hardware access-list tcam region arp-ether 256 double-wide
```

-

Note You can choose either of the following two procedures for creating the NVE interface. Use Option 1 for a small number of VNIs. Use Option 2 to leverage the simplified configuration mode.

Create the network virtualization endpoint (NVE) interface.

Option 1

```
interface nve1
  no shutdown
  source-interface loopback1
  host-reachability protocol bgp
  member vni 900001 associate-vrf
  member vni 2001001
    mcast-group 239.0.0.1
  member vni 2001002
    mcast-group 239.0.0.1
```

Option 2

```
interface nve1
  source-interface loopback1
  host-reachability protocol bgp
  global mcast-group 239.0.0.1 L2
  member vni 2001001
  member vni 2001002
  member vni 2001007-2001010
```

- Configure interfaces for hosts/servers

```
interface Ethernet1/47
  switchport
  switchport access vlan 1002

interface Ethernet1/48
  switchport
  switchport access vlan 1001
```

- Configure BGP underlay for the IPv4 unicast address family.

```
router bgp 200
  router-id 40.1.1.1
  address-family ipv4 unicast
    redistribute direct route-map LOOPBACK
  neighbor 192.168.3.42 remote-as 100
    update-source ethernet2/2
    address-family ipv4 unicast
      allowas-in
      disable-peer-as-check
  neighbor 192.168.2.43 remote-as 100
    update-source ethernet2/3
    address-family ipv4 unicast
      allowas-in
      disable-peer-as-check
```

- Configure BGP overlay for the EVPN address family.

```
  address-family l2vpn evpn
    nexthop route-map NEXT-HOP-UNCH
    retain route-target all
  neighbor 10.1.1.1 remote-as 100
    update-source loopback0
    ebgp-multihop 3
    address-family l2vpn evpn
      send-community both
      disable-peer-as-check
      route-map NEXT-HOP-UNCH out
  neighbor 20.1.1.1 remote-as 100
    update-source loopback0
    ebgp-multihop 3
    address-family l2vpn evpn
      send-community both
      disable-peer-as-check
      route-map NEXT-HOP-UNCH out
  vrf vxlan-900001
```

Note The following commands in EVPN mode do not need to be entered.

```
evpn
  vni 2001001 l2
  vni 2001002 l2
```

Note The **rd auto** and **route-target auto** commands are automatically configured unless one or more are entered as overrides.

```
rd auto
route-target import auto
route-target export auto
```

Note The following commands in EVPN mode do not need to be entered.

```
evpn
  vni 2001001 l2
    rd auto
    route-target import auto
    route-target export auto
  vni 2001002 l2
    rd auto
    route-target import auto
    route-target export auto
```

Example Show Commands

- **show nve peers**

```
9396-B# show nve peers
Interface Peer-IP          State LearnType Uptime   Router-Mac
--------- ---------------  ----- --------- -------- -----------------
nve1      30.1.1.1         Up    CP        00:00:38 6412.2574.9f27
```

- **show nve vni**

```
9396-B# show nve vni
Codes: CP - Control Plane        DP - Data Plane
       UC - Unconfigured

Interface VNI      Multicast-group   State Mode Type [BD/VRF]      Flags
--------- -------- ----------------- ----- ---- ------------------ -----
nve1      900001   n/a               Up    CP   L3 [vxlan-900001]
nve1      2001001  225.4.0.1         Up    CP   L2 [1001]
nve1      2001002  225.4.0.1         Up    CP   L2 [1002]
```

- **show ip arp suppression-cache detail**

```
9396-B# show ip arp suppression-cache detail

Flags: + - Adjacencies synced via CFSoE
       L - Local Adjacency
       R - Remote Adjacency
       L2 - Learnt over L2 interface

Ip Address      Age      Mac Address    Vlan Physical-ifindex    Flags

4.1.1.54        00:06:41 0054.0000.0000 1001 Ethernet1/48        L
4.1.1.51        00:20:33 0051.0000.0000 1001 (null)              R
4.2.2.53        00:06:41 0053.0000.0000 1002 Ethernet1/47        L
4.2.2.52        00:20:33 0052.0000.0000 1002 (null)              R
```

Note The **show vxlan interface** command is not supported for the Cisco Nexus 9300-EX, 9300-FX/FX2/FX3, and 9300-GX platform switches.

- **show vxlan interface**

```
9396-B# show vxlan interface
Interface       Vlan    VPL Ifindex     LTL             HW VP
=========       ====    ===========     ===             =====
Eth1/47         1002    0x4c07d22e      0x10000         5697
Eth1/48         1001    0x4c07d02f      0x10001         5698
```

- **show bgp l2vpn evpn summary**

```
leaf3# show bgp l2vpn evpn summary
BGP summary information for VRF default, address family L2VPN EVPN
BGP router identifier 40.0.0.4, local AS number 10
BGP table version is 60, L2VPN EVPN config peers 1, capable peers 1
21 network entries and 21 paths using 2088 bytes of memory
BGP attribute entries [8/1152], BGP AS path entries [0/0]
BGP community entries [0/0], BGP clusterlist entries [1/4]

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down
State/PfxRcd
40.0.0.1        4    10    8570    8565       60    0    0    5d22h 6
leaf3#
```

- **show bgp l2vpn evpn**

```
leaf3# show bgp l2vpn evpn
BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 60, local router ID is 40.0.0.4
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid,
>-best
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist,
I-injected
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup

   Network            Next Hop            Metric     LocPrf     Weight Path
Route Distinguisher: 40.0.0.2:32868
*>i[2]:[0]:[10001]:[48]:[0000.8816.b645]:[0]:[0.0.0.0]/216
                      40.0.0.2                           100          0 i
*>i[2]:[0]:[10001]:[48]:[0011.0000.0034]:[0]:[0.0.0.0]/216
                      40.0.0.2                           100          0 i
```

- **show l2route evpn mac all**

```
leaf3# show l2route evpn mac all
Topology    Mac Address    Prod   Next Hop (s)
----------- -------------- ------ --------------
101         0000.8816.b645 BGP    40.0.0.2
101         0001.0000.0033 Local  Ifindex 4362086
101         0001.0000.0035 Local  Ifindex 4362086
101         0011.0000.0034 BGP    40.0.0.2
```

- **show l2route evpn mac-ip all**

```
leaf3# show l2route evpn mac-ip all
Topology ID Mac Address    Prod Host IP                 Next Hop (s)
----------- -------------- ---- ------------------------------------------------------
101         0011.0000.0034 BGP  5.1.3.2                      40.0.0.2
102         0011.0000.0034 BGP  5.1.3.2                      40.0.0.2
```

Configuring ND Suppression

ND Suppression on the Overlay

Multicast Neighbor Solicitation packets from host to another host are flooded over the BGP/EVPN VXLAN Core when hosts are behind two different VXLAN peers.

The ND Suppression cache is built by:

- Snooping NS request in the hosts and populating the ND Suppression cache with source IP and MAC bindings in the request.
- Learning IPv6-Host or MAC address information through BGP EVPN MAC route advertisements.

With ND Suppression, for host to host communication behind two different VXLAN peers, if the remote host is not learned in the suppression cache initially, then NS packets are flooded over the BGP/EVPN VXLAN Core. However, once the ND Suppression cache on a switch S1 is populated with the remote host, any subsequent Neighbor Solicitation request packet for the remote host in the hosts behind S1 are proxied by the Switch S1 thereby preventing the flooding of Neighbor Solicitation packet over the BGP-EVPN/VXLAN core

For ND Suppression cache scale values, see *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*.

Guidelines and Limitations for ND Suppression

ND suppression has the following configuration guidelines and limitations:

- Beginning with Cisco NX-OS Release 10.3(1)F, the Cisco Nexus 9300-X Cloud Scale switches supports the ND Suppression feature only on plain BGP EVPN.
- ND Suppression is not supported with BGP-EVPN feature variants like Multisite, Virtual MCT, IRB, Centralized Gateway, Firewall Clustering, vPC.
- For link-local addresses of hosts, ND Suppression is not supported and instead multicast NS for link local address of hosts are flooded over the core of BGP EVPN VXLAN network.
- ND Suppression gets enabled on all VNIs on which suppress-arp is enabled.
- ND Suppression CLI knob must be enabled only under the following conditions:
    - The suppress-arp must be enabled on a VNI and there must be an SVI associated with this VNI/VLAN. Also, this SVI must be in up state and must have both IPv4 and IPv6 address enabled.
    - ND Suppression will not work in the following conditions:
        - If SVI not present for the VLAN/VNI on which suppress-arp/suppress nd is enabled.
        - If SVI associated with VLAN VNI on which suppress-arp/suppress nd is enabled is down.
        - If SVI associated with VLAN/VNI on which suppress-arp/suppress nd is enabled has only IPv4 and no IPv6 address.
        - If SVI associated with VLAN/VNI on which suppress-arp/suppress nd is enabled has only IPv6 and no IPv4 address.

In all the above conditions, host to host traffic can potentially be dropped.

- For ND Suppression VACL to work, increase the SUP TCAM size to 768 or above using the **hardware access-list tcam region sup-tcam 768** command.

Configuring ND Suppression

This procedure describes how to enable/disable the ND suppression feature on the NVE interface.

Before you begin

Ensure that ARP suppression is enabled.

SUMMARY STEPS

1. **configure terminal**
2. **hardware access-list tcam region ing-sup 768**
3. **copy running-config startup-config**
4. **reload**
5. **configure terminal**
6. **interface nve 1**
7. **[no]suppress nd**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**hardware access-list tcam region ing-sup 768** **Example:** `switch# hardware access-list tcam region ing-sup 768`	Carves the Ingress SUP TCAM size to 768.
Step 3	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.
Step 4	**reload** **Example:** `switch# reload`	Reloads the switch.
Step 5	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose

Step 6	**interface nve 1**
	Example:
	`switch(config)# interface nve 1`
	`switch(config-if-nve)#`
		Enters interface nve configuration mode.

Step 7	**[no]suppress nd** **Example:** `switch(config-if-nve)# suppress nd`	Configures ND Suppression for all ARP enabled VNIs. Option **no** disables the ND Suppression for all ARP enabled VNIs.

Note

- When global **suppress arp** command is configured, ND Suppression is enabled on all VNIs.
- When global **suppress arp** command is not configured and instead per VNI **suppress arp** command is configured, then ND Suppression is enabled on all VNIs on which ARP suppression is configured.
- When enabling suppress arp command on a vPC pair, ensure steps 1-4 on both peers are complete before enabling the feature.

Verifying the ND Suppression Configuration

To display the ND Suppression configuration information, enter one of the following commands:

Command	Purpose
show run nv overlay	Displays the ND suppression configuration status.
show nve vni	Displays whether the ND suppression config has been enabled for ARP enabled VNIs.
show nve internal export nve	Displays whether the ND suppression config has been enabled or not in SDB.
show nve internal export vni	Displays the ND suppression state per VNI in SDB.
show ipv6 nd suppression-cache detail command.	Displays the ICMPv6 cache entries that are present in local.
show ipv6 nd suppression-cache remote	Displays the ICMPv6 cache entries that are present in remote.
show ipv6 nd suppression-cache summary	Displays the IPv6 cache entries summary of both local and remote.
show ipv6 nd suppression-cache statistics	Displays the IPv6 ND suppression cache statistics.
show ipv6 nd suppression-cache vlan "vlan_id"	Displays the details of IPv6 ND Suppression cache entries for a particular VLAN.

The following example shows sample output for the **show run nv overlay** command:

```
switch(config-if-nve)# sh run nv overlay
!Command: show running-config nv overlay
!Running configuration last done at: Sat Mar 19 01:07:49 2022
!Time: Sat Mar 19 01:10:00 2022

version 10.2(3) Bios:version 07.68
feature nv overlay

vlan 101-110,200-203,500-501

interface nve1
  no shutdown
  host-reachability protocol bgp
  suppress nd
  global suppress-arp
```

The following example shows sample output for the **show nve vni** command:

```
switch(config-if-nve-vni)# sh nve vni
Codes: CP - Control Plane        DP - Data Plane
       UC - Unconfigured         SA - Suppress ARP
       S-ND Suppress ND
       SU - Suppress Unknown Unicast
       Xconn - Crossconnect
       MS-IR - Multisite Ingress Replication
       HYB - Hybrid IRB mode

Interface VNI      Multicast-group   State Mode Type [BD/VRF]      Flags
--------- -------- ----------------- ----- ---- ------------------ -----
nve1      5000     239.2.0.2         Up    CP   L2 [500]           SA S-ND
```

The following example shows sample output for the **show nve internal export nve** command:

```
switch(config-if-nve-vni)# sh nve internal export nve

NVE Interface information.
+-----------------------------------------------------------+
Interface: nve1, Admin State: Up,
   State: nve-intf-add-complete, Encap: vxlan
   Source interface: loopback3, VRF: default,
   Anycast-interface: <none>
   Mcast-routing src intf <none>
   Primary IP: 4.4.4.4, Secondary IP: 0.0.0.0,
   VNI-VRF: default, Allow-Src-Lpbk-Down: No,
   Advertise MAC route: No,
   Virtual-rMAC: 0000.0000.0000,
   Mcast-routing Primary IP: 0.0.0.0
   Suppress ND: 1
   Host-reachability: CP
   unknown-peer-forwarding-mode: disable
   VNI assignment mode: n/a
   Multisite bgw-if: <none> (ip: 0.0.0.0, admin/oper state: Down/Down)
    src-node-last-notify: None
    anycast-node-last-notify: None
    mcast-src-node-last-notify: None
    multi-src-node-last-notify: None

+-----------------------------------------------------------+
```

The following example shows sample output for the **show nve internal export vni** command:

```
switch(config-if-nve-vni)# sh nve internal export vni

NVE VNI Information.
```

```
+-----------------------------------------------------------+
 VNI: 5000 [500] Mgroup: 239.2.0.2 Provision-State: vni-add-complete
  Primary: 4.4.4.4 Secondary: 0.0.0.0 SRC-VRF: default
  Encap: vxlan Repl-mode: Mcast
  Suppress ARP: SP Suppress ND: Enabled Mode: CP, VNI-VRF: <FALSE>  [vrf-id 0] [vrf flags
0x0]
  Suppress Unknown-Unicast: FALSE
  X-connect : Disabled
  [VNI local configs] SA : TRUE, Mcast-group : TRUE, IR proto BGP: FALSE
  Config Src: CLI, VNI flags: 0x0
  Spine-AGW: Disabled, HYBRID: Disabled
  Multisite optimized IR: Disabled
  Multisite DCI Group Unknown Address

+-----------------------------------------------------------+
```

The following example shows sample output for the **show ipv6 nd suppression-cache detail** command:

```
switch(config)# show ipv6 nd suppression-cache detail

Flags: + - Adjacencies synced via CFSoE
       L - Local Adjacency
       R - Remote Adjacency
       L2 - Learnt over L2 interface
       PS - Added via L2RIB, Peer Sync
       RO - Dervied from L2RIB Peer Sync Entry

IPv6 Address      Age      Mac Address    Vlan Physical-ifindex    Flags    Remote Vtep
Addrs

172:11:1:1::51  00:00:18 acf2.c5f6.7641   11 Ethernet1/51        L
172:11:1:1::201 00:06:14 0000.0011.1111   11 (null)              R        30.100.1.1
172:11:1:1::101 00:06:14 74a0.2f1d.d481   11 (null)              R        10.10.11.11
```

The following example shows sample output for the **show ipv6 nd suppression-cache local** command:

```
switch(config)# show ipv6 nd suppression-cache local

Flags: + - Adjacencies synced via CFSoE
       L - Local Adjacency
       R - Remote Adjacency
       L2 - Learnt over L2 interface

Ip Address      Age      Mac Address    Vlan Physical-ifindex    Flags

172:11:1:1::51  00:00:23 acf2.c5f6.7641   11 Ethernet1/51        L
```

The following example shows sample output for the **show ipv6 nd suppression-cache remote** command:

```
switch(config)# show ipv6 nd suppression-cache remote

Flags: + - Adjacencies synced via CFSoE
       L - Local Adjacency
       R - Remote Adjacency
       L2 - Learnt over L2 interface
       PS - Added via L2RIB, Peer Sync
       RO - Dervied from L2RIB Peer Sync Entry

IPv6 Address      Age      Mac Address    Vlan Physical-ifindex    Flags    Remote Vtep
Addrs

172:11:1:1::201 00:06:24 0000.0011.1111   11 (null)              R        30.100.1.1
172:11:1:1::101 00:06:24 74a0.2f1d.d481   11 (null)              R        10.10.11.11
```

The following example shows sample output for the **show ipv6 nd suppression-cache statistics** command:

```
switch(config)# show ipv6 nd suppression-cache statistics

ND packet statistics for suppression-cache

Suppressed:

Total: 1
L3 mode :       Requests 1, Replies 1
                Flood ND Probe 0

Received:
Total: 1
 L3 mode:       NS 1, Non-local NA 0
                Non-local NS 0

Mobility Requests:
Total: 0
 L3 mode:       Remote-to-local 0, Local-to-remote 0
                Remote-to-remote 0

RARP Signal Refresh: 0

ND suppression-cache Local entry statistics
Adds 3, Deletes 0
```

The following example shows sample output for the **show ipv6 nd suppression-cache summary** command:

```
switch(config)# show ipv6 nd suppression-cache summary

IPV6 ND suppression-cache Summary
Remote              :2
Local               :1
Total               :3
```

The following example shows sample output for the **show ipv6 nd suppression-cache vlan "vlan_id"** command:

```
switch(config)# show ipv6 nd suppression-cache vlan 11

Flags: + - Adjacencies synced via CFSoE
       L - Local Adjacency
       R - Remote Adjacency
       L2 - Learnt over L2 interface
       PS - Added via L2RIB, Peer Sync
       RO - Dervied from L2RIB Peer Sync Entry

IPv6 Address      Age      Mac Address    Vlan Physical-ifindex    Flags    Remote Vtep
Addrs

172:11:1:1::51  00:00:40 acf2.c5f6.7641   11 Ethernet1/51        L
172:11:1:1::201 00:06:36 0000.0011.1111   11 (null)              R        30.100.1.1
172:11:1:1::101 00:06:36 74a0.2f1d.d481   11 (null)              R        10.10.11.11
```