



## Configuring VXLAN BGP EVPN

- [Information About VXLAN BGP EVPN, on page 1](#)
- [Guidelines and Limitations for VXLAN BGP EVPN, on page 2](#)
- [Considerations for VXLAN BGP EVPN deployment, on page 2](#)
- [Configuring VXLAN BGP EVPN, on page 5](#)
- [Examples of VXLAN BGP EVPN \(EBGP\), on page 23](#)

### Information About VXLAN BGP EVPN

VXLAN is a MAC in IP/UDP overlay that allows layer 2 segments to be stretched across an IP core. All the benefits of layer 3 topologies are thereby available with VXLAN. The encapsulation and decapsulation of VXLAN headers is handled by a functionality embedded in VXLAN Tunnel End Points (VTEPs). VTEPs themselves could be implemented in software or a hardware form-factor.

VXLAN natively operates on a flood-n-learn mechanism where BU (Broadcast, Unknown Unicast) traffic in a given VXLAN network is sent over the IP core to every VTEP that has membership in that network. IP multicast is used to send traffic over the network. The receiving VTEPs decapsulate the packet, and based on the inner frame perform layer-2 MAC learning. The inner SMAC is learnt against the outer Source IP Address (SIP) corresponding to the source VTEP. In this way, reverse traffic can be unicasted toward the previously learnt end host.

Motivations for using an overlay architecture include:

- **Scalability** — VXLAN provides Layer-2 connectivity that allows the infrastructure that can scale to 16 million tenant networks. It overcomes the 4094-segment limitation of VLANs. This is necessary to address today's multi-tenant cloud requirements.
- **Flexibility** — VXLAN allows workloads to be placed anywhere, along with the traffic separation required in a multi-tenant environment. The traffic separation is done using network segmentation (segment IDs or virtual network identifiers [VNIs]). Workloads for a tenant can be distributed across different physical devices (since workloads are added as the need arises, into available server space) but the workloads are identified by the same layer 2 or layer 3 VNI as the case may be.
- **Mobility** — VMs can be moved from one data center location to another without updating spine switch tables. This is because entities within the same tenant network in a VXLAN/EVPN fabric setup retain the same segment ID, regardless of their location.

One of the biggest limitations of VXLAN flood-n-learn is the inherent flooding that is required ensuring that learning happens at the VTEPs. In a traditional deployment, a layer-2 segment is represented with a VLAN

that comprises a broadcast domain, which also scopes BU traffic. With VXLAN, now the layer-2 segment spans a much larger boundary across an IP core where floods are translated to IP multicast (or HER). Consequently, the flood-n-learn based scheme presents serious scale challenges especially as the number of end hosts go up. This is addressed via learning using a control-plane for distribution of end host addresses. The control plane of choice is BGP EVPN.

## Guidelines and Limitations for VXLAN BGP EVPN

The following are the limitations for Virtual Extensible LAN (VXLAN) Border Gateway Protocol (BGP) Ethernet VPN (EVPN) has the following:

- Multicast over VXLAN is currently not supported.
- show commands with the keyword **internal** are not supported.
- For EBGp, it is recommended to use a single overlay EBGp EVPN session between loopbacks.
- Bind NVE to a loopback address that is separate from other loopback addresses that are required by Layer 3 protocols. A best practice is to use a dedicated loopback address for VXLAN.
- VXLAN BGP EVPN does not support an NVE interface in a non-default VRF.
- It is recommended to configure a single BGP session over the loopback for an overlay BGP session.
- The VXLAN UDP port number is used for VXLAN encapsulation. It complies with IETF standards and is not configurable.
- VXLAN BGP EVPN currently supports only leaf switch functionality. Spine switch functionality is not supported.
- Support is not available for any integrated underlay technologies such as route-reflector, or anycast rendezvous point, or Multicast Source Discovery Protocol (MSDP) rendezvous point.
- Border leaf functionality and interworking between BGP EVPN and traditional Layer 3 and Layer 2 overlay networks are not supported.
- Auto route-distinguisher and auto route-target for IP VRF is not supported
- Centralized Gateway for Layer 2 VXLAN network identifier (L2VNI) is not supported.
- BGP EVPN Network Virtualization Overlay MIB is not supported.
- In EVPN deployments, once a VLAN is used for a core-facing SVI, it should not be allowed in any trunk. For a core-facing SVI to function properly, the **no autostate** command must be configured under the SVI.

### Topic 2.1

## Considerations for VXLAN BGP EVPN deployment

The following considerations need to be taken into account for VXLAN BGP EVPN deployment:

- A loopback address is required when using the source-interface config command. The loopback address represents the local VTEP IP.
- To establish IP multicast routing in the core, IP multicast configuration, PIM configuration, and RP configuration are required.
- VTEP to VTEP unicast reachability can be configured through any IGP/BGP protocol.
- If the anycast gateway feature is enabled for a specific VNI, then the anyway gateway feature must be enabled on all VTEPs that have that VNI configured. Having the anycast gateway feature configured on only some of the VTEPs enabled for a specific VNI is not supported.
- It is a requirement when changing the primary or secondary IP address of the NVE source interfaces to shut the NVE interface before changing the IP address.
- As a best practice, the RP for the multicast group should be configured only on the spine layer. Use the anycast RP for RP load balancing and redundancy.
- Every tenant VRF needs a VRF overlay, VLAN and SVI for VXLAN routing.
- The following considerations need to be taken into account with eBGP use case:
  - Manual configuration of the Route Targets (RT) is required. RT must be matching between the VTEPs for a given EVPN instance (EVI).
  - The **retain route-target all** BGP knob must be enabled on the Spine nodes under BGP routing process
  - The **set ip next-hop unchanged** BGP knob must be enabled on Spine nodes to set next hop for EVPN routes to the proper VTEP node.
  - Peering between VTEPs can be achieved to multiple Spine nodes to achieve redundancy.
- Ensure the following to create a proper VLAN database:
  - The route targets with eBGP EVPN VxLAN design model cannot be auto generated like in iBGP/IGP model, hence they need to be manually configured for each EVPN instance (EVI) and should be matching for a given EVI. Failure to manually configure route target will result in loss of connectivity and improper operation due to routes not being installed.
  - To ensure proper operation of EVPN VXLAN, assign the vlan first as an access interface to create the vlan and store it in the vlan.dat file. For a trunk interface, trying to create a SVI before creating the vlan in VLAN.dat will put the SVI in a down state.
- In case of a scoped configuration, not all L2 VNIs need to be enabled on all VTEP switches. They will only be enabled as needed on a given VTEP.
- Route Distinguishers (RD) need to be unique per IP VRF (L3 VNI). Route Targets (RT) must match for a given IP VRF (L3 VNI) . There is no auto-generation neither for RD or RT for the case of IP VRF (L3 VNI).
- All VTEP switches need not be configured with same L2 VNIs unless in the scoped configuration. Access VLANs are the VLANs connected to hosts. Access SVIs must have an IP address with the same subnet as the hosts the VLAN is connected to. For AnyCast Gateway support, Access SVIs of the same VLAN should have the same IP and MAC addresses in all VTEPs.

- It is important to configure additional L3 VNIs on all VTEP nodes where Inter-VxLAN communication is needed.

## Network considerations for VXLAN deployments

The following network consideration need to be taken into account for VXLAN deployments:

### MTU Size in the Transport Network

Due to the MAC-to-UDP encapsulation, VXLAN introduces 50-byte overhead to the original frames. Therefore, the maximum transmission unit (MTU) in the transport network needs to be increased by 50 bytes. If the overlays use a 1500-byte MTU, the transport network needs to be configured to accommodate 1550-byte packets at a minimum. Jumbo-frame support in the transport network is required if the overlay applications tend to use larger frame sizes than 1500 bytes.

### ECMP and LACP Hashing Algorithms in the Transport Network

Switches introduce a level of entropy in the source UDP port for ECMP and LACP hashing in the transport network. As a way to augment this implementation, the transport network uses an ECMP or LACP hashing algorithm that takes the UDP source port as an input for hashing, which achieves the best load-sharing results for VXLAN encapsulated traffic.

### Multicast Group Scaling

The VXLAN implementation uses multicast tunnels for broadcast, unknown unicast, and multicast traffic forwarding. Ideally, one VXLAN segment mapping to one IP multicast group is the way to provide the optimal multicast forwarding. It is possible, however, to have multiple VXLAN segments share a single IP multicast group in the core network. VXLAN can support up to 16 million logical Layer 2 segments, using the 24-bit VNID field in the header. With one-to-one mapping between VXLAN segments and IP multicast groups, an increase in the number of VXLAN segments causes a parallel increase in the required multicast address space and the amount of forwarding states on the core network devices. At some point, multicast scalability in the transport network can become a concern. In this case, mapping multiple VXLAN segments to a single multicast group can help conserve multicast control plane resources on the core devices and achieve the desired VXLAN scalability. However, this mapping comes at the cost of suboptimal multicast forwarding. Packets forwarded to the multicast group for one tenant are now sent to the VTEPs of other tenants that are sharing the same multicast group. This causes inefficient utilization of multicast data plane resources. Therefore, this solution is a trade-off between control plane scalability and data plane efficiency.

Despite the suboptimal multicast replication and forwarding, having multiple-tenant VXLAN networks to share a multicast group does not bring any implications to the Layer 2 isolation between the tenant networks. After receiving an encapsulated packet from the multicast group, a VTEP checks and validates the VNID in the VXLAN header of the packet. The VTEP discards the packet if the VNID is unknown to it. Only when the VNID matches one of the VTEP's local VXLAN VNIDs, does it forward the packet to that VXLAN segment. Other tenant networks will not receive the packet. Thus, the segregation between VXLAN segments is not compromised.

## Considerations for the Transport Network

The following considerations need to be taken into account for the configuration of the transport network:

- On the VTEP device:

- Enable and configure IP multicast.
  - Create and configure a loopback interface with a /32 IP address.
  - Enable IP multicast on the loopback interface.
  - Advertise the loopback interface /32 addresses through the routing protocol (static route) that runs in the transport network.
  - Enable IP multicast on the uplink outgoing physical interface.
- Throughout the transport network:
    - Enable and configure IP multicast.

## Configuring VXLAN BGP EVPN

### Configuring Underlay Transport (Unicast and Multicast) between the VTEPs and the Spines

Follow these steps to configure underlay transport on the Spine:

#### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip pim rp-address***rp-address***group-list***prefix*
4. **ip pim rp-candidate** *loopbackif\_number***group-list***prefix*
5. **ip pim ssm range***groups*
6. **ip pim anycast-rp***anycast-rpanycast-rp*
7. **interface** *loopbacknumber*
8. **ip address***ip address*
9. **ip pim sparse-mode**
10. **exit**
11. **interface port-channel***channel-number*
12. **mtu***bytes*
13. **medium p2p**
14. **ip address***ip-address mask*
15. **ip pim sparse-mode**
16. **exit**

#### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b>	Enables privileged EXEC mode.  • Enter your password if prompted.

	Command or Action	Purpose
	Device> <b>enable</b>	
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>ip pim rp-address</b> <i>rp-address</i> <b>group-list</b> <i>prefix</i> <b>Example:</b> Device(config)# <b>ip pim rp-address 100.1.1.1 group-list 239.0.0.0/8</b>	Configures a PIM static route processor (RP) address for a multicast group range and specifies a group range for a static RP.
<b>Step 4</b>	<b>ip pim rp-candidate</b> <i>loopbackif_number</i> <b>group-list</b> <i>prefix</i> <b>Example:</b> Device(config)# <b>ip pim rp-candidate loopback1 group-list 239.0.0.0/8</b>	Configures a PIM address as a RP candidate. Specifies the loopback interface. Specifies a group range handled by the RP.
<b>Step 5</b>	<b>ip pim ssm range<i>groups</i> <b>Example:</b> Device(config)# <b>ip pim ssm range 232.0.0.0/8</b></b>	Configures a group range for SSM.
<b>Step 6</b>	<b>ip pim anycast-rp</b> <i>anycast-rpanycast-rp</i> <b>Example:</b> Device(config)# <b>ip pim anycast-rp 100.1.1.1 10.1.1.1</b>	Configures PIM Anycast-RP peer for the specified Anycast-RP address.
<b>Step 7</b>	<b>interface</b> <i>loopbacknumber</i> <b>Example:</b> Device(config)# <b>interface loopback0</b>	Creates a loopback interface and enters interface configuration mode.
<b>Step 8</b>	<b>ip address</b> <i>ip address</i> <b>Example:</b> Device(config-if)# <b>ip address 10.1.1.1/32</b>	Defines the IP address for an interface.
<b>Step 9</b>	<b>ip pim sparse-mode</b> <b>Example:</b> Device(config-if)# <b>ip pim sparse-mode</b>	Enables Protocol Independent Multicast (PIM) sparse mode on an interface.
<b>Step 10</b>	<b>exit</b> <b>Example:</b> Device(config-if)# <b>exit</b>	Exits the interface configuration mode
<b>Step 11</b>	<b>interface port-channel</b> <i>channel-number</i> <b>Example:</b> Device(config)# <b>interface port-channel1</b>	Specifies the port-channel interface to configure, and enters the interface configuration mode.

	Command or Action	Purpose
Step 12	<b>mtu</b> <i>bytes</i> <b>Example:</b> Device(config-if) # <b>mtu 9198</b>	Sets the interface MTU size.
Step 13	<b>medium</b> <i>p2p</i> <b>Example:</b> Device(config-if) # <b>medium p2p</b>	Configures the interface medium as point to point
Step 14	<b>ip address</b> <i>ip-address mask</i> <b>Example:</b> Device(config-if) # <b>ip address 10.10.1.1/30</b>	Defines the IP address for an interface.
Step 15	<b>ip pim sparse-mode</b> <b>Example:</b> Device(config-if) # <b>ip pim sparse-mode</b>	Enables Protocol Independent Multicast (PIM) sparse mode on an interface.
Step 16	<b>exit</b> <b>Example:</b> Device(config-if) # <b>exit</b>	Exits the interface configuration mode

## Configuring the VTEP

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing**
4. **ip pim rp-address***rp-address*
5. **ip routing**
6. **interface loopback***number*
7. **ip address***ip address*
8. **ip pim sparse-mode**
9. **exit**
10. **interface loopback***number*
11. **ip vrf forwarding***vrf name*
12. **ip address***ip address*
13. **exit**
14. **interface tengigabitethernet***slot/port*
15. **no switchport**
16. **no ip address**
17. **channel-group***number*
18. **exit**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. <ul style="list-style-type: none"><li>• Enter your password if prompted.</li></ul>
Step 2	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
Step 3	<b>ip multicast-routing</b> <b>Example:</b> Device(config)# <b>ip multicast-routing</b>	Enables IP multicast routing.
Step 4	<b>ip pim rp-address</b> <i>rp-address</i> <b>Example:</b> Device(config)# <b>ip pim rp-address 100.1.1.1</b>	Configures a PIM static route processor (RP) address for a multicast group range. The rp address used in this step should be the same one used on the spine.
Step 5	<b>ip routing</b> <b>Example:</b> Device(config)# <b>ip routing</b>	Enables routing on the switch. Even if IP routing was previously enabled, this step ensures that it is activated.
Step 6	<b>interface</b> <i>loopbacknumber</i> <b>Example:</b> Device(config)# <b>interface Loopback0</b>	Creates a loopback interface and enters interface configuration mode. This loopback interface is assigned to the NVE interface.
Step 7	<b>ip address</b> <i>ip address</i> <b>Example:</b> Device(config-if)# <b>ip address 10.11.11.11 255.255.255.255</b>	Defines the IP address for an interface.
Step 8	<b>ip pim sparse-mode</b> <b>Example:</b> Device(config-if)# <b>ip pim sparse-mode</b>	Enables Protocol Independent Multicast (PIM) sparse mode on an interface.
Step 9	<b>exit</b> <b>Example:</b> Device(config-if)# <b>exit</b>	Exits the interface configuration mode
Step 10	<b>interface</b> <i>loopbacknumber</i> <b>Example:</b> Device(config)# <b>interface Loopback2</b>	Creates a loopback interface and enters interface configuration mode. This loopback interface is assigned to the L3 VNI.
Step 11	<b>ip vrf forwarding</b> <i>vrf name</i> <b>Example:</b>	Associates the VRF with the Layer 3 interface.



	Command or Action	Purpose
	Device(config-if) # <b>vrf forwarding tenant_1</b>	
<b>Step 12</b>	<b>ip address</b> <i>ip address</i> <b>Example:</b> Device(config-if) # <b>ip address 11.11.11.11 255.255.255.255</b>	Defines the IP address for an interface.
<b>Step 13</b>	<b>exit</b> <b>Example:</b> Device(config-if) # <b>exit</b>	Exits the interface configuration mode
<b>Step 14</b>	<b>interface</b> <i>tengigabitethernet</i> <i>slot/port</i> <b>Example:</b> Device(config) # <b>interface TenGigabitEthernet1/1/2</b>	Selects the port to configure.
<b>Step 15</b>	<b>no switchport</b> <b>Example:</b> Device(config-if) # <b>no switchport</b>	Makes the interface Layer 3 capable.
<b>Step 16</b>	<b>no ip address</b> <b>Example:</b> Device(config-if) # <b>no ip address</b>	Disables IP processing on a particular interface.
<b>Step 17</b>	<b>channel-group</b> <i>number</i> <b>Example:</b> Device(config-if) # <b>channel-group 1 mode active</b>	Assigns and configure a physical interface to an EtherChannel.
<b>Step 18</b>	<b>exit</b> <b>Example:</b> Device(config-if) # <b>exit</b>	Exits the interface configuration mode

## Configuring eBGP on the Spine:

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip prefix-list** *name* [*seqnumber*] { **permit** | **deny** } *prefix* [*eqlength*] | [*ge* *length*] | [*le* *length*]
4. **route-map** *name* { **permit** | **deny** } [*sequence number*]
5. **set ip next-hop unchanged**
6. **exit**
7. **route-map** *name* { **permit** | **deny** } [*sequence number*]
8. **match ip address prefix-list** *name* [*name*]
9. **exit**

10. **router bgp***number*
11. **router id** { *router id* }
12. **bgp log-neighbor-changes**
13. **address-family ipv4 unicast**
14. **redistribute direct** [ *route-map**map-name* ]
15. **exit**
16. **address-family l2vpn evpn**
17. **nexthop route-map***name*
18. **retain route-target all**
19. **exit**
20. **neighbor***vtep1 loopback address***remote-as***number*
21. **neighbor***ip-address***update-source***interface-type interface-number*
22. **neighbor** { *ip address* | *peer-group-name* } **ebgp-multihop** [ *ttl* ]
23. **address-family ipv4 unicast**
24. **neighbor** { *ip address* | *peer-group-name* } **send-community both**
25. **soft-reconfiguration inbound**
26. **exit**
27. **address-family l2vpn evpn**
28. **neighbor** { *ip address* | *peer-group-name* } **send-community both**
29. **neighbor** { *ip address* | *peer-group-name* } **route-map***map-name* { **in** | **out** }
30. **exit**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode.  • Enter your password if prompted.
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>ip prefix-list</b> [ <i>sequence number</i> ] { <b>permit</b>   <b>deny</b> } <i>prefix</i> [ <i>length</i> ]   [ <i>length</i> ]   [ <i>length</i> ]  <b>Example:</b> Device (config)# <b>ip prefix-list lo_prefix seq 5 permit 0.0.0.0/0 le 32</b>	Creates a prefix list to match IP packets or routes against.
<b>Step 4</b>	<b>route-map name</b> { <b>permit</b>   <b>deny</b> } [ <i>sequence number</i> ]  <b>Example:</b> Device (config)# <b>route-map NH-UNCHANGED permit 10</b>	Creates the route map entry. Enters route-map configuration mode.
<b>Step 5</b>	<b>set ip next-hop unchanged</b>  <b>Example:</b>	Defines the route-map and applies outbound policy for neighbour.

	Command or Action	Purpose
	Device(config-route-map)# <b>set ip next-hop unchanged</b>	
<b>Step 6</b>	<b>exit</b> <b>Example:</b> Device(config-route-map)# <b>exit</b>	Exits the route-map configuration mode
<b>Step 7</b>	<b>route-map name {permit   deny} [sequence number]</b> <b>Example:</b> Device(config)# <b>route-map any_prefix permit 10</b>	Creates the route map entry. Enters route-map configuration mode.
<b>Step 8</b>	<b>match ip address prefix-list name [name]</b> <b>Example:</b> Device(config-route-map)# <b>match ip address prefix-list lo_prefix</b>	Matches against one or more ip address prefix lists.
<b>Step 9</b>	<b>exit</b> <b>Example:</b> Device(config-route-map)# <b>exit</b>	Exits the route-map configuration mode
<b>Step 10</b>	<b>router bgp number</b> <b>Example:</b> Device(config)# <b>router bgp 1</b>	Configures BGP.
<b>Step 11</b>	<b>router id {router id}</b> <b>Example:</b> Device(config-router)# <b>router-id 10.1.1.1</b>	Specifies a fixed router ID in the router configuration mode.
<b>Step 12</b>	<b>bgp log-neighbor-changes</b> <b>Example:</b> Device(config-router)# <b>log-neighbor-changes</b>	Enables the generation of logging messages generated when the status of a BGP neighbor changes.
<b>Step 13</b>	<b>address-family ipv4 unicast</b> <b>Example:</b> Device(config-router)# <b>address-family ipv4 unicast</b>	Enters address family configuration mode and Specifies IP Version 4 unicast address prefixes.
<b>Step 14</b>	<b>redistribute direct [route-map map-name]</b> <b>Example:</b> Device(config-router-af)# <b>redistribute direct route-map any_prefix</b>	Distributes routes that are directly connected on an interface.
<b>Step 15</b>	<b>exit</b> <b>Example:</b> Device(config-router-af)# <b>exit</b>	Exits the address family configuration mode

	Command or Action	Purpose
Step 16	<b>address-family l2vpn evpn</b> <b>Example:</b> Device(config-router)# <b>address-family l2vpn evpn</b>	Specifies the L2VPN address family and enters address family configuration mode. The evpn keyword specifies that EVPN endpoint provisioning information is to be distributed to BGP peers.
Step 17	<b>nexthop route-mapname</b> <b>Example:</b> Device(config-router-af)# <b>nexthop route-map NH-UNCHANGED</b>	Specifies that Border Gateway Protocol (BGP) routes are resolved using only the next hops that have routes that match specific characteristics.
Step 18	<b>retain route-target all</b> <b>Example:</b> Device(config-router-af)# <b>retain route-target all</b>	Accepts received updates with specified route targets.
Step 19	<b>exit</b> <b>Example:</b> Device(config-router-af)# <b>exit</b>	Exits the address family configuration mode
Step 20	<b>neighborvtep1 loopback addressremote-asnumber</b> <b>Example:</b> Device(config-router)# <b>neighbor 10.11.11.11 remote-as 2</b>	Adds an entry to the BGP or multiprotocol BGP neighbor table in the router configuration mode.
Step 21	<b>neighborip-addressupdate-sourceinterface-type interface-number</b> <b>Example:</b> Device(config-router)# <b>neighbor 10.11.11.11 update-source loopback0</b>	Allows BGP sessions to use any operational interface for TCP connections.
Step 22	<b>neighbor { ip address   peer-group-name } ebgp-multihop [ ttl ]</b> <b>Example:</b> Device(config-router)# <b>neighbor 10.11.11.11 ebgp-multihop 10</b>	Allows BGP connections to external peers on networks that are not directly connected.
Step 23	<b>address-family ipv4 unicast</b> <b>Example:</b> Device(config-router)# <b>address-family ipv4 unicast</b>	Enters address family configuration mode and Specifies IP Version 4 unicast address prefixes.
Step 24	<b>neighbor { ip address   peer-group-name } send-community both</b> <b>Example:</b> Device(config-router-af)# <b>neighbor 10.11.11.11 send-community both</b>	Specifies both standard and extended communities attribute should be sent to a BGP neighbour.

	Command or Action	Purpose
Step 25	<b>soft-reconfiguration inbound</b> <b>Example:</b> Device(config-router-af)# <b>soft-reconfiguration inbound</b>	Configures the switch software to start storing BGP peer updates.
Step 26	<b>exit</b> <b>Example:</b> Device(config-router-af)# <b>exit</b>	Exits the address family configuration mode
Step 27	<b>address-family l2vpn evpn</b> <b>Example:</b> Device(config-router)# <b>address-family l2vpn evpn</b>	Specifies the L2VPN address family and enters address family configuration mode. The evpn keyword specifies that EVPN endpoint provisioning information is to be distributed to BGP peers.
Step 28	<b>neighbor { ip address   peer-group-name } send-community both</b> <b>Example:</b> Device(config-router-af)# <b>neighbor 10.11.11.11 send-community both</b>	Specifies both standard and extended communities attribute should be sent to a BGP neighbour.
Step 29	<b>neighbor { ip address   peer-group-name } route-map map-name { in   out }</b> <b>Example:</b> Device(config-router-af)# <b>neighbor 10.11.11.11 route-map NH-UNCHANGED out</b>	Applies the inbound route map to routes received from the specified neighbor, or applies an outbound route map to routes advertised to the specified neighbor.
Step 30	<b>exit</b> <b>Example:</b> Device(config-router-af)# <b>exit</b>	Exits the address family configuration mode

## Configuring eBGP on the VTEP

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgpnumber**
4. **bgp router-id interface loopback address**
5. **bgp log-neighbor-changes**
6. **bgp graceful-restart**
7. **neighbor spine 1 loopback address remote-as number**
8. **neighbor { ip address | peer-group-name } ebgp-multihop [ ttl ]**
9. **neighbor { ip address | group-name } update-source interface**
10. **address-family ipv4**
11. **redistribute connected**

12. `neighbor ip-address activate`
13. `exit`
14. `address-family l2vpn evpn`
15. `neighbor ip-address activate`
16. `neighbor ip-address send-community both`
17. `maximum-paths number-of-paths`
18. `exit`
19. `address-family ipv4 vrf vrf-name`
20. `advertise l2vpn evpn`
21. `redistribute connected`
22. `exit`

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code> <b>Example:</b> Device> <code>enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"><li>• Enter your password if prompted.</li></ul>
Step 2	<code>configure terminal</code> <b>Example:</b> Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	<code>router bgp number</code> <b>Example:</b> Device(config)# <code>router bgp 2</code>	Configures BGP.
Step 4	<code>bgp router-id interface loopback address</code> <b>Example:</b> Device(config-router)# <code>bgp router-id interface Loopback0</code>	Specifies loopback address as router address.
Step 5	<code>bgp log-neighbor-changes</code> <b>Example:</b> Device(config-router)# <code>bgp log-neighbor-changes</code>	Enables the generation of logging messages generated when the status of a BGP neighbor changes.
Step 6	<code>bgp graceful-restart</code> <b>Example:</b> Device(config-router)# <code>bgp graceful-restart</code>	Enables the BGP graceful restart capability for a BGP neighbor.
Step 7	<code>neighbors spine 1 loopback address remote-as number</code> <b>Example:</b> Device(config-router)# <code>neighbor 10.1.1.1 remote-as 1</code>	Defines MP-BGP neighbors. Under each neighbor define l2vpn evpn.

	Command or Action	Purpose
Step 8	<b>neighbor</b> { <i>ip address</i>   <i>peer-group-name</i> } <b>ebgp-multihop</b> [ <i>ttl</i> ]  <b>Example:</b> Device(config-router)# <b>neighbor 10.1.1.1 ebgp-multihop 10</b>	Allows BGP connections to external peers on networks that are not directly connected.
Step 9	<b>neighbor</b> { <i>ip address</i>   <i>group-name</i> } <b>update-source</b> <i>interface</i>  <b>Example:</b> Device(config-router)# <b>neighbor 10.1.1.1 update-source Loopback0</b>	Configures update source. Update source can be configured per neighbor or per peer-group
Step 10	<b>address-family ipv4</b>  <b>Example:</b> Device(config-router)# <b>address-family ipv4</b>	Enters address family configuration mode.
Step 11	<b>redistribute connected</b>  <b>Example:</b> Device(config-router-af)# <b>redistribute connected</b>	Redistributes connected routes from another routing protocol.
Step 12	<b>neighbor ip-address activate</b>  <b>Example:</b> Device(config-router-af)# <b>neighbor 10.1.1.1 activate</b>	Enables the exchange information from a bgp neighbor
Step 13	<b>exit</b>  <b>Example:</b> Device(config-router-af)# <b>exit-address-family</b>	Exits the address family configuration mode
Step 14	<b>address-family l2vpn evpn</b>  <b>Example:</b> Device(config-router)# <b>address-family l2vpn evpn</b>	Specifies the L2VPN address family and enters address family configuration mode.
Step 15	<b>neighbor ip-address activate</b>  <b>Example:</b> Device(config-router-af)# <b>neighbor 10.1.1.1 activate</b>	Enables the exchange information from a bgp neighbor
Step 16	<b>neighbor ip-address send-community both</b>  <b>Example:</b> Device(config-router-af)# <b>neighbor 10.1.1.1 send-community both</b>	Specifies the communities attribute sent to a bgp neighbor
Step 17	<b>maximum-paths number-of-paths</b>  <b>Example:</b>	Controls the maximum number of parallel routes an IP routing protocol can support.

	Command or Action	Purpose
	Device(config-router-af)# <b>maximum-paths 2</b>	
<b>Step 18</b>	<b>exit</b> <b>Example:</b> Device(config-router-af)# <b>exit-address-family</b>	Exits the address family configuration mode
<b>Step 19</b>	<b>address-family ipv4 vrfvrf-name</b> <b>Example:</b> Device(config-router)# <b>address-family ipv4 vrf tenant_1</b>	Specifies the name of the VRF instance to associate with subsequent address family configuration mode commands.
<b>Step 20</b>	<b>advertise l2vpn evpn</b> <b>Example:</b> Device(config-router-af)# <b>advertise l2vpn evpn</b>	Advertises (L2VPN) EVPN routes within a tenant VRF in a VXLAN EVPN fabric.
<b>Step 21</b>	<b>redistribute connected</b> <b>Example:</b> Device(config-router-af)# <b>redistribute connected</b>	Redistributes connected routes from another routing protocol.
<b>Step 22</b>	<b>exit</b> <b>Example:</b> Device(config-router-af)# <b>exit-address-family</b>	Exits the address family configuration mode

## Configuring the NVE Interface and VNIs

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interfacenve-interface**
4. **no ip address**
5. **source-interface loopbacknumber**
6. **host-reachability protocol bgp**
7. **member vni vniassociate-vrf**
8. **member vni vnimcast-groupaddress**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>



	Command or Action	Purpose
Step 2	<b>configure terminal</b> <b>Example:</b> Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	<b>interface</b> <i>nve-interface</i> <b>Example:</b> Device(config)# <code>interface nve1</code>	Configures the NVE interface.
Step 4	<b>no ip address</b> <b>Example:</b> Device(config-if)# <code>no ip address</code>	Disables IP processing on the interface.
Step 5	<b>source-interface</b> <i>loopbacknumber</i> <b>Example:</b> Device(config-if)# <code>source-interface Loopback1</code>	Creates a loopback interface. <b>Note</b> This interface will be a different loopback from the loopback interface used for underlay.
Step 6	<b>host-reachability protocol bgp</b> <b>Example:</b> Device(config-if)# <code>host-reachability protocol bgp</code>	Defines BGP as the mechanism for host reachability advertisement.
Step 7	<b>member vni</b> <i>vni</i> <b>associate-vrf</b> <b>Example:</b> Device(config-if)# <code>member vni 11001 mcast-group 239.0.1.1</code>	Adds Layer-3 VNIs, one per tenant VRF, to the overlay. <b>Note</b> Required for VXLAN routing only.
Step 8	<b>member vni</b> <i>vni</i> <b>mcast-group</b> <i>address</i> <b>Example:</b> Device(config-if)# <code>member vni 900001 vrf tenant_1</code>	Adds Layer 2 VNIs to the tunnel interface and assigns a multicast group to the VNIs.

## Configuring L2VPN EVPN on all VTEPs

### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `l2vpn evpn`
4. `replication-type static`
5. `router-id loopbacknumber`
6. `exit`
7. `l2vpn evpn instanceinstance-numbervlan-based`
8. `encapsulation vxlan`
9. `route-target exportroute-target-id`
10. `route-target importroute-target-id`

11. `no auto-route-target`
12. `exit`
13. `vlan configuration` *vlan-id*
14. `member evpn-instance` *evpn-instance-number* `vni` *vni-number*

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code> <b>Example:</b> Device> <code>enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"><li>• Enter your password if prompted.</li></ul>
Step 2	<code>configure terminal</code> <b>Example:</b> Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	<code>l2vpn evpn</code> <b>Example:</b> Device (config) # <code>l2vpn evpn</code>	Enters L2VPN configuration mode
Step 4	<code>replication-type static</code> <b>Example:</b> Device (config-l2vpn) # <code>replication-type static</code>	Suppresses use of Inclusive Multicast Ethernet Tag (IMET) routes. IP Multicast is used for BUM traffic.
Step 5	<code>router-id loopback</code> <i>number</i> <b>Example:</b> Device (config-l2vpn) # <code>router-id Loopback1</code>	Specifies the interface that will supply the IP addresses to be used in auto-generating route distinguishers.
Step 6	<code>exit</code> <b>Example:</b> Device (config-l2vpn) # <code>exit</code>	Exits the L2VPN configuration.
Step 7	<code>l2vpn evpn instance</code> <i>instance-number</i> <code>vlan-based</code> <b>Example:</b> Device (config) # <code>l2vpn evpn instance 1 vlan-based</code>	Configures VLAN based EVI in the L2VPN configuration mode.  This command is optional if the route targets or the route distinguishers are not needed to be configured manually.
Step 8	<code>encapsulation vxlan</code> <b>Example:</b> Device (config-l2vpn) # <code>encapsulation vxlan</code>	Defines the encapsulation format as VXLAN
Step 9	<code>route-target export</code> <i>route-target-id</i> <b>Example:</b> Device (config-l2vpn) # <code>route-target export 2:1</code>	Configures BGP route exchange.

	Command or Action	Purpose
Step 10	<b>route-target import</b> <i>route-target-id</i> <b>Example:</b> Device(config-l2vpn)# <b>route-target import 2:1</b>	Configures BGP route exchange.
Step 11	<b>no auto-route-target</b> <b>Example:</b> Device(config-l2vpn)# <b>no auto-route-target</b>	Removes the automatically generated route-targets.
Step 12	<b>exit</b> <b>Example:</b> Device(config-l2vpn)# <b>exit</b>	Exits the L2VPN configuration.
Step 13	<b>vlan configuration</b> <i>vlan-id</i> <b>Example:</b> Device(config)# <b>vlan configuration 11</b>	Enters the vlan feature configuration mode.
Step 14	<b>member</b> <b>evpn-instance</b> <i>evpn-instance-number</i> <b>vni</b> <i>vni-number</i> <b>Example:</b> Device(config-vlan)# <b>member evpn-instance 1 vni 11001</b>	Configures the evpn vxlan vni instance.

## Configuring access customer facing VLAN VTEP

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet***slot/port*
4. **switchport access vlan***vlan-id*
5. **switchport mode access**
6. **exit**
7. **interface gigabitethernet***slot/port*
8. **switchport trunk allowed vlan***vlan\_list*
9. **switchport mode trunk**

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <code>configure terminal</code>	Enters global configuration mode.
<b>Step 3</b>	<b>interface gigabitethernet</b> <i>slot/port</i> <b>Example:</b> Device(config)# <code>interface GigabitEthernet1/0/11</code>	Enters the interface configuration mode on the Gigabit Ethernet interface.
<b>Step 4</b>	<b>switchport access vlan</b> <i>vlan-id</i> <b>Example:</b> Device(config-if)# <code>switchport access vlan 11</code>	Sets the access VLAN when the interface is in access mode.
<b>Step 5</b>	<b>switchport mode access</b> <b>Example:</b> Device(config-if)# <code>switchport mode access</code>	Sets the interface as a nontrunking nontagged single-VLAN Ethernet interface.
<b>Step 6</b>	<b>exit</b> <b>Example:</b> Device(config-if)# <code>exit</code>	Exits the interface configuration mode.
<b>Step 7</b>	<b>interface gigabitethernet</b> <i>slot/port</i> <b>Example:</b> Device(config)# <code>interface TenGigabitEthernet1/1/7</code>	Enters the interface configuration mode on the Gigabit Ethernet interface.
<b>Step 8</b>	<b>switchport trunk allowed vlan</b> <i>vlan_list</i> <b>Example:</b> Device(config-if)# <code>switchport trunk allowed vlan 11-210,901-905</code>	Configures the VLAN ids of the allowed VLANs for the interface.
<b>Step 9</b>	<b>switchport mode trunk</b> <b>Example:</b> Device(config-if)# <code>switchport mode trunk</code>	Sets the interface as an Ethernet trunk port.

## Configuring IP VRF on VTEPs for Inter-VxLAN routing

### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `vrf definition`*vrf-name*
4. `rd`*route-distinguisher*
5. `address-family ipv4`
6. `route-target export`*route-target-id*

7. **route-target import***route-target-id*
8. **route-target import***route-target-id***stitching**
9. **route-target export***route-target-id***stitching**
10. **exit-address-family**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. <ul style="list-style-type: none"><li>• Enter your password if prompted.</li></ul>
Step 2	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
Step 3	<b>vrf definition</b> <i>vrf-name</i> <b>Example:</b> Device(config)# <b>vrf definition tenant_1</b>	Configures a virtual routing and forwarding (VRF) routing-table instance and enters VRF configuration mode.
Step 4	<b>rd</b> <i>route-distinguisher</i> <b>Example:</b> Device(config-vrf)# <b>rd 1:1</b>	Creates routing and forwarding tables for a VRF.
Step 5	<b>address-family ipv4</b> <b>Example:</b> Device(config-vrf)# <b>address-family ipv4</b>	Enters address family configuration mode.
Step 6	<b>route-target export</b> <i>route-target-id</i> <b>Example:</b> Device(config-vrf-af)# <b>route-target export 1:1</b>	Creates a list of export RTs for the VRF with the same parameters.
Step 7	<b>route-target import</b> <i>route-target-id</i> <b>Example:</b> Device(config-vrf-af)# <b>route-target import 1:1</b>	Creates a list of import RTs for the VRF with the same parameters.
Step 8	<b>route-target import</b> <i>route-target-id</i> <b>stitching</b> <b>Example:</b> Device(config-vrf-af)# <b>route-target import 1:1 stitching</b>	Configures importing of routes from the EVPN BGP that have the matching route-target value.
Step 9	<b>route-target export</b> <i>route-target-id</i> <b>stitching</b> <b>Example:</b> Device(config-vrf-af)# <b>route-target export 1:1 stitching</b>	Configures exporting of routes from the VRF to the EVPN BGP and assigns the specified route-target identifiers to the BGP EVPN.

	Command or Action	Purpose
<b>Step 10</b>	<b>exit-address-family</b> <b>Example:</b> Device (config-vrf-af) # <b>exit-address-family</b>	Exits address-family configuration mode.

## Verifying the VXLAN BGP EVPN Configuration

Command	Purpose
<b>show nve vni</b>	Displays VNIs associated in the NVE.
<b>show ip mroute</b>	Displays multicast routing table information.
<b>show ip mfib</b>	Displays forwarding entries and interfaces in the IPv4 Multicast Forwarding Information Base (MFIB).
<b>show ip pim neighbors</b>	Displays PIM neighbour table.
<b>show ip pim tunnel</b>	Displays information about the PIM register encapsulation and decapsulation tunnels on an interface.
<b>show ip pim rp</b>	Displays mapping information for the RP.
<b>show l2vpn evpn evi [ evpn-id   all ]</b>	Displays detailed information for a particular EVI or all EVIs.
<b>show mac address-table vlan vlan id</b>	Displays information for a specific VLAN.
<b>show l2route evpn mac [ all   evivlan-id ]</b>	Displays MAC and IP address information learnt by the switch in the EVPN control plane.
<b>show bgp l2vpn evpn</b>	Displays BGP information for L2VPN-EVPN address family.
<b>show ip vrfvrf-name</b>	Displays a summary of all VRFs present on the current router and their associated route-distinguishers and interface(s).
<b>show bgp vpnv4 unicast vrfvrf-name</b>	Displays VPNv4 routes from BGP table for a specific vrf.
<b>show ip route vrfvrf-name</b>	Displays the IP routing table associated with a specific VRF.
<b>show l2vpn evpn mac</b>	Displays the MAC address database for Layer 2 EVPN.
<b>show l2vpn evpn mac ip</b>	Displays the IP address database for Layer 2 EVPN.
<b>show l2route evpn mac ip</b>	Displays MAC IP routes.

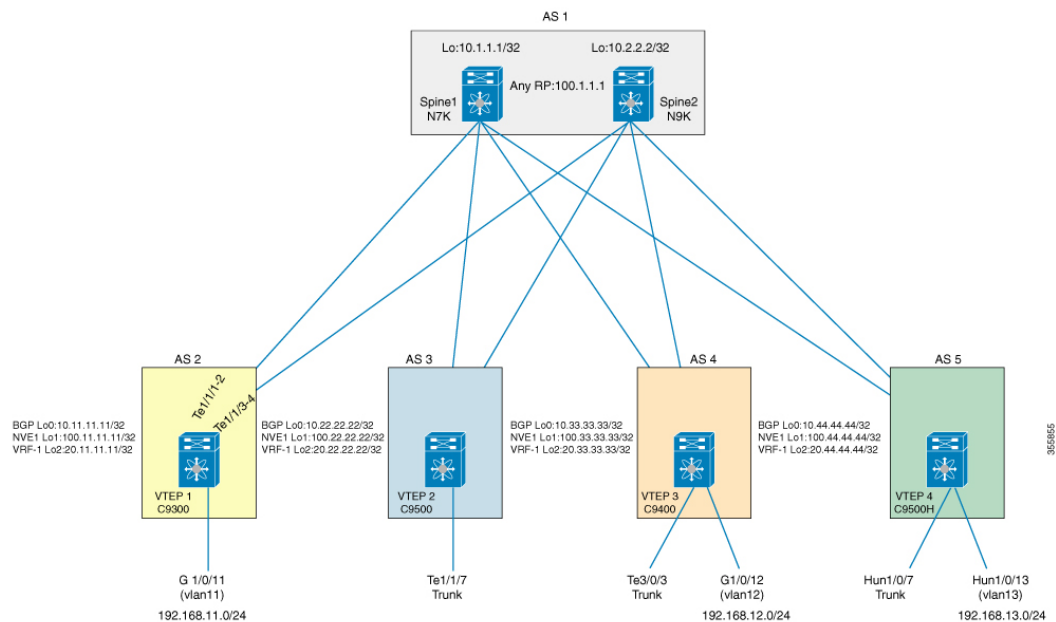


**Note** Although the **show ip bgp** command is available for verifying a BGP configuration, as a best practice, it is preferable to use the **show bgp** command instead.

## Examples of VXLAN BGP EVPN (EBGP)

### Example: Configuring eBGP Multi-AS EVPN VxLAN design model

Figure 1: shows the topology used in the eBGP Multi-AS design model



### Example: Configuring Underlay Transport (Unicast and Multicast) between all the VTEPs and the Spine(s):

eBGP peering between the spine and the VTEPs requires IP connectivity. This can be achieved by using static routes to reach loopback addresses between VTEPs and spines.

Configuring the spine

```
ip pim rp-address 100.1.1.1 group-list 239.0.0.0/8
ip pim rp-candidate loopback1 group-list 239.0.0.0/8
ip pim anycast-rp 100.1.1.1 10.1.1.1
ip pim anycast-rp 100.1.1.1 10.2.2.2
!
interface loopback0
ip address 10.1.1.1/32
ip pim sparse-mode
```

**Example: Configuring Underlay Transport (Unicast and Multicast) between all the VTEPs and the Spine(s):**

```

!
interface loopback1
ip address 100.1.1.1/32
ip pim sparse-mode
!
interface port-channel1
mtu 9198
medium p2p
ip address 10.10.1.1/30
ip pim sparse-mode
!
interface port-channel2
mtu 9198
medium p2p
ip address 10.10.2.1/30
ip pim sparse-mode
!
interface port-channel3
mtu 9198
medium p2p
ip address 10.10.3.1/30
ip pim sparse-mode

```

**Configuring the VTEP**

```

ip multicast-routing
ip pim rp-address 100.1.1.1
!
ip routing
!
interface Loopback0
ip address 10.11.11.11 255.255.255.255
ip pim sparse-mode
!
interface Loopback1
ip address 100.11.11.11 255.255.255.255
ip pim sparse-mode
!
interface Loopback2
vrf forwarding tenant_1
ip address 11.11.11.11 255.255.255.255
!
interface Port-channel1
no switchport
ip address 10.10.1.2 255.255.255.252
ip pim sparse-mode
!
interface Port-channel11
no switchport
ip address 20.20.1.2 255.255.255.252
ip pim sparse-mode
!
interface TenGigabitEthernet1/1/2
no switchport
no ip address
channel-group 1 mode active
!
interface TenGigabitEthernet1/1/3
no switchport
no ip address
channel-group 11 mode active

```



## Example: Configuring eBGP with EVPN address family between the Spine(s) and VTEPs:

### Configuring the spine

```
ip prefix-list lo_prefix seq 5 permit 0.0.0.0/0 le 32
route-map NH-UNCHANGED permit 10
set ip next-hop unchanged
route-map any_prefix permit 10
match ip address prefix-list lo_prefix
!
router bgp 1
router-id 10.1.1.1
log-neighbor-changes
address-family ipv4 unicast
redistribute direct route-map any_prefix
address-family l2vpn evpn
next-hop route-map NH-UNCHANGED
retain route-target all
!
neighbor 10.11.11.11
remote-as 2
update-source loopback0
ebgp-multihop 10
address-family ipv4 unicast
send-community both
soft-reconfiguration inbound
address-family l2vpn evpn
send-community both
route-map NH-UNCHANGED out
```

### Configuring the VTEP

```
router bgp 2
bgp router-id interface Loopback0
bgp log-neighbor-changes
bgp graceful-restart
neighbor 10.1.1.1 remote-as 1
neighbor 10.1.1.1 ebgp-multihop 10
neighbor 10.1.1.1 update-source Loopback0
!
address-family ipv4
redistribute connected
neighbor 10.1.1.1 activate
exit-address-family
!
address-family l2vpn evpn
neighbor 10.1.1.1 activate
neighbor 10.1.1.1 send-community both
maximum-paths 2
exit-address-family
!
address-family ipv4 vrf tenant_1
advertise l2vpn evpn
redistribute connected
exit-address-family
```

## Example: Configuring NVE on all VTEPs

### Configuring the VTEP

**Example: Configuring L2VPN EVPN on VTEPs**

```

interface nve1
no ip address
source-interface Loopback1
host-reachability protocol bgp
member vni 11001 mcast-group 239.0.1.1
member vni 11002 mcast-group 239.0.1.1
member vni 900001 vrf tenant_1

```

**Example: Configuring L2VPN EVPN on VTEPs**

Configuring the VTEP

```

l2vpn evpn
replication-type static
router-id Loopback1
!
l2vpn evpn instance 1 vlan-based
encapsulation vxlan
route-target export 2:1
route-target import 2:1
no auto-route-target
!
l2vpn evpn instance 2 vlan-based
encapsulation vxlan
route-target export 2:2
route-target import 2:2
no auto-route-target

```

**Example: Configuring Access customer facing VLAN VTEPs**

Configuring the VTEP

```

interface GigabitEthernet1/0/11
switchport access vlan 11
switchport mode access
!
interface TenGigabitEthernet1/1/7
switchport trunk allowed vlan 11-210,901-905
switchport mode trunk

```

**Example: Configuring additional VNI, EVI and VLAN on VTEPs**

```

VTEP1(config)#vlan 4000
VTEP1 (config-vlan)#state active
VTEP1 (config)#vlan configuration 4000
VTEP1 (config-vlan-config)#member evpn-instance 20000

```

Configuring the VTEP

```

vlan 11
state active
vlan 12
state active
vlan 901
state active
!
vlan configuration 11
member evpn-instance 1 vni 11001
!
vlan configuration 12

```

```

member evpn-instance 2 vni 11002
!
vlan configuration 901
member vni 900001
!
interface Vlan901
description connected to vni_900001
vrf forwarding tenant_1
ip unnumbered Loopback2
!
interface nve1
no ip address
source-interface Loopback1
host-reachability protocol bgp
member vni 11001 mcast-group 239.0.1.1
member vni 11002 mcast-group 239.0.1.1
member vni 900001 vrf tenant_1

```

## Example: Configuring IP VRF on VTEPs for Inter-VxLAN routing

Configuring the VTEP

```

vrf definition tenant_1
rd 1:1
!
address-family ipv4
route-target export 1:1
route-target import 1:1
route-target export 1:1 stitching
route-target import 1:1 stitching
exit-address-family

```

## Example: Configuring Access VLAN Interfaces (SVIs) on VTEPs

Configuring the VTEP

```

interface Vlan11
description vni_11001
mac-address 0001.0001.0001
vrf forwarding tenant_1
ip address 192.168.1.254 255.255.255.0
!
interface Vlan12
description vni_11002
mac-address 0001.0001.0001
vrf forwarding tenant_1
ip address 192.168.2.254 255.255.255.0

```

## Example: Configuring additional L3-VNI in NVE interfaces

Configuring the VTEP

```

interface nve1
no ip address
source-interface Loopback1
host-reachability protocol bgp
member vni 11001 mcast-group 239.0.1.1
member vni 11002 mcast-group 239.0.1.1
member vni 900001 vrf tenant_1

```

## Example: Configuring Core-facing VLANs and VLAN Interfaces

Configuring the VTEP

```
vlan configuration 901
member vni 900001
!
interface Vlan901
description connected to vni_900001
vrf forwarding tenant_1
ip unnumbered Loopback2
```

## Example: Configuring iBGP/IGP EVPN VxLAN design model

Configuring the spine:

```
feature-set fabric
hostname spine-1
!
feature telnet
feature scp-server
feature fabric forwarding
nv overlay evpn
feature ospf
feature bgp
feature pim
feature ipp
feature isis
feature fabric multicast
feature interface-vlan
feature lldp
feature fabric access
feature nv overlay
feature nxapi
!
ip pim rp-address 4.5.4.5 group-list 224.0.0.0/4
!
vlan 1
!
interface Vlan1
!
interface Ethernet1/1 ip address 10.14.1.4/24
ip router ospf 1 area 0.0.0.0
ip pim sparse-mode
no shutdown
!
interface loopback0
ip address 4.4.4.4/32
ip router ospf 1 area 0.0.0.0
ip pim sparse-mode
interface loopback1
ip address 4.5.4.5/32
ip router ospf 1 area 0.0.0.0
ip pim sparse-mode
!
router ospf 1
router-id 4.4.4.4
!
router bgp 100
router-id 4.4.4.4
address-family l2vpn evpn
neighbor 1.1.1.1 remote-as 100
```

```

update-source loopback0
address-family ipv4 unicast
send-community both
route-reflector-client
address-family l2vpn evpn
send-community both
route-reflector-client

```

### Configuring the VTEP

```

vrf definition l3vni50000
rd 101:1
!
address-family ipv4
route-target export 100:1 stitching
route-target import 100:1 stitching
exit-address-family
!
ip multicast-routing
ip pim rp-address 4.5.4.5
!
l2vpn evpn
replication-type static
!
vlan 10
State active
vlan 11
State active
vlan 501
state active
!
vlan configuration 10
member evpn-instance 10 vni 100010
!
vlan configuration 11
member evpn-instance 11 vni 100011
!
vlan configuration 501
member vni 50000
!
interface Loopback0
ip address 1.1.1.1 255.255.255.255
ip pim sparse-mode
ip ospf 1 area 0
!
interface GigabitEthernet1/0/1
switchport mode trunk
!
interface GigabitEthernet1/0/2
switchport access vlan 10
switchport mode access
!
interface TenGigabitEthernet3/0/1
description To Spine1
no switchport
ip address 10.14.1.1 255.255.255.0
ip pim sparse-mode
ip ospf 1 area 0
!
interface TenGigabitEthernet3/0/2
description To Spine1
no switchport
ip address 10.15.1.1 255.255.255.0
ip pim sparse-mode
ip ospf 1 area 0

```

```

!
interface Vlan10
description connected to 100010
mac-address 0001.0001.0001
vrf forwarding l3vni50000
ip address 192.168.10.1 255.255.255.0
!
interface Vlan11
description connected to 100011
mac-address 0001.0001.0001
vrf forwarding l3vni50000
ip address 192.168.11.1 255.255.255.0
!
interface Vlan501
description connected to 50000
vrf forwarding l3vni50000
ip unnumbered Loopback0
!
router ospf 1
router-id 1.1.1.1
nsr
!
router bgp 100
bgp router-id 1.1.1.1
bgp log-neighbor-changes
bgp graceful-restart
neighbor 4.4.4.4 remote-as 100
neighbor 4.4.4.4 update-source Loopback0
!
address-family ipv4
redistribute connected
neighbor 4.4.4.4 activate
exit-address-family
!
address-family l2vpn evpn
neighbor 4.4.4.4 activate
neighbor 4.4.4.4 send-community both
exit-address-family
!
address-family ipv4 vrf l3vni50000
advertise l2vpn evpn
redistribute connected
exit-address-family
!
interface nve1
no ip address
source-interface Loopback0
host-reachability protocol bgp
member vni 100010 mcast-group 227.0.0.1
member vni 100011 mcast-group 227.0.0.1
member vni 50000 vrf l3vni50000

```

## Example: Verifying L2/L3 VNI in NVE

```

# show nve vni
!
Interface VNI Type Peer-IP RMAC/Num_RTs eVNI state flags UP time
nve1 900001 L3CP 100.22.22.22 0011.2233.449d 900001 UP A/M 2d18h
nve1 900001 L3CP 100.44.44.44 380e.4d9b.5388 900001 UP A/M 2d18h
nve1 900001 L3CP 100.33.33.33 380e.4d9b.5448 900001 UP A/M 2d18h
nve1 11001 L2CP 100.22.22.22 2 11001 UP N/A 2d18h
nve1 11001 L2CP 100.33.33.33 2 11001 UP N/A 15:24:17
nve1 11001 L2CP 100.44.44.44 2 11001 UP N/A 2d18h

```

```
nve1 11002 L2CP 100.22.22.22 2 11002 UP N/A 2d18h
nve1 11002 L2CP 100.33.33.33 3 11002 UP N/A 5d16h
nve1 11002 L2CP 100.44.44.44 2 11002 UP N/A 2d18h
```

## Example: Verifying Multicast in multicast routing table

```
# show ip mroute
Leaf_1#sh ip mroute
IP Multicast Routing Table
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
(*, 239.0.1.1), 5d16h/stopped, RP 100.1.1.1, flags: SJCFx
Incoming interface: Port-channel11, RPF nbr 20.20.1.1
Outgoing interface list:
Tunnel0, Forward/Sparse-Dense, 5d16h/00:01:17
!
(100.11.11.11, 239.0.1.1), 00:02:18/00:00:41, flags: FTx
Incoming interface: Loopback1, RPF nbr 0.0.0.0, Registering
Outgoing interface list:
Port-channel11, Forward/Sparse, 00:02:18/00:03:14

# show ip mfib
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts: Total/RPF failed/Other drops
I/O Item Counts: FS Pkt Count/PS Pkt Count
Default
(*,224.0.0.0/4) Flags: C HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
(*,224.0.1.40) Flags: C HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
Port-channel11 Flags: A NS
Loopback0 Flags: F IC NS
Pkts: 0/0
(*,239.0.1.1) Flags: C HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 187/0/190/0, Other: 0/0/0
Port-channel11 Flags: A NS
Tunnel0, VXLAN Decap Flags: F NS
Pkts: 0/0
(100.11.11.11,239.0.1.1) Flags: HW
SW Forwarding: 0/0/0/0, Other: 0/0/0
HW Forwarding: 0/0/0/0, Other: 0/0/0
Null0 Flags: A NS
Port-channel11 Flags: F NS
Pkts: 0/0
Tunnel1 Flags: F
Pkts: 0/0

# show ip pim neighbors
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
P - Proxy Capable, S - State Refresh Capable, G - GenID Capable,
L - DR Load-balancing Capable
Neighbor Interface Uptime/Expires Ver DR
Address Prio/Mode
10.10.1.1 Port-channel1 5d16h/00:01:40 v2 1 / G
20.20.1.1 Port-channel11 5d16h/00:01:20 v2 1 / G
```

**Example: Verifying EVPN Instance in EVPN Manager**

```
# show ip pim tunnel
Tunnell*
Type : PIM Encap
RP : 100.1.1.1
Source : 20.20.1.2
State : UP
Last event : Created (5d16h)
# sh ip pim rp
Group: 239.0.1.1, RP: 100.1.1.1, uptime 5d16h, expires never
```

**Example: Verifying EVPN Instance in EVPN Manager**

```
# show l2vpn evpn evi 1 detail
EVPN instance: 1 (VLAN Based)
RD: 100.11.11.11:1 (auto)
Import-RTs: 2:1
Export-RTs: 2:1
Per-EVI Label: none
State: Established
Encapsulation: vxlan
Vlan: 11
Ethernet-Tag: 0
State: Established
Core If: Vlan901
Access If: Vlan11
RMAC: eclid.8b75.eac8
Core Vlan: 901
L2 VNI: 11001
L3 VNI: 900001
VTEP IP: 100.11.11.11
MCAST IP: 239.0.1.1
VRF: tenant_1
Pseudoports:
TenGigabitEthernet1/1/7 service instance 11
```

**Example: Verifying MAC Table**

```
# show mac address-table vlan 11
Mac Address Table
-----
Vlan Mac Address Type Ports
-----
11 0001.0001.0001 STATIC Vl11 -----□ SVI mac for Anycast Gateway
11 0011.0011.0005 DYNAMIC Te1/1/7-----□ dynamically learned
Total Mac Addresses for this criterion: 2
```

**Example: Verifying MAC entries in EVPN Manager**

```
# show l2vpn evpn mac
MAC Address EVI VLAN ESI Ether Tag Next Hop
-----
0011.0011.00c9 1 11 0000.0000.0000.0000.0000 0 Te1/1/7:11
0012.0012.0001 1 11 0000.0000.0000.0000.0000 0 100.22.22.22
0013.0013.0001 1 11 0000.0000.0000.0000.0000 0 100.33.33.33
0014.0014.0001 1 11 0000.0000.0000.0000.0000 0 100.44.44.44
```



## Example: Verifying MAC routes in BGP

```
# show bgp l2vpn evpn evi 1----- only evi 1 will be shown
BGP table version is 654847, local router ID is 10.11.11.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
x best-external, a additional-path, c RIB-compressed,
t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
Network Next Hop Metric LocPrf Weight Path
Route Distinguisher: 100.11.11.1:1
*> [2][100.11.11.11:1][0][48][0011001100C9][0][*]/20
:: 32768 ?
*> [2][100.11.11.11:1][0][48][001200120001][0][*]/20
100.22.22.22 0 1 3 ?
*> [2][100.11.11.11:1][0][48][001200120001][32][192.168.1.2]/24
100.22.22.22 0 1 3 ?
*> [2][100.11.11.11:1][0][48][001300130001][0][*]/20
100.33.33.33 0 1 4 ?
*> [2][100.11.11.11:1][0][48][001300130001][32][192.168.1.3]/24
100.33.33.33 0 1 4 ?
*> [2][100.11.11.11:1][0][48][001400140001][0][*]/20
100.44.44.44 0 1 4 ?
*> [2][100.11.11.11:1][0][48][001400140001][32][192.168.1.4]/24
100.44.44.44 0 1 4 ?
```

## Example: Verifying MAC routes in Layer 2 Routing Information Base

```
#show l2route evpn mac
EVI ETag Prod Mac Address Next Hop(s) Seq Number
-----
1 0 BGP 0012.0012.0001 V:11001 100.22.22.22 0
1 0 BGP 0013.0013.0001 V:11001 100.33.33.33 0
1 0 BGP 0014.0014.0001 V:11001 100.44.44.44 0
1 0 L2VPN 0011.0011.00c9 Tel1/1/7:11 0
```

## Example: Verifying IP VRF with all SVIs

```
# show ip vrf
Name                               Default RD           Interfaces
Mgmt-vrf                           <not set>           Gi0/0
tenant_1                             1:1                 Lo2
Vl11
Vl12
```

## Example: Verifying MAC/IP entries in MAC VRFs (EVIs)

```
# show bgp l2vpn evpn evi 1 route-type 2
BGP routing table entry for [2][100.11.11.11:1][0][48][0011001100C9][32][10.0.0.2]/24,
version 7
Paths: (1 available, best #1, table evi_1)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  Local
  :: (via default) from 0.0.0.0 (10.11.11.11)
  Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best
  EVPN ESI: 00000000000000000000, Label1 11001- L2 VNI
```

### Example: Verifying Remote MAC/IP and IP Prefix routes in L3VNI (IP VRF)

```

Extended Community: RT:2:1 ENCAP:8
Local irb vxlan vtep:
  vrf:tenant_1, l3-vni:900001-----□ IP VRF and L3 VNI
  local router mac:EC1D.8B75.EAC8
  core-irb interface:Vlan901----□ core SVI
  vtep-ip:100.11.11.11
  rx pathid: 0, tx pathid: 0x0

```

### Example: Verifying Remote MAC/IP and IP Prefix routes in L3VNI (IP VRF)

```

# show bgp vpnv4 unicast vrf tenant_1-----□ not all routes will be shown
BGP table version is 8583, local router ID is 10.11.11.11
  Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 1:1 (default for vrf tenant_1)
AF-Private Import to Address-Family: L2VPN E-VPN, Pfx Count/Limit: 11/1000
*> 11.11.11.11/32   0.0.0.0           0           32768 ?
*> 11.22.22.22/32   100.22.22.22      0           0 1 3 ?
*> 11.33.33.33/32   100.33.33.33      0           0 1 4 ?
*> 11.44.44.44/32   100.44.44.44      0           0 1 4 ?
* 192.168.1.0      100.44.44.44      0           0 1 4 ?
*                   100.33.33.33      0           0 1 4 ?
*                   100.22.22.22      0           0 1 3 ?
*>                   0.0.0.0           0           32768 ?
*> 192.168.1.2/32   100.22.22.22      0           0 1 3 ?
*> 192.168.1.3/32   100.33.33.33      0           0 1 4 ?
*> 192.168.1.4/32   100.44.44.44      0           0 1 4 ?
* 192.168.2.0      100.44.44.44      0           0 1 4 ?
*                   100.33.33.33      0           0 1 4 ?
*                   100.22.22.22      0           0 1 3 ?
*>                   0.0.0.0           0           32768 ?

```

### Example: Verifying IP routes are installed in L3 VNI (IP VRF)

```

# show ip route vrf tenant_1
Routing Table: tenant_1
Gateway of last resort is not set

  11.0.0.0/32 is subnetted, 3 subnets
C    11.11.11.11 is directly connected, Loopback2
B    11.22.22.22 [20/0] via 100.22.22.22, 00:13:21, Vlan901
B    11.33.33.33 [20/0] via 100.33.33.33, 00:13:21, Vlan901
B    11.44.44.44 [20/0] via 100.44.44.44, 00:12:51, Vlan901
  192.168.1.0/24 is variably subnetted, 4 subnets, 2 masks
C    192.168.1.0/24 is directly connected, Vlan11
B    192.168.1.3/32 [20/0] via 100.33.33.33, 16:26:48, Vlan901
B    192.168.1.4/32 [20/0] via 100.44.44.44, 2d19h, Vlan901
L    192.168.1.254/32 is directly connected, Vlan11
  192.168.2.0/24 is variably subnetted, 4 subnets, 2 masks
C    192.168.2.0/24 is directly connected, Vlan12
B    192.168.2.3/32 [20/0] via 100.33.33.33, 02:52:20, Vlan901
B    192.168.2.4/32 [20/0] via 100.44.44.44, 2d19h, Vlan901
L    192.168.2.254/32 is directly connected, Vlan12
  192.168.3.0/24 is variably subnetted, 4 subnets, 2 masks
C    192.168.3.0/24 is directly connected, Vlan13
B    192.168.3.3/32 [20/0] via 100.33.33.33, 2d19h, Vlan901

```

### Example: Verifying MAC/IP entries in EVPN Manager

```
#show l2vpn evpn mac ip
```

```
IP Address EVI VLAN MAC Address Next Hop(s)
-----
10.0.0.1 1 11 0011.0011.00c9 Tel1/1/7:11
10.0.0.2 1 11 0012.0012.0001 100.22.22.22
```

## Example: Verifying MAC/IP routes in Layer 2 Routing Informatio Base

```
#show l2route evpn mac ip
```

```
EVI ETag Prod Mac Address Host IP Next Hop(s)
-----
1 0 BGP 0012.0012.0001 10.0.0.2 V:11001 100.22.22.22
1 0 L2VPN 0011.0011.00c9 10.0.0.1 Tel1/1/7:11
```

