

EVPN VxLAN L3

This chapter provides information on Layer 3 Data-Center-Interconnect (DCI) VXLAN EVPN Support.

- Feature Information for EVPN VxLAN L3, on page 1
- Restrictions for EVPN VxLAN L3, on page 2
- Information About EVPN VxLAN L3, on page 2
- How to Configure EVPN VxLAN L3, on page 5
- Importing Between EVPN and VRF/VPN, on page 8
- Configure EVPN VxLAN Handoff, on page 8
- Verifying EVPN VxLAN L3, on page 14
- Verifying EVPN VxLAN Handoff, on page 15
- Configuring EVPN: Basic Configuration, on page 17
- Additional References for EVPN VxLAN L3, on page 26

Feature Information for EVPN VxLAN L3

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for EVPN VxLAN L3

Feature Name	Releases	Feature Information
EVPN VxLAN L3	Cisco IOS XE Denali 16.3.1	The EVPN VxLAN L3 is a new feature.
VXLAN EVPN Fabric DCI - MPLS L3VPN	Cisco IOS XE Everest 16.4.1	The VXLAN EVPN Fabric DCI - MPLS L3VPN is a new feature.
VXLAN EVPN External Connectivity	Cisco IOS XE Bengaluru 17.5.1	Support for IPv6 for the EVPN VxLAN L3 feature.

Restrictions for EVPN VxLAN L3

- VNI range CLI for L3VNI is not supported.
- Egress traffic stops, if local VNI is down.
- L3 VNI and L2 VNI cannot co-exist in the same bridge domain as L3 VNI is not supported.
- MAC learning is not done with L3VNI via control plane learning.
- External connectivity with VPLS networks is supported only when bridging is the mode of interworking between the two domains. Integrated routing and bridging (IRB) is not supported between a BGP EVPN VXLAN fabric and a VPLS network.
- External Connectivity with Layer 3 networks is supported only for IPv4 and IPv6 unicast traffic.
- MVPN network is not supported for multicast traffic.
- Import of EVPN IP routes, which includes route type 5 and route type 2 host which routes to the global routing table is not supported.

Information About EVPN VxLAN L3

Data Center Interconnect VXLAN Layer 3 Gateway

The Cisco device can serve as a Data Center Interconnect (DCI) L3 Gateway to provide IP connectivity between multi-tenant remote Data Center sites. The multi-tenant Data Centers use VxLAN encapsulation to carry separate tenant IP traffic. The VXLAN-enabled Data Center sites use MP-BGP EVPN control plane for distributing both Layer-2 and Layer-3 forwarding information within the site. RFC 5512 and draft-ietf-bess-evpn-inter-subnet-forwarding-00 define how MP-BGP Network Layer Reachability Information (NLRI) carries VXLAN encapsulation as well as L2/L3 forwarding information details to provide an integrated routing and bridging solution within the Data Center site.

Route Targets

For each VRF on the DCI router, there are two sets of manually configured import and export route-targets. One set of import and export route-targets is associated with the Data Center BGP neighbor that uses EVPN address-family to exchange L3 information; the other set of import and export route-targets is associated with the L3VPN BGP neighbor that use VPNv4 unicast address-family to exchange L3 information. This separation of route targets (RTs) enables the two sets of RTs to be independently configured. The DCI router effectively stitches the two set of RTs. The RTs associated with the EVPN BGP neighbor are labelled as stitching RTs. The RTs associated with the L3VPN BGP neighbor are normal RTs.

A new keyword is added to the existing route-target configuration to specify the route targets to be used when doing EVPN-VXLAN related processing. The base (existing) route target configuration does not affect EVPN-VXLAN related processing. You can have the same RT values for both base and VxLAN routes.

Local VPNv4 Routes Advertisement

On the DCI router, the locally sourced VPNv4 routes can be advertised to the BGP EVPN neighbors with the normal route targets (RTs) configured for the VRF or the stitching RTs associated with the BGP EVPN neighbors. By default, these routes are advertised with the normal route targets.



Note You cannot configure the advertise command for VPNv4 or VPNv6 neighbors. RTs can be applied only to the sourced routes and routes learned from VRF neighbors.

Data Center VXLAN with Support for MP-BGP

The Data Center VXLAN uses MP-BGP for control-plane learning of end-host Layer 2 and Layer 3 reachability information. The DCI router is configured with a VXLAN Tunnel EndPoint (VTEP). You also need to run the host-reachabilty protocol BGP command to specify that control-plane learning within Data center site is through BGP routing protocol.

The DCI Gateway router and the EVPN BGP neighbor (Data Center BGP neighbor) exchange BGP EVPN NLRIs of route type 5 that carry L3 routing information and associated VXLAN encapsulation information.

EVPN Route Targets

A new keyword is added to the existing route-target configuration to specify the route targets to be used when doing EVPN-VXLAN related processing. The base (existing) route target configuration does not affect EVPN-VXLAN related processing. You can have the same RT values for both base and vxlan routes

MAC/IP Advertisement Route and IP Prefix Route is supported. The l2vpn evpn address-family can be configured and neighbors can exchange EVPN NLRI. The l2vpn-evpn-prefix-advertisement is supported fully and for the non-MAC portions only the NLRI is supported. IP Prefix route type is added to carry IP prefixes. The IP Prefix NLRI can carry IPv4 Prefix or IPv6 Prefix. The NLRI length determines whether it has IPv4 Prefix or IPv6 Prefix.

NLRI Format:

```
[Type][Len][RD][ESI][ETag][IP Addr Mask][IP Addr][GW IP Addr][Label]
```

Key:

```
[Type][ETag][IP Addr Len][IP Addr]
```

When BGP attribute, encapsulation type EXTCOMM value of 0x8 (VxLAN) is present, then Label carries VNI (VXLAN ID).

EVPN RT5 and RT2 that contain a RT matching an import “stitching RT” specified in a vrf configuration is accepted by the router and imported into the corresponding BGP L3VPN vrf. The resulting L3VPN prefix retains the same route target. L3VPN routes that are imported into EVPN via “advertise l2vpn evpn” contains route targets specified by that vrf’s export “stitching RT”. Any original route targets is removed.

Bridge Domain Interface

Bridge Domain Interface (BDI) is used for Inter-VLAN routing for EVC. It supports ping from local BDI interface to peer BDI/BVI/SVI. ARP is not used to resolve adjacency. BGP is asked to advertise the BDI IP address in EVPN route and use RMAC as an adjacency.

Downstream VNI

A downstream VNI is assigned at the downstream BGP peer. The BGP peer sends VNI as part of EVPN route type 2 or 5, so that it can use the VNI to send EVPN traffic to peer. This VNI is called as egress VNI; this egress VNI is used to send EVPN traffic to peer on data path. BGP also sends the local VNI to peer as part of EVPN route type 2 or 5 and it is expected from the peer to send EVPN traffic with the VNI, so that it can route the PKT to right VRF. This VNI is called as ingress VNI.

For the local VNI, VNI number range is 4k to 16m. For the egress VNI, valid VNI number range can be any valid VNI number, from 1-16m.

Router MAC

EVPN introduces a Router's MAC extended community to exchange Router's MAC between EVPN peer. BGP send BDI's MAC address to EVPN Peer as its RMAC. By default, all the BDI interface share the same MAC address, so all EVPN VRF will send the same RMAC to EVPN peer by default. It is flexible to configure MAC address of BDI interface. So, it is possible that different EVPN VRF may send different RMAC to EVPN peer.

VRF Lite

VRF-lite (VPN routing/forwarding) allows a service provider to support two or more VPNs with overlapping IP addresses. VRF-lite is achieved by configuring sub-interfaces (VLANs) on a physical interface and by putting each sub-interface in a VRF.

EVPN Route Type 2 - MAC Advertisement Route

MAC Advertisement Route can be used to carry only MAC Address or MAC Address and IP Address (/32 for IPv4 or /128 for IPv6).

NLRI Format:

```
[Type][Len][RD][ESI][ETag][MAC Addr Mask][MAC Addr][IP Addr Len][IP Addr] [Label1] [Label2]
```

Key:

```
[Type][ETag][MAC Addr Len][MAC Addr][IP Addr Len][IP Addr]
[Type][ETag][MAC Addr Len][MAC Addr][IP Addr Len]
```

Label1 is associated with MAC Address and Label2 is associated with IP Address. When BGP attribute, encapsulation type EXTCOMM value of 0x8 (VxLAN) is present, then Label carries VNI (VXLAN ID).

L3 VRF EVPN Import

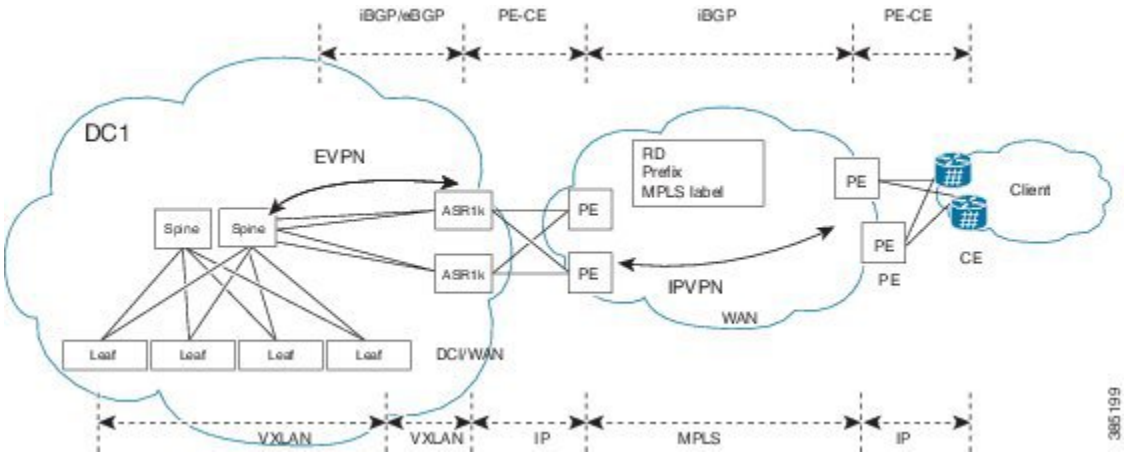
To advertise L3 VPN routing and forwarding (VRF) prefixes to EVPN neighbors define a new import type that takes prefixes from VRF neighbors, redistributed VRF routes, and import them into EVPN table. The import of VRF routes is controlled per VRF. The import of VRF is performed only when `advertise l2vpn evpn` is configured under that VRF and local VTEP is up.

EVPN DCI Solution

ASR1000 (IOS-XE Platform) series routers, acting as a Data Centre Interconnect (DCI) device can be deployed at the edge of two Cisco Data Center solutions, that is, Nexus 9000 Standalone-mode Data Centre or Nexus 9000 ACI-mode Data Centre. It provides flexible and safe WAN connections to the Internet or Branch sites with multiple different WAN types. Currently ASR1000 supports multiple WAN connection types, including iWAN, MPLS VPN(PE and ASBR), DMVPN, and VRF Lite. You can also deploy more than one ASR1000 router as multihoming deployment, if you require traffic load balancing, redundancy or customized path selection policy based on special requirements of different applications.

How to Configure EVPN VxLAN L3

The following is the sample topology that is used as an example to explain the configuration of this feature.



Configuring Customer Edge (CE) 1 Using VRF Lite

1. Define VRF and IPv4 address family. EVPN RT is 65535:1

```
vrf definition evpn1
 rd 65535:1
 address-family ipv4
route-target both 65535:1 stitching
exit-address-family
!
```

2. Define Bridge Domain and associate vxlan vni 3000.

```
bridge-domain 200
 member vni 30000
```

```
Interface loopback0
  ip address 33.33.33.33 255.255.255.255
```

3. Define Bridge Domain Interface (BDI).

```
interface BDI200
 vrf forwarding evpn1
 ip address 100.1.1.1 255.255.255.0
 encapsulation dot1Q 200
```

4. Create Interface NVE1.

```
Interface gi0/0/0.2
  enc dot1q 2
  ip address 4.0.0.1 255.255.255.0
Interface gi0/0/1.2
  enc dot1q 2
  vrf forwarding evpn1
  ip address 3.3.3.1 255.255.255.0
interface nve1
 no ip address
 source-interface Loopback0
 host-reachability protocol bgp
 member vni 30000 vrf evpn1
```

5. Define OSPF for underlay reachability.

```
Router ospf 100
  router-id 33.33.33.33
  network 33.33.33.33 0.0.0.0 area 0
  network 4.0.0.1 0.0.0.0 area 0
!
```

6. Define BGP and EVPN address-family.

```
router bgp 65535
 bgp router-id 33.33.33.33
 neighbor 44.44.44.44 remote-as 65535
 neighbor 44.44.44.44 update-source Loopback0
 !
 address-family l2vpn evpn
  neighbor 44.44.44.44 activate
  neighbor 44.44.44.44 send-community both
 exit-address-family
 !
 address-family ipv4 vrf evpn1
  advertise l2vpn evpn
  neighor 3.3.3.254 remote-as 65530
  neighor 3.3.3.254 update-source Gi0/0/1.2
  neighor 3.3.3.254 ebgp-multihop 255
  redistribute connected
 exit-address-family
```

Configuring Provider Edge 1

Define VRF and RD/RT.

```
vrf definition vrf1
 rd 65530:1
 address-family ipv4
  route-target both 65530:1
exit-address-family
!
interface loopback0
```

```
  ip address 33.33.33.22 255.255.255.255
Interface GigabitEthernet0/0/0.2
  enc dot1q 2
  vrf forwarding vrf1
  ip address 3.3.3.254 255.255.255.0
Interface gigabitEthernet0/0/1
  mpls ip
  ip address 2.2.2.1 255.255.255.0
!
Router ospf 100
  router-id 33.33.33.22
  network 33.33.33.22 0.0.0.0 area 0
  network 2.2.2.1 0.0.0.0 area 0
!

router bgp 65530
 bgp router-id 33.33.33.22
 neighbor 22.22.22.22 remote-as 65530
 neighbor 22.22.22.22 update-source Loopback0
 !
 address-family vpnv4
  neighbor 22.22.22.22 activate
  neighbor 22.22.22.22 send-community both
 exit-address-family
 !
 address-family ipv4 vrf vrf1
  neighor 3.3.3.253 remote-as 65535
  neighor 3.3.3.253 update-source Gi0/0/0.2
  neighor 3.3.3.253 ebgp-multihop 255
  redistribute connected
 exit-address-family
```

Configuring Provider Edge 2 and Branch Router

```
vrf definition vrf1
 rd 65530:1
 address-family ipv4
  route-target both 65530:1
exit-address-family
!
interface loopback0
  ip address 22.22.22.22 255.255.255.255
!
Interface GigabitEthernet0/0/0.200
  enc dot1q 200
  vrf forwarding vrf1
  ip address 1.1.1.254 255.255.255.0
!
Interface gigabitEthernet0/0/1
  mpls ip
  ip address 2.2.2.254 255.255.255.0
!
Router ospf 100
  router-id 22.22.22.22
  network 22.22.22.22 0.0.0.0 area 0
  network 2.2.2.254 0.0.0.0 area 0
!

router bgp 65530
 bgp router-id 22.22.22.22
 neighbor 33.33.33.22 remote-as 65530
 neighbor 33.33.33.22 update-source Loopback0
 !
 address-family vpnv4
```

```
 neighbor 33.33.33.22 activate
 neighbor 33.33.33.22 send-community both
exit-address-family
!
address-family ipv4 vrf vrf1
  redistribute connected
exit-address-family
```

Configuring Customer Edge 2

```
Interface GigabitEthernet0/0/0.200
  enc dot1q 200
ip address 1.1.1.1 255.255.255.0
ip route 0.0.0.0 0.0.0.0 1.1.1.254
```

Importing Between EVPN and VRF/VPN

```
router bgp 100
 address-family ipv4 vrf example-vrf
  advertise l2vpn evpn
  neighbor 7.7.7.7 remote-as 400
  neighbor 7.7.7.7 activate
 exit-address-family
```

Configure EVPN VxLAN Handoff

External connectivity or handoff refers to the movement of Layer 2 and Layer 3 traffic between an EVPN VXLAN network and an external network. This connectivity enables the EVPN VXLAN network to exchange routes with the externally connected network.

The EVPN VXLAN network imports the reachability routes from the external network and extends the Layer 2 or Layer 3 overlay network outside the VXLAN network. The process of extending a Layer 2 or Layer 3 network outside the EVPN VXLAN network is also known as handoff.

The following procedures tell you how to configure the external handoff between an EVPN VXLAN network and an external layer 2 or a layer 3 network. To enable the EVPN VLAN Layer 3 external handoff with an MPLS Layer 3 VPN network:

1. Run the **mpls label mode all-vrfs protocol all-afs per-vrf** command in the global configuration mode on the border VTEP.
2. Configure BGP with with a new route type for Layer 2 VPN, VPNv4, VPNv6 address families on the border VTEP.
3. Configure BGP on the border VTEP for external connectivity with the MPLS Layer 3 VPN.

Configuration on the EVPN VxLAN Fabric

Run the following configuration on the EVPN VxLAN fabric. Here, the EVPN routes imported from the EVPN fabric into the VPNv4 address family as VPNv4 routes and distributes them to the external network.

```
vrf definition evpn0
 rd 65535:4000
```

```
 !
 address-family ipv4
  route-target export 65535:1
  route-target import 65535:1
  route-target export 65535:1 stitching
  route-target import 65535:1 stitching
 exit-address-family
!
 address-family ipv6
  route-target export 65535:1
  route-target import 65535:1
  route-target export 65535:1 stitching
  route-target import 65535:1 stitching
  exit-address-family
!
interface GigabitEthernet3
 ip address 10.0.4.1 255.255.255.0
!
interface Loopback0
 ip address 4.4.4.1 255.255.255.255
!
interface BDI200
 vrf forwarding evpn0
 ip address 150.1.1.1 255.255.255.0
 ipv6 address 2000::150:1:1:1/112
interface nve1
 no ip address
 source-interface Loopback0
 host-reachability protocol bgp
 member vni 50000 vrf evpn0
!
bridge-domain 200
 member vni 50000
!
router bgp 65535
 bgp log-neighbor-changes
 bgp graceful-restart
 neighbor 6.6.6.1 remote-as 1
 neighbor 6.6.6.1 ebgp-multihop 3
 neighbor 6.6.6.1 update-source Loopback0
 !
 address-family l2vpn evpn
  neighbor 6.6.6.1 activate
  neighbor 6.6.6.1 send-community both
 exit-address-family
 !
 address-family ipv4 vrf evpn0
  advertise l2vpn evpn
  redistribute connected
 exit-address-family
 !
 address-family ipv6 vrf evpn0
  redistribute connected
  advertise l2vpn evpn
 exit-address-family
!
router ospf 10
 nsf cisco
 network 4.4.4.1 0.0.0.0 area 0
 network 10.0.4.1 0.0.0.0 area 0
```

Configuration on the Border Node

Perform the following configuration on the border node. Here, the lsvpn evpn re-originate command imports the EVPN route into VPNv6 address family and carried over L3-VPN MPLS network.

The import vpnv4 unicast and the vpnv6 unicast commands enable the import of the VPN route into the EVPN table.

```
vrf definition evpn0
 rd 65535:6000
 !
 address-family ipv4
  route-target export 65535:1
  route-target import 65535:1
  route-target export 65535:1 stitching
  route-target import 65535:1 stitching
 exit-address-family
!
 address-family ipv6
  route-target export 65535:1
  route-target import 65535:1
  route-target export 65535:1 stitching
  route-target import 65535:1 stitching
 exit-address-family
!
bridge-domain 200

 member vni 40000

!
interface Loopback0

 ip address 6.6.6.1 255.255.255.255

!

interface Loopback10

 ip address 60.60.60.1 255.255.255.255
!
interface GigabitEthernet2
 ip address 10.0.4.2 255.255.255.0
!
interface GigabitEthernet4
 ip address 102.2.2.2 255.255.255.0
 ip ospf 1 area 0
 mpls ip
!
interface BDI200
 vrf forwarding evpn0
 ip address 200.1.1.1 255.255.255.0
 ipv6 address 2000::200:1:1:1/112
!
interface nve1

 no ip address
 source-interface Loopback0
 host-reachability protocol bgp
 member vni 40000 vrf evpn0
!
router bgp 1

 bgp log-neighbor-changes
```

```
bgp graceful-restart
no bgp default ipv4-unicast
neighbor 4.4.4.1 remote-as 65535
neighbor 4.4.4.1 ebgp-multihop 3
neighbor 4.4.4.1 update-source Loopback0
neighbor 8.8.8.1 remote-as 1
neighbor 8.8.8.1 update-source Loopback10
!
address-family ipv4
exit-address-family
!
address-family vpnv4
 import l2vpn evpn re-originate
 neighbor 8.8.8.1 activate
 neighbor 8.8.8.1 send-community both
 neighbor 8.8.8.1 next-hop-self all
exit-address-family
!
address-family ipv6
exit-address-family
!
address-family vpnv6
 import l2vpn evpn re-originate
 neighbor 8.8.8.1 activate
 neighbor 8.8.8.1 send-community both
 neighbor 8.8.8.1 next-hop-self
exit-address-family
!
address-family l2vpn evpn
 import vpnv4 unicast
 import vpnv6 unicast
 neighbor 4.4.4.1 activate
 neighbor 4.4.4.1 send-community both
```

```
 exit-address-family

 !

 address-family ipv4 vrf evpn0

  advertise l2vpn evpn

  redistribute connected

 exit-address-family

 !
 address-family ipv6 vrf evpn0

  advertise l2vpn evpn

  redistribute connected

 exit-address-family
!
router ospf 1
 redistribute connected
 passive-interface GigabitEthernet2 !don't notify the ospf to gi2
!
router ospf 10
 nsf cisco
 passive-interface GigabitEthernet4
 network 6.6.6.1 0.0.0.0 area 0
 network 10.0.4.2 0.0.0.0 area 0
```

Configuration on the WAN Network

Perform the following configuration on the WAN network. In this BGP VPN configuration, the VPNv6 route is received from the border node and is imported into the VRF based on the VRF RT configuration. This route from the VRF is advertised as the VPN route to the remote end.

```
vrf definition evpn0
 rd 65535:8000
 !
 address-family ipv4
  route-target export 65535:1
  route-target import 65535:1
 exit-address-family
!
 address-family ipv6
  route-target export 65535:1
  route-target import 65535:1
 exit-address-family

!
interface Loopback0

 ip address 8.8.8.1 255.255.255.255
!
interface Loopback10
 vrf forwarding evpn0

 ip address 58.1.1.1 255.255.255.255
 ipv6 address 2000::58:1:1:1/128
!
```

```
interface GigabitEthernet2

 ip address 102.2.2.1 255.255.255.0

 ip ospf 1 area 0

 speed 1000

 mpls ip
!
router bgp 1

 bgp log-neighbor-changes
 bgp graceful-restart
 neighbor 60.60.60.1 remote-as 1

 neighbor 60.60.60.1 update-source Loopback0

 !
address-family vpnv4
  neighbor 60.60.60.1 activate

  neighbor 60.60.60.1 send-community both

  neighbor 60.60.60.1 route-reflector-client

  neighbor 60.60.60.1 next-hop-self

 exit-address-family

 !

 address-family vpnv6

  neighbor 60.60.60.1 activate

  neighbor 60.60.60.1 send-community both

  neighbor 60.60.60.1 route-reflector-client

 exit-address-family

 !

 address-family ipv4 vrf evpn0

  redistribute connected  ! redistribute local network to evpn

  redistribute static  ! redistribute static route to evpn

 exit-address-family
!
 address-family ipv6 vrf evpn0
  redistribute connected
  redistribute static
 exit-address-family
!
router ospf 1
 nsf cisco
 redistribute connected

 network 0.0.0.0 255.255.255.255 area 0
```

-

Verifying EVPN VxLAN L3

Use the following commands to verify the configuration:

- **show ip bgp l2vpn evpn**: Displays Layer 2 Virtual Private Network (L2VPN) address family information from the Border Gateway Protocol (BGP) table.
- **show mlrib evpn mac**: Displays the MLRIB information pertaining to an EVPN network.
- **show nve peers**: Displays information that determine if the VNI is configured for peer.

Show Command-BGP

```
#show ip bgp l2vpn evpn summary
BGP router identifier 19.0.0.1, local AS number 1
BGP table version is 2, main routing table version 2
1 network entries using 376 bytes of memory
1 path entries using 196 bytes of memory
1/1 BGP path/bestpath attribute entries using 272 bytes of memory
1 BGP extended community entries using 40 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 884 total bytes of memory
BGP activity 1/0 prefixes, 1/0 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
20::46          4            1    7852    7849        2    0    0 4d22h           0
19.0.101.1      4            1       0       0        1    0    0 never    Idle
19.0.101.2      4            1       0       0        1    0    0 never    Idle
19.0.101.3      4            1       0       0        1    0    0 never    Idle
19.0.101.4      4            1       0       0        1    0    0 never    Idle
19.0.101.5      4            2       0       0        1    0    0 never    Idle
19.0.101.6      4            1       0       0        1    0    0 never    Idle
19.0.101.7      4            1   80385    7853        2    0    0 4d22h           1
20.0.0.47       4            1    7857    7844        2    0    0 4d22h           0
FEC0::1001      4            1       0       0        1    0    0 never    Idle
```

Show Command-MLRIB

```
# show mlrib evpn mac
EVI   MAC Address  Owner Next-Hop  iVNI  eVNI
-----------------------------------------------------
100   aaa.bbb.cc1   NVE   1.2.3.4   10000 1000

# show mlrib evpn mac detailed
EVI MAC Address Owner Next-Hop  iVNI  eVNI lVTEP     port
------------------------------------------------------------
100 aaa.bbb.cc1 NVE   1.2.3.4   1000  1000 1.2.3.2   2000

# show mlrib evpn vtep local
BD    RMAC Address    VTEP-IP       VRF     VNI  BDI
-----------------------------------------------------------
100   aaa.bbb.cc2    101.2.3.4      vrf1    10000    BDI100
```

Show NVE Peers

```
#sh nve peers vni 10135
Interface  VNI      Type Peer-IP          Router-RMAC    eVNI     state flags UP time
nve1       10135    L3CP 66.66.66.66      5c83.8f5f.5c97 10135      UP   A/M 00:08:53
```

Verifying EVPN VxLAN Handoff

To verify whether the external handoff is successful, run the following show commands:

Procedure

Step 1 Verify EVPN by running the show evpn peers command.

a) VxLAN Fabric

```
router#show evpn peers
Intf     VNI      Type   Peer-IP          Router-RMAC    eVNI     state flags   UP time
nve1     50000    L3NVE  6.6.6.1          001e.bd6e.22bf 40000      UP  A/-/4  01:48:57
nve1     50000    L3NVE  6.6.6.1          001e.bd6e.22bf 40001      UP  A/-/4  01:48:57
nve1     50000    L3NVE  6.6.6.1          001e.bd6e.22bf 40002      UP  A/M/4  01:48:57
nve1     50000    L3NVE  6.6.6.1          001e.bd6e.22bf 40000      UP  A/-/6  01:48:57
```

b) Border Node

```
BorderRouter#show evpn peers
'M' - MAC entry download flag  'A' - Adjacency download flag
'4' - IPv4 flag  '6' - IPv6 flag

Intf     VNI      Type   Peer-IP          Router-RMAC    eVNI     state flags   UP time
nve1     40000    L3NVE  4.4.4.1          001e.1403.63bf 50000      UP  A/-/4  01:58:36
nve1     40001    L3NVE  4.4.4.1          001e.1403.63bf 50000      UP  A/M/4  01:58:36
nve1     40002    L3NVE  4.4.4.1          001e.1403.63bf 50000      UP  A/M/4  01:58:36
nve1     40000    L3NVE  4.4.4.1          001e.1403.63bf 50000      UP  A/M/6  01:58:36
```

Step 2 Verify the BGP EVPN RT5 route by running the show bgp l2vpn evpn route-type 5 command:

a) VxLAN Fabric

```
Router#show bgp l2vpn evpn route-type 5
BGP routing table entry for [5][65535:1][0][112][2000::58:1:1:0]/29, version 10
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 1
  1
    6.6.6.1 (metric 2) (via default) from 6.6.6.1 (60.60.60.1)
      Origin incomplete, localpref 100, valid, external, best
      EVPN ESI: 00000000000000000000, Gateway Address: ::, VNI Label 40000, MPLS VPN Label 0
      Extended Community: RT:65535:1 ENCAP:8 Router MAC:001E.BD6E.22BF
      rx pathid: 0, tx pathid: 0x0
      net: 0x7F3224F19DF8, path: 0x7F32250DCAE0, pathext: 0x7F31B92F7D38
      flags: net: 0x0, path: 0x3, pathext: 0x81
      attribute: 0x7F32250CD030, ref: 5
      Updated on Feb 3 2021 18:05:04 UTC
```

b) Border Node

```
BorderNode#show bgp l2vpn evpn route-type 5
...
BGP routing table entry for [5][65535:1][0][112][2000::58:1:1:0]/29, version 7
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Advertised to update-groups:
     1
  Refresh Epoch 1
  Local, imported path from base
    ::FFFF:8.8.8.1 (metric 2) (via default) from 8.8.8.1 (80.80.80.1)
```

```
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      EVPN ESI: 00000000000000000000, Gateway Address: 0.0.0.0, local vtep: 6.6.6.1, VNI Label
40000, MPLS VPN Label 24
      Extended Community: RT:65535:1 ENCAP:8 Router MAC:001E.BD6E.22BF
      rx pathid: 0, tx pathid: 0x0
     net: 0x7F67C81E5128, path: 0x7F67B8AC0210, pathext: 0x7F67C8371CC0, exp_net: 0x7F67C74227F8

      flags: net: 0x0, path: 0x7, pathext: 0xA1
      attribute: 0x7F67C7661858, ref: 3
      Updated on Feb 3 2021 18:04:47 UTC
```

Step 3 Verify the BGP binding label by running the show ip bgp vpnv6 unicast vrf command:

a) VxLAN Fabric

```
Router#sh ipv6 cef vrf evpn0 2000::58:1:1:0/112 internal
2000::58:1:1:0/112, epoch 0, flags [rnolbl, rlbls], RIB[B], refcnt 5, per-destination sharing
  sources: RIB
  feature space:
    IPRM: 0x00018000
    Broker: linked, distributed at 3rd priority
  ifnums:
    BDI200(26): 6.6.6.1
  path list 7F321ABA2F28, 5 locks, per-destination, flags 0x269 [shble, rif, rcrsv, hwcn, bgp]
    path 7F322454CB88, share 1/1, type recursive, for IPv6
      recursive via 50331650[Binding-Sid Label:Default], fib 7F31B6375230, 1 terminal fib,
bslbl:Default:50331650
      path list 7F321ABA3318, 3 locks, per-destination, flags 0x49 [shble, rif, hwcn]
          path 7F322454D068, share 1/1, type attached nexthop, for IPv6, cid [40000]
            nexthop 6.6.6.1 BDI200, IPV6 adj out of BDI200, addr 6.6.6.1, cid: 40000 7F3224F853F8

  output chain:
    IPV6 adj out of BDI200, addr 6.6.6.1, cid: 40000 7F3224F853F8
```

```
Router# show ip bgp vpnv6 unicast vrf evpn0 2000::58:1:1:0/112 internal
BGP routing table entry for [65535:1]2000::58:1:1:0/112, version 4
Paths: (1 available, best #1, table evpn0)
Not advertised to any peer
Refresh Epoch 1
1, imported path from [5][65535:1][0][112][2000::58:1:1:0]/29 (global)
::FFFF:6.6.6.1(metric 2) (via default) from 6.6.6.1(60.60.60.1)
Origin incomplete, localpref 100, valid, external, best
Extended Community: RT:65535:1 ENCAP:8 Router MAC:001E.BD6E.22BF
Local vxlan vtep:
vrf:evpn0, vni:50000
local router mac:001E.1403.63BF
encap:8
vtep-ip:4.4.4.1
bdi:BDI200
Remote VxLAN [attr 0x7F31B8B5E9C8(Ref:7), rnh 0x7F321C245668(Ref:2)]:
Topoid 0x1E000002(vrf evpn0)
Remote Router MAC:001E.BD6E.22BF
Encap 8
Egress VNI 40000
RTEP 6.6.6.1
Binding Label: 0x3000002
rx pathid: 0, tx pathid: 0x0
net: 0x7F31B8B4E7A8, path: 0x7F31B8B7EAA8, pathext: 0x7F3224EFAA88, exp_net: 0x7F3224F19DF8
flags: net: 0x0, path: 0x7, pathext: 0x181
attribute: 0x7F31B8B5E9C8, ref: 6
Updated on Feb 3 2021 18:05:04 UTC
```

Configuring EVPN: Basic Configuration

Perform the following tasks to configure EVPN:

1. Create a VRF.

```
vrf definition EVPN
 rd 100:1
 !
 address-family ipv4
  route-target export 100:1 stitching
  route-target import 100:1 stitching
 exit-address-family
```

2. Create a bridge domain and assign a VNI.

```
bridge-domain 1234
 member vni 101234
```

3. Create a BDI interface and assign it to the EVPN VRF.

```
interface BDI1234
 vrf forwarding EVPN
 ip address 10.20.30.40 255.255.255.0
 encapsulation dot1Q 1234
```

4. Create an NVE interface.

```
interface nve1
 no ip address
 source-interface Loopback1
 host-reachability protocol bgp
 member vni 101234 vrf EVPN

router bgp 100
 bgp router-id 10.10.10.10
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.10.10.111 remote-as 100
 neighbor 10.10.10.111 ebgp-multihop 255
 neighbor 10.10.10.111 update-source Loopback1
 neighbor 10.10.10.222 remote-as 100
 neighbor 10.10.10.222 ebgp-multihop 255
 neighbor 10.10.10.222 update-source Loopback1
 !
```

5. Configure a EVPN sessions to two spines.

```
address-family l2vpn evpn
  neighbor 10.10.10.111 activate
  neighbor 10.10.10.111 send-community both
  neighbor 10.10.10.222 activate
  neighbor 10.10.10.222 send-community both
 exit-address-family
```

Example: EVPN Interconnect With MPLS VPN as ASBR

```
router bgp 100
 bgp router-id 10.10.10.10
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
```

```
neighbor 9.9.8.8 remote-as 200
Neighbor 9.9.8.8 ebgp-multihop 255

neighbor 9.9.8.8 update-source Loopback0
!
address-family vpnv4
 import l2vpn evpn
 neighbor 9.9.8.8 activate
 neighbor 9.9.8.8 send-community extended
 neighbor 9.9.8.8 next-hop-self all
 Neighbor 9.9.8.8 inter-as-hybrid
```

Configuring Inter-AS Option AB

The following sections describe how to configure the Inter-AS Option AB feature on an ASBR for either an MPLS VPN or an MPLS VPN that supports CSC:



Note If Inter-AS Option AB is already deployed in your network and you want to do Option B style peering for some prefixes (that is, implement Inter-AS Option AB+), configure the **inter-as-hybrid global** command as described in the “Configuring the Routing Policy for VPNs that Need Inter-AS Connections” section.

Configuring the VRFs on the ASBR Interface for Each VPN Customer

Use the following steps to configure the VRFs on the ASBR interface for each VPN customer so that these VPNs have connectivity over the MPLS VPN--Inter-AS Option AB network.



Note The **mpls bgp forwarding** command is used only on the ASBR interface for VRFs that support CSC.

Use all of the steps in the following procedure to configure additional VRFs that need to be configured on the ASBR interface and the VRFs that need to be configured on the peer ASBR interface.

1. Enable privileged EXEC mode. Enter your password if prompted.

   ```
   enable
   Example:
   Router> enable
   ```

2. Enter global configuration mode.

   ```
   configure terminal
   Example:
   Router# configure terminal
   ```

3. Specify the interface to configure and enter the interface configuration mode.

   - The *type* argument specifies the type of interface to be configured.
   - The *number* argument specifies the port, connector, or interface card number.

   ```
   interface type number
   Example:
    Router(config)# interface Ethernet 5/0
   ```

4. Associate a VRF with the specified interface or subinterface.

- The *vrf-name* argument is the name assigned to a VRF.

```
ip vrf forwarding vrf-name
Example:
 Router(config-if)# ip vrf forwarding vpn1
```

5. (Optional) Configures BGP to enable MPLS forwarding on connecting interfaces for VRFs that must support MPLS traffic.
   - This step applies to a CSC network only.

```
mpls bgp forwarding
Example:
 Router(config-if)# mpls bgp forwarding
```

6. (Optional) Exits to privileged EXEC mode.

```
end
Example:
 Router(config-if)# end
```

Configuring MP-BGP Session Between ASBR Peers

BGP propagates reachability information for VPN-IPv4 prefixes among PE routers by means of the BGP multiprotocol extensions (see RFC 2283, *Multiprotocol Extensions for BGP-4*), which define support for address families other than IPv4. Using the extensions ensures that the routes for a given VPN are learned only by other members of that VPN, enabling members of the VPN to communicate with each other.

Follow the steps in this section to configure the MP-BGP session on the ASBR.

Use all of the steps in the following procedure to configure the MP BGP session on the peer ASBR.

1. Enable privileged EXEC mode. Enter your password if prompted.

```
enable
Example:
Router> enable
```

2. Enter global configuration mode.

```
configure terminal
Example:
Router# configure terminal
```

3. Configures a BGP routing process and places the router in router configuration mode.
   - The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.

```
router bgp as-number
Example:
 Router(config)# router bgp 100
```

4. Adds an entry to the BGP or multiprotocol BGP neighbor table.
   - The *ip-address* argument specifies the IP address of the neighbor.
   - The *peer-group-name* argument specifies the name of a BGP peer group.
   - The *as-number* argument specifies the autonomous system to which the neighbor belongs.

```
neighbor {ip-address | peer-group-name} remote-as as-number
Example:
 Router(config-router)# neighbor 192.168.0.1
remote-as 200
```

5. Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes.
   - The **unicast** keyword specifies IPv4 unicast address prefixes.

```
address-family vpnv4 [unicast]
Example:
 Router(config-router)# address-family vpnv4
```

6. Enables the exchange of information with a neighboring router.
   - The *ip-address* argument specifies the IP address of the neighbor.
   - The *peer-group-name* argument specifies the name of a BGP peer group.

```
neighbor {ip-address | peer-group-name} activate
Example:
 Router(config-router-af)# neighbor 192.168.0.1
activate
```

7. Configures eBGP peer router (ASBR) as an Inter-AS Option AB peer.
   - The *ip-address* argument specifies the IP address of the neighbor.
   - The *peer-group-name* argument specifies the name of a BGP peer group.
   - If any prefixes are imported into Option AB VRFs, then the imported paths are advertised to this peer.
   - If any prefixes are received from this peer and are imported into Option AB VRFs, then the imported paths are advertised to iBGP peers.



Note Advertised routes have RTs that are configured on the VRF. Advertised routes do not have their original RTs.

```
neighbor {ip-address | peer-group-name} inter-as-hybrid
Example:
 Router(config-router-af)# neighbor 192.168.0.1
inter-as-hybrid
```

8. Exits from address family configuration mode.

```
exit-address-family
Example:
 Router(config-router-af)# exit-address-family
```

9. (Optional) Exits to privileged EXEC mode.

```
end
Example:
 Router(config-af)# end
```

Configuring the Routing Policy for VPNs that Need Inter-AS Connections

Use the steps in this section to configure VRFs for the VPNs that need Inter-AS connections between ASBR peers, by configuring the appropriate routing policy and Option AB configuration.

Use all of the steps in the following procedure to configure additional VPNs that need Inter-AS Option AB connectivity on this ASBR and the peer ASBR.

1. Enable privileged EXEC mode. Enter your password if prompted.

```
enable
Example:
Router> enable
```

2. Enter global configuration mode.

```
configure terminal
Example:
Router# configure terminal
```

3. Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode.
   - The *vrf-name* argument is the name assigned to a VRF.

```
vrf definition vrf-name
Example:
 Router(config)# vrf definition vpn1
```

4. Creates routing and forwarding tables.
   - The *route-distinguisher* argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter an RD in either of these formats:
     - 16-bit autonomous system number: your 32-bit number, for example, 101:3
     - 32-bit IP address: your 16-bit number, for example, 192.168.122.15:1

```
rd route-distinguisher
Example:
 Router(config-vrf)# rd 100:1
```

5. Enters VRF address family configuration mode to specify an address family for a VRF.
   - The **ipv4** keyword specifies an IPv4 address family for a VRF.

```
address-family ipv4
Example:
Router(config-vrf)# address-family ipv4
```

6. Creates a route-target extended community for a VRF.
   - The **import** keyword imports routing information from the target VPN extended community.
   - The **export** keyword exports routing information to the target VPN extended community.
   - The **both** keyword imports routing information from and exports routing information to the target VPN extended community.
   - The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF list of import, export, or both (import and export) route-target extended communities.

```
route-target {import | export | both}
route-target-ext-community
```

```
Example:
 Router(config-vrf-af)# route-target import
100:1
```

7. For Inter-AS Option AB+, go to Step 10; otherwise, go to Step 8.
8. Specifies the VRF as an Option AB VRF, which has the following effects:
   - Routes imported to this VRF can be advertised to Option AB peers and VPNv4 iBGP peers.
   - When routes received from Option AB peers and are imported into the VRF, the next hop table ID of the route is set to the table ID of the VRF.
   - If the **csc** keyword is not used, a per-VRF label is allocated for imported routes.
   - When routes are received from Option AB peers and are imported next into the VRF, the learned out label can be installed only in forwarding when the **csc** keyword is used.

   The **csc** keyword implies the following:
   - A per-prefix label is allocated for imported routes.
   - For routes received from Option AB peers that are imported into the VRF, the learned out label is installed in forwarding.

```
inter-as-hybrid [csc]
Example:
 Router(config-vrf-af)# inter-as-hybrid
```

9. (Optional) Specifies the next hop IP address to be set on paths that are imported into the VRF and that are received from an Option AB peer.
   - The next hop context is also set to the VRF, which imports these paths.
   - The **csc** keyword implies the following:
     - A per-prefix label is allocated for imported routes.
     - For routes received from Option AB peers that are imported into the VRF, the learned out label is installed in forwarding.

```
inter-as-hybrid next-hop global
Example:
 Router(config-vrf-af)# inter-as-hybrid next-hop
global
```

10. (For Option AB+) Enables Inter-AS Option AB+.
    - Specifies that the next-hop address for BGP updates to be set on paths that are imported to the VRF and that are received from an Option AB+ peer are placed in the global routing table.
    - The address used is the address of the interface that is at the remote end of the external BGP (eBGP) global shared link. The next-hop context is retained as global and not modified to that of the importing VRF.

```
inter-as-hybrid next-hop global
Example:
 Router(config-vrf-af)# inter-as-hybrid next-hop
global
```

11. (Optional) Exits to privileged EXEC mode.

```
end
Example:
 Router(config-vrf-af)# end
```

Example: EVPN Interconnect With MPLS VPN as ASBR

```
router bgp 100
 bgp router-id 10.10.10.10
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 9.9.8.8 remote-as 200
 Neighbor 9.9.8.8 ebgp-multihop 255

 neighbor 9.9.8.8 update-source Loopback0
 !
 address-family vpnv4
  import l2vpn evpn
  neighbor 9.9.8.8 activate
  neighbor 9.9.8.8 send-community extended
  neighbor 9.9.8.8 next-hop-self all
  Neighbor 9.9.8.8 inter-as-hybrid
```

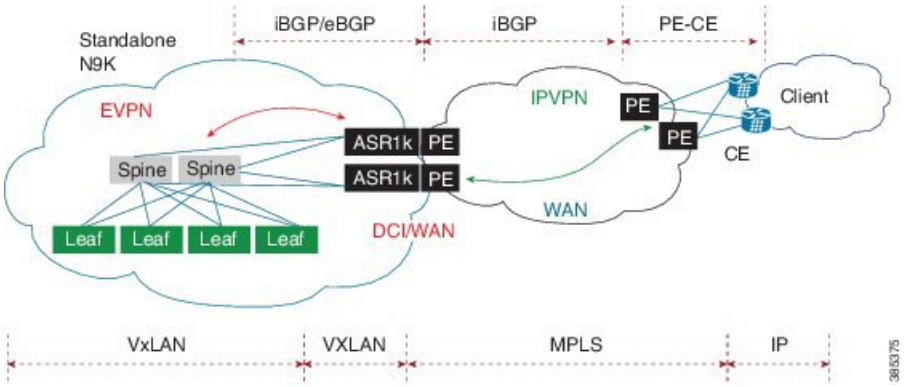
Configuring EVPN Interconnect With MPLS VPN as PE

ASR1000 supports direct prefix redistribution between BGP VPNv4 and BGP L2VPN EVPN address families. ASR1000 can act as gateway of Data Centre network and PE of MPLS VPN network both. It receives MPLS VPN prefixes from P/PE routers and these prefixes can be imported into BGP EVPN rib and then forwarded to DC's spine via BGP EVPN session. It can also import BGP EVPN prefixes sent by spine into BGP VPNv4 rib and send to P/PE in MPLS VPN network. During the prefixes redistribution, ASR1k set itself as the next-hop of the prefix before sending update to its neighbors.

In this release (16.4.1), ASR1000 only supports only bi-directional redistribution between EVPN and VPNv4. Redistribution between EVPN and VPNv6 is not supported.

In the scenario explained in the below figure shows, ASR1k acting as a PE in the MPLS-VPN network. Firstly, VRF is needed for the EVPN RT-5 routes to be imported, and then re-originate as VPN route into the MPLS-VPN side. VPN route that is learnt from the MPLS-VPN side will then first be imported into VRF, and the re-originated into EVPN as RT-5 routes.

Figure 1: EVPN Interconnect With MPLS VPN as PE



1. Define VRF and IPv4 address family.

```
vrf definition EVPN
 rd 100:1
 !
 address-family ipv4
  route-target import 100:1
  route-target import 100:1
  route-target export 100:1 stitching
  route-target import 100:1 stitching
 exit-address-family
!
```

2. Configure interface Loopback0.

```
interface Loopback0
MPLS VPN
 ip address 9.9.10.10 255.255.255.255
 ip router isis vpn
 Ip ospf 100 area 0
!
```

3. Configure interface GigabitEthernet.

```
interface GigabitEthernet0/0/0
facing MPLS VPN P/PE
 ip address 9.9.108.10 255.255.255.0
 ip router isis vpn
 negotiation auto
 mpls ip
cdp enable
!
Interface gi0/0/1.4
  Description facing to ACI spine
 Encapsulation dot1q 4
  Ip address 10.10.10.1 255.255.255.0
  Ip ospf 100 area 0
```

4. Create Interface NVE1.

```
interface nve1
 no ip address
 source-interface Loopback0
 host-reachability protocol bgp
 member vni 101234 vrf EVPN
!
```

5. Configure bridge domain.

```
Bridge-domain 100
  Member vni 101234
Interface bdi100
  Vrf forwarding EVPN
   Encapsulation dot1q 100
 Ip address 9.10.0.1 255.255.255.0

Router ospf 100
  Router-id 9.9.10.10
  Area 0.0.0.100 nssa

router isis vpn
 net 49.0001.1010.1010.1010.00
 is-type level-2-only
 metric-style wide
!
```

6. Define BGP and EVPN address-family.

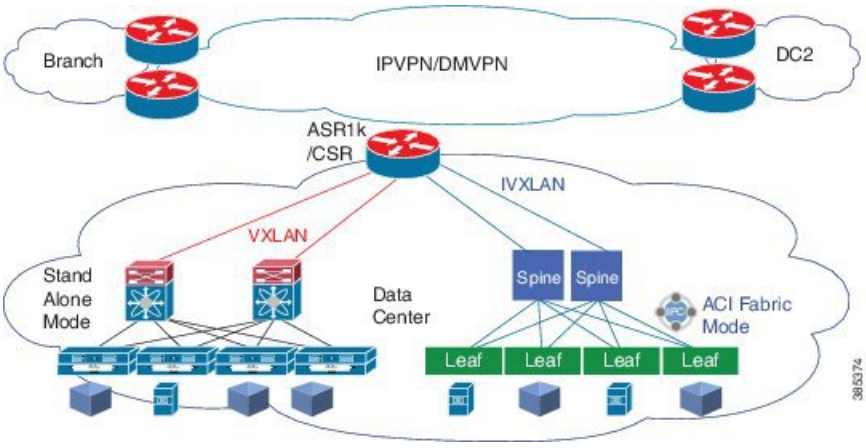
```
router bgp 200
 bgp router-id 10.10.10.10
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 9.9.8.8 remote-as 200
 neighbor 9.9.8.8 update-source Loopback0
 neighbor 10.10.10.111 remote-as 100
 neighbor 10.10.10.111 ebgp-multihop 255
 neighbor 10.10.10.111 update-source Loopback0
 neighbor 10.10.10.222 remote-as 100
 neighbor 10.10.10.222 ebgp-multihop 255
 neighbor 10.10.10.222 update-source Loopback0
 !
 address-family vpnv4
  import l2vpn evpn
  neighbor 9.9.8.8 activate
  neighbor 9.9.8.8 send-community extended
  neighbor 9.9.8.8 next-hop-self all
 exit-address-family
 !
 address-family l2vpn evpn
  import vpnv4 unicast
  neighbor 10.10.10.111 activate
  neighbor 10.10.10.111 send-community both
  neighbor 10.10.10.222 activate
  neighbor 10.10.10.222 send-community both
 exit-address-family
 !
```

7. Define VXLAN UDP port.

```
vxlan udp port 0xBEEF
```

Configuring DCI EVPN Peer to ACI Spine

Figure 2: DCI EVPN Peer to ACI Spine



1. Configure interface.

```
Interface gi0/0/1.4
  Description facing to ACI spine
 Encapsulation dot1q 4
  Ip address 10.10.10.1 255.255.255.0
```

```
  Ip ospf 100 area 0
```

2. Configure bridge domain.

```
Bridge-domain 100
  Member vni 101234
Interface bdi100
  Vrf forwarding EVPN
   Encapsulation dot1q 100
 Ip address 9.10.0.1 255.255.255.0

Router ospf 100
  Router-id 9.9.10.10
  Area 0.0.0.100 nssa
vxlan udp port 0xBEEF
```

Additional References for EVPN VxLAN L3

MIBs

MIB	MIBs Link
• CISCO-EVPN-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html