Forwarding Configurations

Forwarding Configurations

Forwarding Configurations for Cisco Nexus 5600, 7000 and 9000 Series Switches in the Programmable Fabric

Use these configurations for configuring your Cisco Nexus 5600, 7000 and 9000 Series switches.

Note For ease of use, the configuration mode from which you need to start configuring a task is mentioned at the beginning of each configuration.

Cisco Nexus 5600 Series switch configuration

The following configurations are required for the Cisco Nexus 5600 Series switch for supporting BGP-EVPN with VXLAN overlay. Note that most of the configurations required for enabling VXLAN remain the same, EVPN configurations are what will be the emphasis here:

1. Initial configuration - Install the network virtualization overlay, BGP, and EVPN features on the VTEPs.
2. Implement Layer 2 VNI configurations for tenant networks within a tenant.
3. Implement Layer 3 VNI configurations for the tenant.

Note Though configuration examples are mainly IPv4, IPv6 addresses are also supported in the VXLAN EVPN fabric.

Initial configuration

(config) #

```
install feature-set fabric
feature-set fabric
feature fabric forwarding
feature interface-vlan
feature ospf
OR
feature isis
```

Attention You can use either OSPF or IS-IS as the routing protocol.

(config) #

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

Configure the anycast gateway MAC address

(config) #

```
fabric forwarding anycast-gateway-mac 2020.0000.00aa
```

Configure BGP L2VPN EVPN address family

(config) #

```
router bgp 100
  neighbor 10.1.1.53 remote-as 100
    update-source loopback0
    address-family l2vpn evpn
      send-community both
```

Layer 2 VNI configurations for a tenant network

Associate a VLAN to the Layer 2 VNI

(config) #

```
vlan 200
  vn-segment 30000
```

Create a loopback interface for BGP and assign an IP address to it

(config) #

```
interface loopback 0
   ip address 10.1.1.54/32
```

Create a loopback interface for NVE and assign an IP address to it

(config) #

```
interface loopback 1
   ip address 10.1.2.54/32
```

Associate the Layer 2 VNI to the overlay and configure multicast group membership

(config) #

```
interface nve 1
 no shutdown
 source-interface loopback1
 host-reachability protocol bgp
 member vni 30000
   suppress-arp
   mcast-group 239.1.1.0
```

Associate the Layer 2 VNI to the EVPN address family, and enable route distinguisher and route target functions for the VNI

(config) #

```
evpn
  vni 30000 l2
    rd auto
    route-target import auto
    route-target export auto
```

Note Alternatively, the following config can also be used:

```
evpn
  vni 30000 l2
    rd auto
    route-target both auto
```

The combination of the **router BGP** command (configured earlier) and the **evpn** command ensures that BGP EVPN is configured to advertise 'MAC route' or 'MAC + associated host routes' of servers attached to the VTEP, for the specified Layer 2 VNI (Route type 2 [Refer to the EVPN RFC document for more details]). By default, the MAC route will be advertised, and the associated host route will be advertised if either there is an SVI configured for that VLAN in anycast-gateway mode or if suppress-arp option is enabled for that L2 VNI (See *ARP Suppression* section).

In the above NVE example, the MAC+IP routes for the hosts are advertised into BGP-EVPN for hosts belonging to layer 2 VNI 30000.

Layer 3 VNI configurations for a tenant

Associate the VRF VNI (Layer 3 VNI) to the customer VRF.

Enable VRF route distinguisher and route target functions.

(config) #

```
vrf context coke
  vni 50000
  rd auto
  address-family ipv4 unicast
     route-target both auto evpn
```

In the above example, the option *both* is used to import and export routes associated with the Layer 3 VNI 50000.

Associate the VRF VNI to a VLAN and associate an SVI to the customer VRF

(config) #

```
vlan 2200
```

```
   vn-segment 50000
```

(config) #

```
interface vlan 2200
   vrf member coke
   ip forward
   ipv6 forward
   no ip redirects
   no ipv6 redirects
   no shutdown
```

In order to avoid the overhead of creating a core facing vlan and corresponding SVI on a per vrf basis, we also provide an option of using a vrf-tenant-profile that automatically takes care of this. Note that if there is a **vrf-tenant-profile** configured, then the user must ensure the following CLIs related to dynamic and core-VLANs are also enabled.

(config) #

```
system fabric dynamic-vlans 100-2400
system fabric core-vlans 100-300
```

switch #

```
configure profile vrf-tenant-profile
  vlan $vrfVlanId
    vn-segment $vrfSegmentId
  interface vlan $vrfVlanId
    vrf member $vrfName
      ip forward
      ipv6 forward
      no ip redirects
      no ipv6 redirects
      no shutdown
end
```

Add the Layer 3 VRF VNI to the overlay network

(config) #

```
interface nve 1
  host-reachability protocol bgp
  member vni 50000 associate-vrf
```

Associate the customer VRF to BGP and enable L2VPN EVPN route distribution

(config) #

```
router bgp 100
  vrf coke
    address-family ipv4 unicast
      advertise l2vpn evpn
```

Enable host/server facing SVI (and associate it to a VRF) for Layer 3 connectivity on the distributed anycast gateway

(config) #

```
interface vlan 200
  vrf member coke
  ip address 209.165.202.129/27
  fabric forwarding mode anycast-gateway
```

Cisco Nexus 5600 Series switches verification

For verification of MAC routes, refer these commands:

```
switch# show mac address-table dynamic

Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link
   VLAN     MAC Address      Type      age     Secure NTFY   Ports/SWID.SSID.LID
---------+-----------------+--------+---------+------+----+------------------
* 200      2010.0000.0010    dynamic   270      F      F     Eth100/1/1
* 200      2010.0000.0011    dynamic   0        F      F     nve1/10.1.1.56
* 200      2010.0000.0012    dynamic   0        F      F     nve1/10.1.1.74
* 200      2010.0000.0013    dynamic   0        F      F     nve1/10.1.1.56
* 200      8080.c800.0038    dynamic   0        F      F     nve1/10.1.1.74
* 1        24e9.b392.316b    dynamic   1190     F      F     Eth100/1/1
```

```
switch# show l2route evpn mac all

Topology    Mac Address    Prod   Next Hop (s)
----------- -------------- ------ ---------------
200         2010.0000.0010 Local  Eth100/1/1
200         2010.0000.0011 BGP    10.1.1.56
200         2010.0000.0012 BGP    10.1.1.74
200         2010.0000.0013 BGP    10.1.1.56
200         8080.c800.0038 BGP    10.1.1.74
2200        002a.6ab2.0181 VXLAN  10.1.1.56
2200        8c60.4f14.2efc VXLAN  10.1.1.74
```

Command output description

Prod (producer) column displays the source of origination of the MAC address.

Local means a MAC address learnt locally via a server facing or edge port, **BGP** means the remote end host MAC was learnt from a remote VTEP via BGP-EVPN and **VXLAN** indicates the router MAC of the remote VTEP as carried in the extended community in the BGP advertisement.

```
switch# show bgp l2vpn evpn

BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 198, local router ID is 10.1.1.54
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist, I-injected
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup

   Network            Next Hop            Metric     LocPrf     Weight Path
Route Distinguisher: 10.1.1.54:32967    (L2VNI 30000)
*>l[2]:[0]:[0]:[48]:[2010.0000.0010]:[0]:[0.0.0.0]/216
                      10.1.1.54                          100      32768 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0011]:[0]:[0.0.0.0]/216
                      10.1.1.56                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0012]:[0]:[0.0.0.0]/216
                      10.1.1.74                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0013]:[0]:[0.0.0.0]/216
                      10.1.1.56                          100          0 i
*>i[2]:[0]:[0]:[48]:[8080.c800.0038]:[0]:[0.0.0.0]/216
                      10.1.1.74                          100          0 i
*>l[2]:[0]:[0]:[48]:[2010.0000.0010]:[32]:[200.0.0.10]/272
                      10.1.1.54                          100      32768 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0011]:[32]:[200.0.0.11]/272
                      10.1.1.56                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0012]:[32]:[200.0.0.12]/272
                      10.1.1.74                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0013]:[32]:[200.0.0.13]/272
                      10.1.1.56                          100          0 i
*>i[2]:[0]:[0]:[48]:[8080.c800.0038]:[32]:[200.0.0.56]/272
                      10.1.1.74                          100          0 i
```

```
Route Distinguisher: 10.1.1.56:3
*>i[5]:[0]:[0]:[24]:[209.165.202.130]:[0.0.0.0]/224
                      10.1.1.56                 0        100          0 ?

Route Distinguisher: 10.1.1.56:32967
*>i[2]:[0]:[0]:[48]:[2010.0000.0011]:[0]:[0.0.0.0]/216
                      10.1.1.56                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0013]:[0]:[0.0.0.0]/216
                      10.1.1.56                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0011]:[32]:[209.165.202.140]/272
                      10.1.1.56                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0013]:[32]:[209.165.202.142]/272
                      10.1.1.56                          100          0 i
Route Distinguisher: 10.1.1.74:32967
*>i[2]:[0]:[0]:[48]:[2010.0000.0012]:[0]:[0.0.0.0]/216
                      10.1.1.74                          100          0 i
*>i[2]:[0]:[0]:[48]:[8080.c800.0038]:[0]:[0.0.0.0]/216
                      10.1.1.74                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0012]:[32]:[209.165.202.141]/272
                      10.1.1.74                          100          0 i
*>i[2]:[0]:[0]:[48]:[8080.c800.0038]:[32]:[209.165.202.143]/272
                      10.1.1.74                          100          0 i
```

```
switch# show nve peers

Interface Peer-IP    State LearnType Uptime   Router-Mac
--------- ---------------  ----- --------- -------- -----------------
nve1      10.1.1.56   Up    CP        1d12h   002a.6ab2.0181
nve1      10.1.1.74   Up    CP        1d12h   8c60.4f14.2efc
```

For verification of IP host and prefix routes, refer these commands:

```
switch# show ip arp vrf coke

Flags: * - Adjacencies learnt on non-active FHRP router
       + - Adjacencies synced via CFSoE
       # - Adjacencies Throttled for Glean
       D - Static Adjacencies attached to down interface

IP ARP Table for context coke
Total number of entries: 1
Address         Age       MAC Address      Interface
209.165.202.13  00:18:23  2010.0000.0010   Vlan200
```

```
switch# show ip route vrf coke

IP Route Table for VRF "coke"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.1.1.0/24, ubest/mbest: 1/0, attached
    *via 10.1.1.1, Vlan10, [0/0], 1d12h, direct
10.1.1.1/32, ubest/mbest: 1/0, attached
    *via 10.1.1.1, Vlan10, [0/0], 1d12h, local
209.165.202.130/27, ubest/mbest: 1/0, attached
    *via 209.165.202.129, Vlan200, [0/0], 1d12h, direct, tag 12345,
209.165.202.129/32, ubest/mbest: 1/0, attached
    *via 209.165.202.129, Vlan200, [0/0], 1d12h, local, tag 12345,
209.165.202.139/32, ubest/mbest: 1/0, attached
    *via 209.165.202.139, Vlan200, [190/0], 1d12h, hmm
209.165.202.140 /32, ubest/mbest: 1/0
    *via 10.1.1.56%default, [200/0], 1d12h, bgp-100, internal, tag 100,  (mpls-vpn)segid
50000 tunnel: 16843064 encap: 1
```

Command output description

Direct means that the subnet prefix is configured locally under a Layer-3 interface on this switch. **Local** means the IP address belongs to the switch aka locally configured under a Layer-3 interface on that switch (10.1.1.254/24).

```
switch# show l2route evpn mac-ip all

Topology ID Mac Address    Prod     Host IP           Next Hop(s)
----------- -------------- ---- --------------------------------------- --------
-------
200         2010.0000.0010 HMM      209.165.202.139   N/A

200         2010.0000.0011 BGP      209.165.202.140   10.1.1.56

200         2010.0000.0012 BGP      209.165.202.141   10.1.1.74

200         2010.0000.0013 BGP      209.165.202.142   10.1.1.56

200         8080.c800.0038 BGP      209.165.202.143   10.1.1.74
```

```
switch# show bgp l2vpn evpn

Route Distinguisher: 10.1.1.54:3    (L3VNI 50000)
*>i[2]:[0]:[0]:[48]:[2010.0000.0011]:[32]:[209.165.202.144]/272
                      10.1.1.56                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0012]:[32]:[209.165.202.141]/272
                      10.1.1.74                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0013]:[32]:[209.165.202.143]/272
                      10.1.1.56                          100          0 i
*>l[5]:[0]:[0]:[24]:[209.165.202.130]:[0.0.0.0]/224
                      10.1.1.54                0        100      32768 ?
* i                   10.1.1.56                0        100          0 ?
```

Cisco Nexus 7000 Series switch configuration

The following BGP, EVPN and overlay configurations are required for the Cisco Nexus 7000 Series and 7700 Series switches with F3 or M3 cards:

1. Initial configuration - Install the network virtualization overlay, BGP, and EVPN features on the VTEPs.
2. Layer 2 VNI configurations for tenant networks within a tenant.
3. Layer 3 VNI configurations for a tenant.

Note Though configuration examples are mainly IPv4, IPv6 addresses are also supported in the VXLAN EVPN fabric.

VXLAN BGP EVPN configuration for the Cisco Nexus 7000 Series switches is also available here. While the 7.2 release only supported the border leaf and border spine functionality, the 7.3 version in addition also supports the leaf functionality.

A switch VDC with M3 modules cannot perform the role of a VXLAN BGP EVPN leaf switch.

Initial configurations

(config) #

```
install feature-set fabric
feature-set fabric
feature fabric forwarding
```

```
feature interface-vlan
feature ospf
OR
feature isis
```

Attention You can use either OSPF or IS-IS as the underlay routing protocol.

Note The **install feature-set fabric** command should only be used in the admin VDC. When using a VDC, ensure the VDC is of type F3 or M3, for EVPN. A sample configuration is given below:
(config) #

```
vdc test
    limit-resource module-type f3
```

(config) #

```
feature nv overlay
feature bgp
feature vni
nv overlay evpn
```

Configure the anycast gateway MAC address

(config) #

```
fabric forwarding anycast-gateway-mac 2020.0000.00aa
```

Configure BGP L2VPN EVPN address family

(config) #

```
router bgp 100
   neighbor 10.1.1.53 remote-as 100
     update-source loopback0
     address-family l2vpn evpn
        send-community both
```

Layer 2 VNI configurations for a tenant network

Create a bridge domain and associate the Layer 2 VNI with it

(config) #

```
vni 30000
system bridge-domain 200-210
bridge-domain 200
   member vni 30000
```

While the **system bridge-domain** command identifies the bridge domain IDs, the **bridge-domain** command configures the specified bridge domain(s).

Associate a VLAN (or dot1q tag) with the Layer 2 VNI:

(config) #

```
encapsulation profile vni cisco
  dot1q 50 vni 30000
```

Note For an access port, you should use the **untagged** keyword, as shown below.

```
encapsulation profile vni ACCESS
  untagged vni 30000
```

Associate the encapsulation profile with the server facing interface

(config) #

```
interface Ethernet 1/12
   no shutdown
   no switchport
   service instance 1 vni
   encapsulation profile cisco default
      no shutdown
```

Create a loopback interface for BGP and assign an IP address to it

(config) #

```
interface loopback 0
   ip address 10.1.1.54/32
```

Create a loopback interface for NVE and assign an IP address to it

(config) #

```
interface loopback 1
   ip address 10.1.2.54/32
```

Associate the Layer 2 VNI to the overlay and configure multicast group membership

(config) #

```
interface nve 1
   no shutdown
   source-interface loopback0
   host-reachability protocol bgp
   member vni 30000
      suppress-arp
      mcast-group 239.1.1.0
```

Enable EVPN and associate the Layer 2 VNI to it

Enable route distinguisher and route target functions for the Layer 2 VNI

(config) #

```
evpn
  vni 30000 l2
    rd auto
    route-target import auto
    route-target export auto
```

Note that with the Cisco Nexus 7000 Series switches, a VNI is associated with a bridge-domain (1:1). Refer to the respective configuration guide for more information on bridge-domains. The combination of the **router BGP** command (configured earlier) and the **evpn** command ensures that BGP EVPN is configured to advertise 'MAC route' or 'MAC + associated host routes' of servers attached to the VTEP, for the specified Layer 2 VNI.

In the above NVE example, MAC+ IP routes are advertised into BGP-EVPN for hosts belonging to layer 2 VNI 30000.

Layer 3 VNI configurations for a tenant

Associate the VRF VNI to the customer VRF

Enable VRF route distinguisher and VRF route target functions for the Layer 3 VNI

(config) #

```
vrf context coke
   vni 50000
   rd auto
   address-family ipv4 unicast
      route-target both auto evpn
```

In the above example, the option *both* is used to import and export routes associated with the Layer 3 VNI 50000. Specifically, the layer-3 routes will be advertised with route-target 100:50000 where 100 is the BGP Autonomous system number and 50000 is the layer-3 VNI.

Associate the VRF VNI to a bridge-domain and associate a BDI to the customer VRF

(config) #

```
system bridge-domain add 2200
vni 50000
bridge-domain 2200
  member vni 50000

interface bdi2200
  vrf member coke
  ip forward
  no ip redirects
  no shutdown
```

While the **system bridge-domain** command identifies the bridge domain IDs, the **bridge-domain** command configures the specified bridge domain(s).

Add the Layer 3 VRF VNI to the overlay network and enable BGP reachability

(config) #

```
interface nve 1
   host-reachability protocol bgp
   member vni 50000 associate-vrf
```

Configure BGP, associate the customer VRF to BGP and enable L2VPN EVPN route distribution

(config) #

```
router bgp 100
   vrf coke
      address-family ipv4 unicast
         advertise l2vpn evpn
```

Enable host/server facing BDI (and associate it to a VRF) for Layer 3 connectivity on the distributed anycast gateway

(config) #

```
interface bdi200
   vrf member coke
   ip address 10.1.1.1/24
   fabric forwarding mode anycast-gateway
```

```
   no shutdown
```

Cisco Nexus 7000 Series switches verification

For verification of MAC routes, refer these commands:

The following is sample output to verify that end host MAC addresses (local and remote) are added to the MAC address table:

```
switch# show mac address-table dynamic

Note: MAC table entries displayed are getting read from software.
 Use the 'hardware-age' keyword to get information related to 'Age'

 Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link, E -
 EVPN entry
        (T) - True, (F) - False ,  ~~~ - use 'hardware-age' keyword to retrieve
age info

VLAN/BD   MAC Address      Type      age     Secure NTFY Ports/SWID.SSID.LID
---------+-----------------+--------+---------+------+----+------------------

* 200      2010.0000.0010    dynamic   270      F    F   Eth100/1/1
* 200      2010.0000.0011    dynamic   0        F    F   nve1/10.1.1.56
* 200      2010.0000.0012    dynamic   0        F    F   nve1/10.1.1.74
* 200      2010.0000.0013    dynamic   0        F    F   nve1/10.1.1.56
* 200      8080.c800.0038    dynamic   0        F    F   nve1/10.1.1.74
* 1        24e9.b392.316b    dynamic   1190     F    F   Eth100/1/1
```

The following is sample output for viewing MAC addresses of end hosts across all EVPN instances (EVIs) pertaining to the switch:

```
switch# show l2route evpn mac all

Topology    Mac Address    Prod   Next Hop (s)
----------- -------------- ------ ---------------
200         2010.0000.0010 Local  Eth100/1/1
200         2010.0000.0011 BGP    10.1.1.56
200         2010.0000.0012 BGP    10.1.1.74
200         2010.0000.0013 BGP    10.1.1.56
200         8080.c800.0038 BGP    10.1.1.74
2200        002a.6ab2.0181 VXLAN  10.1.1.56
2200        8c60.4f14.2efc VXLAN  10.1.1.74
```

The following sample output displays BGP routing table information for the L2VPN EVPN address family. It includes route distinguisher and next hop information.

```
switch # show bgp l2vpn evpn

BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 198, local router ID is 10.1.1.54
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist, I-injected
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup

   Network            Next Hop            Metric     LocPrf     Weight Path

Route Distinguisher: 10.1.1.54:32967    (L2VNI 30000)
*>l[2]:[0]:[0]:[48]:[2010.0000.0010]:[0]:[0.0.0.0]/216
                      10.1.1.54                          100      32768 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0011]:[0]:[0.0.0.0]/216
                      10.1.1.56                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0012]:[0]:[0.0.0.0]/216
                      10.1.1.74                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0013]:[0]:[0.0.0.0]/216
                      10.1.1.56                          100          0 i
*>i[2]:[0]:[0]:[48]:[8080.c800.0038]:[0]:[0.0.0.0]/216
                      10.1.1.74                          100          0 i
```

```
*>l[2]:[0]:[0]:[48]:[2010.0000.0010]:[32]:[209.165.202.139]/272
                      10.1.1.54                          100      32768 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0011]:[32]:[209.165.202.140]/272
                      10.1.1.56                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0012]:[32]:[209.165.202.141]/272
                      10.1.1.74                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0013]:[32]:[209.165.202.142]/272
                      10.1.1.56                          100          0 i
*>i[2]:[0]:[0]:[48]:[8080.c800.0038]:[32]:[209.165.202.143]/272
                      10.1.1.74                          100          0 i

Route Distinguisher: 10.1.1.56:3
*>i[5]:[0]:[0]:[24]:[209.165.202.130]:[0.0.0.0]/224
                      10.1.1.56                 0        100          0 ?

Route Distinguisher: 10.1.1.56:32967
*>i[2]:[0]:[0]:[48]:[2010.0000.0011]:[0]:[0.0.0.0]/216
                      10.1.1.56                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0013]:[0]:[0.0.0.0]/216
                      10.1.1.56                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0011]:[32]:[209.165.202.140]/272
                      10.1.1.56                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0013]:[32]:[209.165.202.142]/272
                      10.1.1.56                          100          0 i

Route Distinguisher: 10.1.1.74:32967
*>i[2]:[0]:[0]:[48]:[2010.0000.0012]:[0]:[0.0.0.0]/216
                      10.1.1.74                          100          0 i
*>i[2]:[0]:[0]:[48]:[8080.c800.0038]:[0]:[0.0.0.0]/216
                      10.1.1.74                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0012]:[32]:[209.165.202.141]/272
                      10.1.1.74                          100          0 i
*>i[2]:[0]:[0]:[48]:[8080.c800.0038]:[32]:[209.165.202.143]/272
                      10.1.1.74                          100          0 i
```

The following sample output displays peer VTEP device information.

```
switch # show nve peers

Interface Peer-IP    State LearnType Uptime   Router-Mac
--------- ---------------  ----- --------- -------- -----------------
nve1      10.1.1.56         Up    CP        1d12h    002a.6ab2.0181
nve1      10.1.1.74         Up    CP        1d12h    8c60.4f14.2efc
```

For IP host and prefix routes verification, refer these commands:

The following sample output displays tenant (VRF) information

```
switch # show ip arp vrf coke

Flags: * - Adjacencies learnt on non-active FHRP router
       + - Adjacencies synced via CFSoE
       # - Adjacencies Throttled for Glean
       D - Static Adjacencies attached to down interface

IP ARP Table for context coke
Total number of entries: 1
Address              Age       MAC Address       Interface
209.165.202.144  00:18:23       2010.0000.0010    Bdi200
```

The following sample output displays tenant (VRF) information

```
switch # show ip route vrf coke

IP Route Table for VRF "coke"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.1.1.0/24, ubest/mbest: 1/0, attached
    *via 10.1.1.1, Bdi10, [0/0], 1d12h, direct
10.1.1.1/32, ubest/mbest: 1/0, attached
```

```
    *via 10.1.1.1, Bdi10, [0/0], 1d12h, local
209.165.202.130/27, ubest/mbest: 1/0, attached
    *via 209.165.202.129, Bdi200, [0/0], 1d12h, direct, tag 12345,
209.165.202.129/32, ubest/mbest: 1/0, attached
    *via 209.165.202.129, Bdi200, [0/0], 1d12h, local, tag 12345,
209.165.202.139/32, ubest/mbest: 1/0, attached
    *via 209.165.202.139, Bdi200, [190/0], 1d12h, hmm
209.165.202.140 /32, ubest/mbest: 1/0
    *via 10.1.1.56%default, [200/0], 1d12h, bgp-100, internal, tag 100,  (mpls-vpn)segid
50000 tunnel: 16843064 encap: 1
```

The following sample output displays MAC - IP address binding for all attached and remote end hosts (learned through the BGP EVPN control plane).

```
switch # show l2route evpn mac-ip all

Topology ID Mac Address    Prod     Host IP           Next Hop(s)
----------- -------------- ---- --------------------------------------- --------
200         2010.0000.0010 HMM      209.165.202.139    N/A

200         2010.0000.0011 BGP      209.165.202.140    10.1.1.56

200         2010.0000.0012 BGP      209.165.202.141    10.1.1.74

200         2010.0000.0013 BGP      209.165.202.142    10.1.1.56

200         8080.c800.0038 BGP      209.165.202.143    10.1.1.74
```

The following sample output displays BGP routing table information for Layer-3 VNIs.

```
switch # show bgp l2vpn evpn

Route Distinguisher: 10.1.1.54:3    (L3VNI 50000)
*>i[2]:[0]:[0]:[48]:[2010.0000.0011]:[32]:[209.165.202.144]/272
                      10.1.1.56                        100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0012]:[32]:[209.165.202.141]/272
                      10.1.1.74                        100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0013]:[32]:[209.165.202.143]/272
                      10.1.1.56                        100          0 i
*>l[5]:[0]:[0]:[24]:[209.165.202.130]:[0.0.0.0]/224
                      10.1.1.54                 0        100      32768 ?
* i                   10.1.1.56                 0        100          0 ?
```

Cisco Nexus 9000 Series switch configuration

The following configurations are required for a Cisco Nexus 9000 Series switch for the VXLAN BGP EVPN fabric.

1. Initial configuration - Install the network virtualization overlay, BGP, and EVPN features on the VTEPs.
2. Layer 2 VNI configurations for tenant networks within a tenant.
3. Layer 3 VNI configurations for a tenant.

Note Though configuration examples are mainly IPv4, IPv6 addresses are also supported in the VXLAN BGP EVPN fabric.

Initial configuration

(config) #

```
nv overlay evpn
```

```
feature bgp
feature ospf
OR
feature isis
```

Attention You can use either OSPF or IS-IS as the underlay routing protocol.

(config) #

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

Configure the anycast gateway MAC address

(config) #

```
fabric forwarding anycast-gateway-mac 2020.0000.00aa
```

Configure BGP L2VPN EVPN address family

(config) #

```
router bgp 100
  neighbor 192.0.2.1
  remote-as 100
  update-source loopback0
  address-family l2vpn evpn
    send-community
    send-community extended
```

Layer 2 VNI configurations for a tenant network

Associate a VLAN to the Layer 2 VNI

(config) #

```
vlan 200
  vn-segment 30000
```

Create a loopback interface for BGP and assign an IP address to it

(config) #

```
interface loopback 0
   ip address 192.0.2.10/32
```

Create a loopback interface for NVE and assign an IP address to it

(config) #

```
interface loopback 1
   ip address 198.51.100.1/32
```

Associate the Layer 2 VNI to the overlay and configure multicast group membership

(config) #

```
interface nve 1
  no shutdown
  source-interface loopback1
  host-reachability protocol bgp
```

```
  member vni 30000
    suppress-arp
    mcast-group 239.1.1.0
```

Associate the Layer 2 VNI to the EVPN address family, and enable route distinguisher and route target functions for the VNI

(config) #

```
evpn
  vni 30000 l2
    rd auto
    route-target import auto
    route-target export auto
```

Note Alternatively, the following configurations can also be used:

```
evpn
  vni 30000 l2
    rd auto
    route-target both auto
```

The combination of the **router BGP** command (configured earlier) and the **evpn** command ensures that BGP EVPN is configured to advertise 'MAC route' or 'MAC + associated host routes' of servers attached to the VTEP, for the specified Layer 2 VNI. (Route type 2 [Refer to the EVPN RFC document for more details]). By default, the MAC route will be advertised, and the associated host route will be advertised if there is an SVI configured for that VLAN in the anycast-gateway mode or if suppress-arp option is enabled for that L2 VNI (see *ARP Suppression* section).

In the above NVE example, MAC and IP routes are advertised into BGP-EVPN for end hosts belonging to layer 2 VNI 30000.

Layer 3 VNI configurations for a tenant

Associate the VRF VNI (Layer 3 VNI) to the customer VRF

Enable VRF route distinguisher and route target functions

(config) #

```
vrf context coke:vrf1
  vni 50000
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
```

In the above example, the option *both* is used to import and export routes associated with the Layer 3 VNI 50000.

Associate the VRF VNI to a VLAN and associate an SVI to the customer VRF

(config) #

```
vlan 2500
  vn-segment 50000
```

(config) #

```
interface vlan 2500
  vrf member coke:vrf1
```

```
  ip forward
  ipv6 forward
  no ip redirects
  no ipv6 redirects
  no shutdown
```

In order to avoid the overhead of creating a core facing VLAN and corresponding SVI on a per VRF basis, the vrf-tenant-profile (that automatically takes care of this) is provided. If you configure a **vrf-tenant-profile**, you should enable the following CLIs related to dynamic and core VLANs.

(config) #

```
system fabric dynamic-vlans 2500-3500
system fabric core-vlans 2500-2999


configure profile vrf-tenant-profile
   vlan $vrfVlanId
     vn-segment $vrfSegmentId
   interface vlan $vrfVlanId
     vrf member $vrfName
     ip forward
     ipv6 forward
     no ip redirects
     no ipv6 redirects
     no shutdown
end
```

Add the Layer 3 VRF VNI to the overlay network

(config) #

```
interface nve 1
   host-reachability protocol bgp
   member vni 50000 associate-vrf
```

Associate the customer VRF to BGP and enable L2VPN EVPN route distribution

(config) #

```
router bgp 100
  vrf coke:vrf1
    address-family ipv4 unicast
      advertise l2vpn evpn
```

Enable host/server facing SVI (and associate it to a VRF) for Layer 3 connectivity on the distributed anycast gateway

(config) #

```
interface vlan 200
  vrf member coke:vrf1
  ip address 203.0.113.3/24 tag 12345
  fabric forwarding mode anycast-gateway
```

Cisco Nexus 9000 Series switches verification

For verification of MAC routes, refer these commands

The following is sample output to verify that end host MAC addresses (local and remote) are added to the MAC address table:

```
switch# show mac address-table dynamic

Legend:
```

```
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link,
        (T) - True, (F) - False, C - ControlPlane MAC

   VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
*    1     a036.9f22.a277   dynamic  0         F      F    Eth1/7
C  200     002a.6a85.a67c   dynamic  0         F      F    nve1(198.51.100.10)
*  200     2010.0000.0012   dynamic  0         F      F    Eth1/7
C  200     2010.0000.0015   dynamic  0         F      F    nve1(198.51.100.10)
```

The following is sample output for viewing MAC addresses of end hosts across all EVPN instances (EVIs) pertaining to the switch:

```
switch# show l2route evpn mac all

Topology    Mac Address    Prod   Flags         Seq No     Next-Hops
----------- -------------- ------ ------------- ---------- ----------------
200         002a.6a85.a67c BGP    SplRcv        0          198.51.100.10
200         2010.0000.0012 Local  L,            0          Eth1/7
200         2010.0000.0015 BGP    SplRcv        0          198.51.100.10
2500        7c0e.ceca.f2ff VXLAN  Rmac          0          198.51.100.10
```

Command output description

Prod (producer) column displays the source of origination of the MAC address.

Local means a MAC address learnt locally via a server facing or edge port, BGP means the remote end host MAC was learnt from a remote VTEP via BGP-EVPN and VXLAN indicates the router MAC of the remote VTEP as carried in the extended community in the BGP advertisement.

The following sample output displays BGP routing table information for the L2VPN EVPN address family. It includes route distinguisher and next hop information:

```
switch # show bgp l2vpn evpn

BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 26, local router ID is 192.0.2.10

Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist, I-i
njected
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup

   Network            Next Hop            Metric     LocPrf     Weight Path
Route Distinguisher: 192.0.2.20:3
*>i[5]:[0]:[0]:[24]:[203.0.113.6]:[0.0.0.0]/224
                     198.51.100.10         0       100          0 ?

Route Distinguisher: 192.0.2.20:32967
*>i[2]:[0]:[0]:[48]:[002a.6a85.a67c]:[0]:[0.0.0.0]/216
                     198.51.100.10                         100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0015]:[0]:[0.0.0.0]/216
                     198.51.100.10                         100          0 i
*>i[2]:[0]:[0]:[48]:[002a.6a85.a67c]:[32]:[200.0.0.52]/272
                     198.51.100.10                         100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0015]:[32]:[200.0.0.15]/272
                     198.51.100.10                         100          0 i

Route Distinguisher: 192.0.2.30:3
*>i[5]:[0]:[0]:[24]:[200.0.0.0]:[0.0.0.0]/224
                     198.51.100.10                0        100          0 ?

Route Distinguisher: 192.0.2.30:32967
*>i[2]:[0]:[0]:[48]:[002a.6a85.a67c]:[0]:[0.0.0.0]/216
                     198.51.100.10                         100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0015]:[0]:[0.0.0.0]/216
                     198.51.100.10                         100          0 i
*>i[2]:[0]:[0]:[48]:[002a.6a85.a67c]:[32]:[200.0.0.52]/272
                     198.51.100.10                         100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0015]:[32]:[200.0.0.15]/272
```

```
                     198.51.100.10                          100          0 i

Route Distinguisher: 192.0.2.10:32967    (L2VNI 30000)
* i[2]:[0]:[0]:[48]:[002a.6a85.a67c]:[0]:[0.0.0.0]/216
                     198.51.100.10                          100          0 i
*>i                  198.51.100.10                          100          0 i
*>l[2]:[0]:[0]:[48]:[2010.0000.0012]:[0]:[0.0.0.0]/216
                     192.0.2.10                             100      32768 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0015]:[0]:[0.0.0.0]/216
                     198.51.100.10                          100          0 i
* i                  198.51.100.10                          100          0 i
*>i[2]:[0]:[0]:[48]:[002a.6a85.a67c]:[32]:[203.0.113.12]/272
                     198.51.100.10                          100          0 i
* i                  198.51.100.10                          100          0 i
*>l[2]:[0]:[0]:[48]:[2010.0000.0012]:[32]:[203.0.113.5]/272
                     192.0.2.10                         100      32768 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0015]:[32]:[203.0.113.8]/272
                     198.51.100.10                          100          0 i
* i                  198.51.100.10                          100          0 i

Route Distinguisher: 192.0.2.10:3    (L3VNI 50000)
*>i[2]:[0]:[0]:[48]:[002a.6a85.a67c]:[32]:[203.0.113.12]/272
                     198.51.100.10                          100          0 i
* i                  198.51.100.10                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0015]:[32]:[203.0.113.8]/272
                     198.51.100.10                          100          0 i
* i                  198.51.100.10                          100          0 i
* i[5]:[0]:[0]:[24]:[200.0.0.0]:[0.0.0.0]/224
                     198.51.100.10                 0        100          0 ?
* i                  198.51.100.10                 0        100          0 ?
*>l                  198.51.100.1                  0        100      32768 ?
```

The following sample output displays peer VTEP device information:

```
switch # show nve peers

Interface Peer-IP          State LearnType Uptime   Router-Mac
--------- ---------------  ----- --------- -------- -----------------
nve1      198.51.100.10    Up    CP        3d00h    7c0e.ceca.f2ff
```

For verification of IP host and prefix routes, refer these commands

The following sample output displays tenant (VRF) information:

```
switch # show ip arp vrf coke:vrf1

Flags: * - Adjacencies learnt on non-active FHRP router
       + - Adjacencies synced via CFSoE
       # - Adjacencies Throttled for Glean
       CP - Added via L2RIB, Control plane Adjacencies
       PS - Added via L2RIB, Peer Sync
       RO - Dervied from L2RIB Peer Sync Entry
       D - Static Adjacencies attached to down interface

IP ARP Table for context coke:vrf1
Total number of entries: 1
Address         Age       MAC Address     Interface       Flags
203.0.113.5     00:12:46  2010.0000.0012  Vlan200
```

The following sample output displays tenant (VRF) information:

```
switch # show ip route vrf coke:vrf1

IP Route Table for VRF "coke:vrf1"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

203.0.113.6/24, ubest/mbest: 1/0, attached
    *via 203.0.113.3, Vlan200, [0/0], 3d00h, direct, tag 12345
203.0.113.3/32, ubest/mbest: 1/0, attached
    *via 203.0.113.3, Vlan200, [0/0], 3d00h, local, tag 12345
```

```
203.0.113.5/32, ubest/mbest: 1/0, attached
    *via 203.0.113.5, Vlan200, [190/0], 00:14:04, hmm
203.0.113.8/32, ubest/mbest: 1/0
    *via 198.51.100.10%default, [200/0], 3d00h, bgp-100, internal, tag 100 (evpn) segid:
50000 tunnelid: 0x16020202 encap: VXLAN

203.0.113.12/32, ubest/mbest: 1/0
    *via 198.51.100.10%default, [200/0], 00:13:46, bgp-100, internal, tag 100 (evpn) segid:
 50000 tunnelid: 0x16020202 encap: VXLAN
```

Command output description

Direct means that the subnet prefix is configured locally under a Layer-3 interface on this switch. Local means the IP address belongs to the switch aka locally configured under a Layer-3 interface on that switch (200.0.0.1/24).

The following sample output displays MAC - IP address binding for all attached and remote end hosts (learned through the BGP EVPN control plane):

```
switch # show l2route evpn mac-ip all

Topology    Mac Address    Prod   Flags         Seq No     Host IP         Next-Hops
----------- -------------- ------ ---------- --------------- ---------------
200         2010.0000.0012 HMM    --            0          203.0.113.5     Local
200         2010.0000.0015 BGP    --            0          203.0.113.8     198.51.100.10
200         002a.6a85.a67c BGP    --            0          203.0.113.12    198.51.100.10
```

ARP Suppression

BGP-EVPN distributes MAC and host IP information for hosts below a VTEP. Remote VTEPs can use this information to learn about other hosts and thereby suppress ARP requests early by proxying on behalf of the destination. All the IP-MAC binding aka ARP information learnt either about local end hosts or remote end hosts is shown into an ARP suppression cache. This early ARP termination functionality is enabled on a per Layer 2 VNI basis via a configuration knob (specifically **suppress-arp**). The detailed description of how the suppress-arp function works was described in the **ARP Suppression** section (Chapter 2 - Introducing VXLAN/EVPN). Here we cover the configuration and related show CLIs. This functionality is identical on Cisco Nexus 5000, 7000, 9000 Series switches.

ARP suppression at the VTEP level

(config) #

```
interface nve 1
  source-interface loopback 1
  host-reachability protocol bgp
  member vni 30000
     mcast-group 239.1.1.0
     suppress-arp
```

ARP suppression verification

The following sample output displays ARP suppression information in the cache:

```
switch# show ip arp suppression-cache detail

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

Ip Address      Age      Mac Address    Vlan Physical-ifindex    Flags    Remote Vtep Addrs
```

```
203.0.113.5     00:16:01 2010.0000.0012  200 Ethernet1/7         L
203.0.113.12    00:34:28 002a.6a85.a67c  200 (null)              R        198.51.100.10
203.0.113.8     3d00h    2010.0000.0015  200 (null)              R        198.51.100.10
```

The following sample output displays ARP suppression information for a VLAN, in the cache memory:

```
switch# show ip arp suppression-cache vlan 200

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

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

203.0.113.5    00:17:19 2010.0000.0012  200  Ethernet1/7       L
203.0.113.12   00:35:46 002a.6a85.a67c  200  (null)            R        198.51.100.10
203.0.113.8    3d00h    2010.0000.0015  200  (null)            R        198.51.100.10
```

The following sample output displays ARP suppression cache statistics information:

```
switch# show ip arp suppression-cache statistics

ARP packet statistics for suppression-cache
Suppressed:
Total 0, Requests 0, Requests on L2 0, Gratuitous 0, Gratuitous on L2 0

Forwarded :
Total: 0
 L3 mode :      Requests 0, Replies 0
                Request on core port 0, Reply on core port 0
                Dropped 0
 L2 mode :      Requests 0, Replies 0
                Request on core port 0, Reply on core port 0
                Dropped 0

Received:
Total: 3
 L3 mode:       Requests 3, Replies 0
                Local Request 3, Local Responses 0
                Gratuitous 0, Dropped 0
 L2 mode :      Requests 0, Replies 0
                Gratuitous 0, Dropped 0
ARP suppression-cache Local entry statistics
Adds 3, Deletes 0
```

Unknown unicast (packet) suppression

Configuration example for implementing the *suppress unknown unicast* function on a leaf/ToR switch

(config) #

```
interface nve 1
   member vni 30000
     suppress-unknown-unicast
```

FCoE

FCoE over the VXLAN fabric is not supported. However, FCoE and VXLAN can co-exist. FCoE and VXLAN services are provided on separate ports

To enable FCoE, use separate links from the fabric to MDS and connect to the target device. Refer *Cisco NX-OS FCoE Configuration Guide for Nexus 7000 Series and MDS 9000* and *Cisco Nexus 5600 Series NX-OS Fibre Channel over Ethernet Configuration Guide* for details.

DHCP

Manual and auto configurations of DHCP/DHCPv6 server and client functions on the default VRF, management VRF and non default VRF are given below.

DHCPv4 (clients in the non-default vrf)

With Auto Configuration—Supported scenarios

DHCP server and DHCP client in the same vlan/L2VNI but no DHCP relay on the leaf switch (This results in a Layer-2 flood within the same VLAN/L2 VNI).

DHCP server in the management VRF with the DHCP relay on the leaf switch under an Switch Virtual Interface (SVI) / Bridge Domain Interface (BDI).

Note An SVI is applicable to Cisco Nexus 5600 Series switches and a BDI to Cisco Nexus 7000 Series switches.

DHCP server in the default VRF with DHCP relay on the leaf switch under an SVI/BDI.

Without Auto Configuration—Supported scenarios

DHCP server and DHCP client in the same vlan/L2VNI but no DHCP relay on the leaf switch (so Layer-2 flood within the same VLAN/L2VNI).

DHCP server in the management VRF with DHCP relay on the leaf switch under an SVI/BDI.

DHCP server in the default VRF with the DHCP relay on the leaf switch under an SVI/BDI.

DHCP server and client in a non default VRF with a DHCP relay on the leaf switch under an SVI/BDI.

- For this scenario, you must enable the **advertise-pip** command on the leaf switch (for vPC scenarios).

DHCP server and client in different non default VRFs with a DHCP relay on the leaf switch under an SVI/BDI.

- For this scenario, you must enable the **advertise-pip** command on the leaf switch (for vPC scenarios).

DHCPv6 (clients in the non-default vrf)

Without Auto Configuration—Supported scenarios

DHCPv6 server and DHCPv6 client in the same vlan/L2VNI but no DHCPv6 relay on the leaf switch (so Layer-2 flood within the same VLAN/L2VNI).

DHCP server in the management VRF with DHCP relay on the leaf switch under an SVI/BDI.

DHCP configuration in a vPC setup

When DHCP or DHCPv6 relay function is configured on leaf switches in a vPC setup, and the DHCP server is in the non default, non management VRF, then configure the **advertise-pip** command on the vPC leaf switches. This allows BGP EVPN to advertise Route-type 5 routes with the next-hop using the primary IP address of the VTEP interface

A sample configuration is given below:

(config) #

```
router bgp 100
  address-family l2vpn evpn
    advertise-pip
```

vPC and FEX

The following vPC and FEX (the fabric is extended using Cisco Nexus 2000 Series Fabric Extender device) scenarios are supported:

Cisco Nexus 5600 Series Switches

- FEX, no vPC
- vPC with Active-Active FEX
- vPC with Straight Through FEX
- eVPC or 2-layer vPC with FEX

Refer the FEX link for Cisco Nexus 5600 Series switches. This has the topology and configurations for FEX AA, FEX ST, and FEX ST with vPC scenarios.

Cisco Nexus 7000 Series Switches

- Straight Through FEX connected to the leaf switch.
- vPC with straight through FEX.

In the VXLAN EVPN fabric, the Cisco Nexus 7000 Series switches supports integration with Fabric Extender (FEX) devices.

For detailed conceptual and configuration information for FEX support, refer to *Cisco Nexus 2000 Series Fabric Extender Software Configuration Guide for Cisco Nexus 7000 Series Switches, Release 7.x.*

For detailed vPC information, refer to *Design and Configuration Guide: Best Practices for Virtual Port Channels (vPC) on Cisco Nexus 7000 Series Switches.*

Attention In the VXLAN EVPN fabric, Nexus 7000 Series switches do not support FEX in Active-Active mode.

Cisco Nexus 9000 Series Switches

- Straight Through FEX connected to the leaf switch.
- vPC with Straight Through FEX.

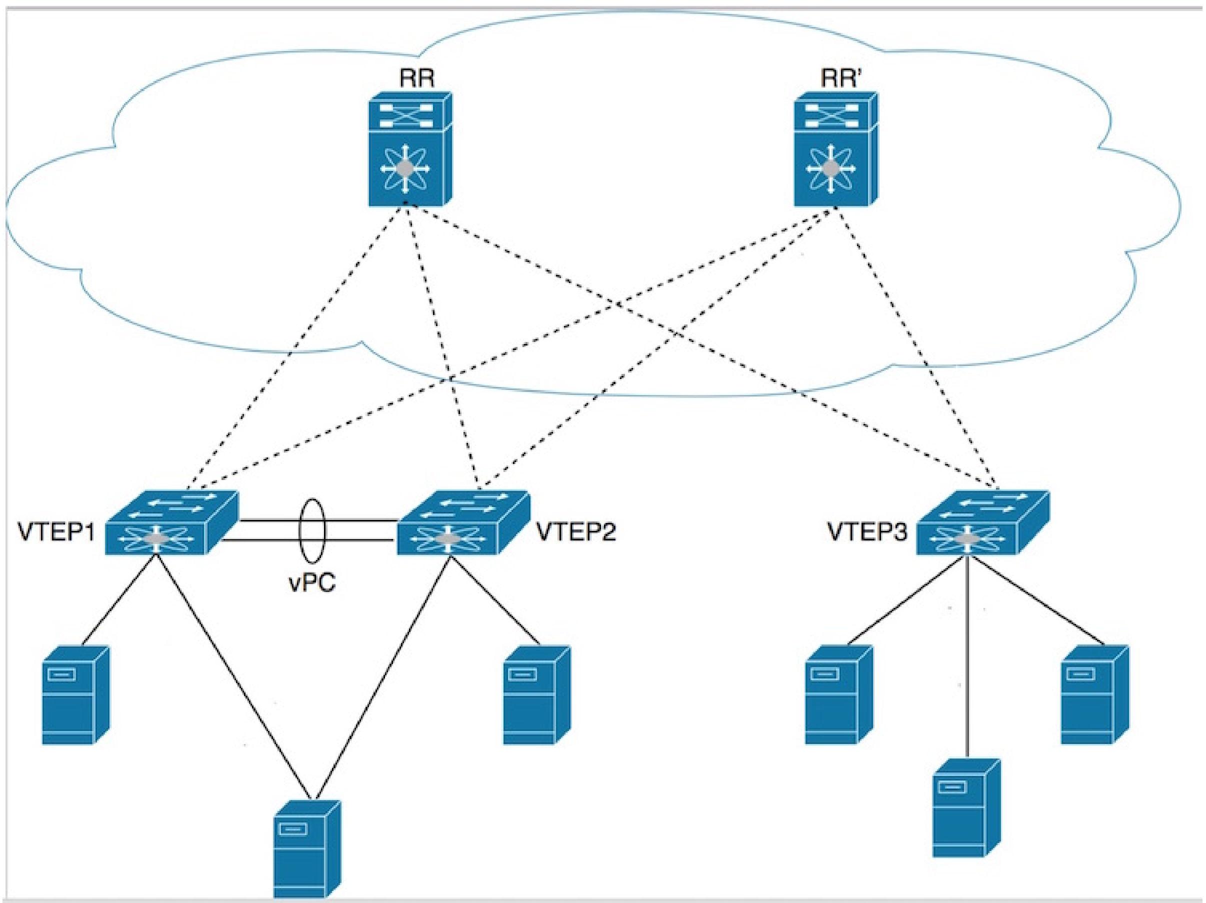
In the VXLAN EVPN fabric, the Cisco Nexus 9000 Series switches supports integration with Fabric Extender (FEX) devices.

For detailed conceptual and configuration information for FEX support, refer to *Cisco Nexus 2000 Series NX-OS Fabric Extender Configuration Guide for Cisco Nexus 9000 Series Switches, Release 7.x*

Attention In the VXLAN EVPN fabric, Nexus 9000 Series switches do not support FEX in Active-Active mode.

vPC configuration for Cisco Nexus 5600 Series Switches

Figure 1: vPC configuration



In the above topology, VTEP 1 and VTEP 2 are ToR switches and vPC peers. Sample vPC configurations are given below. For comprehensive information on vPC, refer to the respective Cisco Nexus Series 5600 and 7000 Series vPC design/configuration guide.

Attention Many of the configurations mentioned below need to be configured identically on the primary and secondary vPC peer switches, noted as **vPC-primary and vPC-secondary peer switches**. Configurations that are different between the peer switches are explicitly mentioned as **vPC-primary peer switch** and **vPC-secondary peer switch**.

Configure the vPC features

(config) #

```
feature lacp
feature vpc
```

Create a vPC domain

vPC-primary peer switch

(config) #

```
vpc domain 100
   peer-keepalive destination 10.1.1.156 source 10.1.1.154
   delay restore 150
   auto-recovery
   ip arp synchronize
   ipv6 nd synchronize
```

vPC-secondary peer switch

(config) #

```
vpc domain 100
  peer-keepalive destination 10.1.1.154 source 10.1.1.156
  delay restore 150
  auto-recovery
  ip arp synchronize
  ipv6 nd synchronize
```

Configure the secondary IP address on the loopback. This will be used as the virtual IP address (vIP) for both vPC peers

The secondary IP address of the source VTEP interface of the fabric (say, VTEP1/VTEP2 as source and VTEP 3 as destination) will be used as the source IP address in the VxLAN outer IP header. In a vPC scenario when EVPN is enabled, EVPN advertises the secondary IP address as the next hop address in the BGP update message. This is true for all route types including MAC routes, host IP routes, prefix routes etc. This is different from VXLAN flood-n-learn operation where for orphan ports the VXLAN outer IP header is set to the physical Peer IP or PIP when traffic ingresses in from the orphan ports and the VIP is only used when traffic ingresses in from the vPC ports.

vPC-primary peer switch

(config) #

```
interface loopback1
  ip address 10.1.2.54/32
  ip address 192.0.2.110/32 secondary

interface nve 1
   source-interface loopback0
   host-reachability protocol bgp
```

vPC-secondary peer switch

(config) #

```
interface loopback1
  ip address 10.1.2.56/32
  ip address 192.0.2.110/32 secondary

interface nve 1
   source-interface loopback1
   host-reachability protocol bgp
```

Note that the secondary IP address configured on the vPC primary and vPC secondary peer switches is the same.

Create the peer-link port-channel

vPC-primary and vPC-secondary peer switches

(config) #

```
interface port-channel 10
  description "vpc-peer-link"
  switchport mode trunk
  spanning-tree port type network
  vpc peer-link
```

Configure the peer-link interface

vPC-primary and vPC-secondary peer switches

(config) #

```
interface Ethernet1/1
  switchport mode trunk
  channel-group 10 mode active
```

Configure the peer link VLAN and routing between the vPC peer switches

Note The **vpc nve peer-link-vlan** command needs to be used only in the Cisco Nexus 5600 Series switches. Cisco Nexus 5600 Series switches encapsulate VXLAN packets over the MCT port with the configured VLAN as the outer-vlan tag while Cisco Nexus 7000,7700,9000 Series switches decapsulate VXLAN packets coming from the core and the decapsulated packet is bridged across the MCT link since they use ASM/SSM protocols.

You can use IS-IS or OSPF as the routing protocol between the vPC peer switches, as mentioned below:

IS-IS

vPC-primary peer switch

(config) #

```
vlan 123
interface Vlan123
  no shutdown
  ip address 38.38.38.54/24
  isis metric 10 level-1
  ip router isis PEER-LINK
  ip pim sparse-mode

vpc nve peer-link-vlan 123
```

vPC-secondary peer switch

(config) #

```
vlan 123
interface Vlan123
  no shutdown
  ip address 38.38.38.56/24
  isis metric 10 level-1
  ip router isis PEER-LINK
  ip pim sparse-mode
```

```
vpc nve peer-link-vlan 123
```

OSPF

vPC-primary peer switch

(config) #

```
vlan 123
interface vlan123
 no shutdown
 no ip redirects
 ip address 38.38.38.54/24
 ip ospf cost 10
 ip router ospf PEER-LINK area 0.0.0.0
 ip pim sparse-mode

vpc nve peer-link-vlan 123
```

vPC-secondary peer switch

(config) #

```
vlan 123
interface vlan123
 no shutdown
 no ip redirects
 ip address 38.38.38.56/24
 ip ospf cost 10
 ip router ospf PEER-LINK area 0.0.0.0
 ip pim sparse-mode

vpc nve peer-link-vlan 123
```

Configure the vPC host interface

From the image, you can see that an end host is (dual) attached to the peer switches. You need to configure the peer switches on the same port channel to enable end host dual attachment.

vPC-primary and vPC-secondary peer switches

(config) #

```
interface Ethernet1/5
  switchport mode trunk
  channel-group 35

interface port-channel 35
  switchport mode trunk
  spanning-tree port type edge trunk
```

Exclude the peer link VLAN from server facing ports

vPC-primary and vPC-secondary peer switches

(config) #

```
interface port-channel 35
  switchport trunk allowed vlan except 123

interface e1/5
  switchport trunk allowed vlan except 123
```

BUM (Layer-2 multicast) traffic behavior in VXLAN EVPN environments is identical to that in VXLAN flood and learn environments. For additional information on VXLAN flood and learn, refer to the respective Cisco Nexus Series 5600 or 7000/7700 VXLAN configuration guide.

Scenarios for advertising the primary IP address as the BGP next hop address in a vPC setup

In certain scenarios in a vPC setup (involving ToR leaf or border leaf switches) in the VXLAN EVPN fabric, you need to enable BGP EVPN to advertise Route-type 5 routes with the next-hop using the primary IP address of the VTEP interface. Recall that this is different from the default behavior where the vPC VIP associated with the VTEP interface is used as the next-hop for all advertised routes (Route-types 2/3/5). The scenarios are:

- The leaf switch and its vPC peer have asymmetric external Layer-3 connections that some IP prefix routes are only reachable from one of the leaf switches, and not from both of them.

  For example, when a pair of border leaf switches that run in vPC mode, and are connected to DCI boxes asymmetrically. (A symmetric topology can become asymmetric due to link failure.)

- A DHCP or DHCPv6 relay is configured on the leaf switch and the DHCP server is in the non default, non management VRF.
- Traffic is enabled between the vPC leaf switch and a remote end host.

  For example, to initiate a ping from the leaf switch's loopback address in a non default VRF to a remote end host.

There are 3 types of Layer-3 routes that can be advertised by BGP EVPN. They are:

1. Local host routes - These routes are leant from the attached servers or hosts.
2. Prefix routes - These routes are learnt via other routing protocol at the leaf, border leaf and border spine switches.
3. Leaf switch generated routes - These routes include interface routes and static routes.

On a vPC enabled leaf or border leaf switch, by default all Layer-3 routes are advertised with the secondary IP address (VIP) of the leaf switch VTEP as the BGP next-hop IP address. Prefix routes and leaf switch generated routes are not synced between vPC leaf switches. Using the VIP as the BGP next-hop for these types of routes can cause traffic to be forwarded to the wrong vPC leaf or border leaf switch and black-holed. The provision to use the primary IP address (PIP) as the next-hop when advertising prefix routes or loopback interface routes in BGP on vPC enabled leaf or border leaf switches allows users to select the PIP as BGP next-hop when advertising these types of routes, so that traffic will always be forwarded to the right vPC enabled leaf or border leaf switch.

The configuration command for advertising the PIP is **advertise-pip**.

A sample configuration is given below:

(config) #

```
router bgp 100
  address-family l2vpn evpn
    advertise-pip
    advertise-system-mac
```

The **advertise-pip** command lets BGP use the PIP as next-hop when advertising prefix routes or leaf generated routes if vPC is enabled. The **advertise-system-mac** command lets BGP advertise Route-type-2 routes that includes VIP and router-mac information. This is needed for solving an issue in EVPN decapsulation on remote leaf switches when PIP is used as next-hop in the BGP advertisement.
***Ping from a vPC switch when a PIP is not advertised*—**If you ping from switch A in a vPC setup (comprising of switches A and B) to a connected device or a remote end host, the common, virtual IP address (VIP) is considered the source IP address, and a successful response to the ping will be sent either to A, or to B. If the response is sent to B, then A (the sender) will not receive it.

As a workaround, create a loopback interface with a unique IP address for each vPC switch, and use the loopback IP address as the source for pinging attached devices or end hosts. Also leak the unique address between the vPC pair to ensure that the (ICMP) response is routed back to the sending vPC switch.

Also, you can use the VXLAN OAM functionality as a workaround.

Verify vPC configuration

For verification of MAC routes, refer these commands:

```
vPC-primary peer switch# show mac address-table dynamic

Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link
   VLAN     MAC Address      Type      age     Secure NTFY   Ports/SWID.SSID.LID
---------+-----------------+--------+---------+------+----+------------------
* 200      002a.6a44.9381    dynamic   1800       F    F  Po35
* 200      2010.0000.0010    dynamic   700        F    F  Eth100/1/1
+ 200      2010.0000.0011    dynamic   0          F    F  nve1/10.1.1.56
* 200      2010.0000.0012    dynamic   0          F    F  nve1/10.1.1.74
+ 200      2010.0000.0013    dynamic   0          F    F  nve1/10.1.1.56
* 123      002a.6ab2.0181    dynamic   0          F    F  Po10
* 1        a036.9f19.8ee4    dynamic   0          F    F  Po10
+ 1        a036.9f1a.b970    dynamic   0          F    F  Po10
* 1        a036.9f1a.c134    dynamic   0          F    F  Po10
* 1        a036.9f1a.c135    dynamic   120        F    F  Eth100/1/3
+ 1        a036.9f22.a30e    dynamic   0          F    F  Po10

vPC-secondary peer switch# show mac address-table dynamic

Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link
   VLAN     MAC Address      Type      age     Secure NTFY   Ports/SWID.SSID.LID
---------+-----------------+--------+---------+------+----+------------------
* 200      002a.6a44.9381    dynamic   300        F    F  Po35
* 200      2010.0000.0010    dynamic   30         F    F  Eth100/1/1
* 200      2010.0000.0011    dynamic   40         F    F  Eth101/1/1
* 200      2010.0000.0012    dynamic   0          F    F  nve1/10.1.1.74
* 200      2010.0000.0013    dynamic   20         F    F  Eth1/6
* 123      002a.6a6e.cbc1    dynamic   0          F    F  Po10
* 1        a036.9f19.8ee4    dynamic   0          F    F  Eth101/1/4
* 1        a036.9f1a.b970    dynamic   1770       F    F  Eth101/1/1
* 1        a036.9f1a.c134    dynamic   30         F    F  Eth101/1/3
* 1        a036.9f1a.c135    dynamic   110        F    F  Eth100/1/3

vPC-primary peer switch# show l2route evpn mac all

Topology    Mac Address    Prod   Next Hop (s)
----------- -------------- ------ ---------------
200         002a.6a44.9381 Local  Po35
200         2010.0000.0010 Local  Eth100/1/1
200         2010.0000.0011 Local  nve1/10.1.1.56
200         2010.0000.0012 BGP    10.1.1.74
200         2010.0000.0013 Local  nve1/10.1.1.56
2200        8c60.4f14.2efc VXLAN  10.1.1.74

vPC-secondary peer switch# show l2route evpn mac all

Topology    Mac Address    Prod   Next Hop (s)
----------- -------------- ------ ---------------
200         002a.6a44.9381 Local  Po35
200         2010.0000.0010 Local  Eth100/1/1
200         2010.0000.0011 Local  Eth101/1/1
200         2010.0000.0012 BGP    10.1.1.74
200         2010.0000.0013 Local  Eth1/6
2200        8c60.4f14.2efc VXLAN  10.1.1.74

vPC-primary peer switch# show bgp l2vpn evpn
```

```
BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 410, local router ID is 10.1.1.54
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist, I-i
njected
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup

   Network            Next Hop            Metric     LocPrf     Weight Path
Route Distinguisher: 10.1.1.54:32967    (L2VNI 30200)
*>l[2]:[0]:[0]:[48]:[002a.6a44.9381]:[0]:[0.0.0.0]/216
                      2.2.2.2                           100      32768 i
* i                   2.2.2.2                           100          0 i
*>l[2]:[0]:[0]:[48]:[2010.0000.0010]:[0]:[0.0.0.0]/216
                      2.2.2.2                           100      32768 i
* i                   2.2.2.2                           100          0 i
*>l[2]:[0]:[0]:[48]:[2010.0000.0011]:[0]:[0.0.0.0]/216
                      2.2.2.2                           100      32768 i
* i                   2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0012]:[0]:[0.0.0.0]/216
                      10.1.1.74                          100          0 i
*>l[2]:[0]:[0]:[48]:[2010.0000.0013]:[0]:[0.0.0.0]/216
                      2.2.2.2                           100      32768 i
* i                   2.2.2.2                           100          0 i
*>l[2]:[0]:[0]:[48]:[002a.6a44.9381]:[32]:[200.0.0.35]/272
                      2.2.2.2                           100      32768 i
* i                   2.2.2.2                           100          0 i
* i[2]:[0]:[0]:[48]:[2010.0000.0010]:[32]:[200.0.0.10]/272
                      2.2.2.2                           100          0 i
*>l                   2.2.2.2                           100      32768 i
*>l[2]:[0]:[0]:[48]:[2010.0000.0011]:[32]:[200.0.0.11]/272
                      2.2.2.2                           100      32768 i
* i                   2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0012]:[32]:[200.0.0.12]/272
                      10.1.1.74                          100          0 i
*>l[2]:[0]:[0]:[48]:[2010.0000.0013]:[32]:[200.0.0.13]/272
                      2.2.2.2                           100      32768 i
* i                   2.2.2.2                           100          0 i

Route Distinguisher: 10.1.1.56:3
*>i[5]:[0]:[0]:[24]:[200.0.0.0]:[0.0.0.0]/224
                      2.2.2.2                  0        100          0 ?

Route Distinguisher: 10.1.1.56:32967
*>i[2]:[0]:[0]:[48]:[002a.6a44.9381]:[0]:[0.0.0.0]/216
                      2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0010]:[0]:[0.0.0.0]/216
                      2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0011]:[0]:[0.0.0.0]/216
                      2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0013]:[0]:[0.0.0.0]/216
                      2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[002a.6a44.9381]:[32]:[200.0.0.35]/272
                      2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0010]:[32]:[200.0.0.10]/272
                      2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0011]:[32]:[200.0.0.11]/272
                      2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0013]:[32]:[200.0.0.13]/272
                      2.2.2.2                           100          0 i

Route Distinguisher: 10.1.1.74:32967
*>i[2]:[0]:[0]:[48]:[2010.0000.0012]:[0]:[0.0.0.0]/216
                      10.1.1.74                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0012]:[32]:[200.0.0.12]/272
                      10.1.1.74                          100          0 i

vPC-secondary peer switch# show bgp l2vpn evpn

BGP routing table information for VRF default, address family L2VPN EVPN
BGP table version is 308, local router ID is 10.1.1.56
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist, I-i
njected
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup

   Network            Next Hop            Metric     LocPrf     Weight Path

Route Distinguisher: 10.1.1.54:3
*>i[5]:[0]:[0]:[24]:[200.0.0.0]:[0.0.0.0]/224
                      2.2.2.2                  0        100          0 ?
Route Distinguisher: 10.1.1.54:32967
*>i[2]:[0]:[0]:[48]:[002a.6a44.9381]:[0]:[0.0.0.0]/216
                      2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0010]:[0]:[0.0.0.0]/216
                      2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0011]:[0]:[0.0.0.0]/216
                      2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0013]:[0]:[0.0.0.0]/216
                      2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[002a.6a44.9381]:[32]:[200.0.0.35]/272
                      2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0010]:[32]:[200.0.0.10]/272
                      2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0011]:[32]:[200.0.0.11]/272
                      2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0013]:[32]:[200.0.0.13]/272
                      2.2.2.2                           100          0 i

Route Distinguisher: 10.1.1.56:32967    (L2VNI 30200)
* i[2]:[0]:[0]:[48]:[002a.6a44.9381]:[0]:[0.0.0.0]/216
                      2.2.2.2                           100          0 i
*>l                   2.2.2.2                           100      32768 i
* i[2]:[0]:[0]:[48]:[2010.0000.0010]:[0]:[0.0.0.0]/216
                      2.2.2.2                           100          0 i
*>l                   2.2.2.2                           100      32768 i
* i[2]:[0]:[0]:[48]:[2010.0000.0011]:[0]:[0.0.0.0]/216
                      2.2.2.2                           100          0 i
*>l                   2.2.2.2                           100      32768 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0012]:[0]:[0.0.0.0]/216
                      10.1.1.74                          100          0 i
* i[2]:[0]:[0]:[48]:[2010.0000.0013]:[0]:[0.0.0.0]/216
                      2.2.2.2                           100          0 i
*>l                   2.2.2.2                           100      32768 i
* i[2]:[0]:[0]:[48]:[002a.6a44.9381]:[32]:[200.0.0.35]/272
                      2.2.2.2                           100          0 i
*>l                   2.2.2.2                           100      32768 i
*>l[2]:[0]:[0]:[48]:[2010.0000.0010]:[32]:[200.0.0.10]/272
                      2.2.2.2                           100      32768 i
* i                   2.2.2.2                           100          0 i
* i[2]:[0]:[0]:[48]:[2010.0000.0011]:[32]:[200.0.0.11]/272

*>i[2]:[0]:[0]:[48]:[2010.0000.0012]:[32]:[200.0.0.12]/272
                      10.1.1.74                          100          0 i
* i[2]:[0]:[0]:[48]:[2010.0000.0013]:[32]:[200.0.0.13]/272
                      2.2.2.2                           100          0 i
*>l                   2.2.2.2                           100      32768 i

Route Distinguisher: 10.1.1.74:32967
*>i[2]:[0]:[0]:[48]:[2010.0000.0012]:[0]:[0.0.0.0]/216
                      10.1.1.74                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0012]:[32]:[200.0.0.12]/272
                      10.1.1.74                          100          0 i

vPC-primary peer switch# show nve peers

Interface Peer-IP          State LearnType Uptime   Router-Mac
--------- ---------------  ----- --------- -------- -----------------
nve1      10.1.1.56         Up    CP        01:41:24 n/a
nve1      10.1.1.74         Up    CP        01:41:19 8c60.4f14.2efc

vPC-secondary peer switch# show nve peers

Interface Peer-IP          State LearnType Uptime   Router-Mac
--------- ---------------  ----- --------- -------- -----------------
nve1      10.1.1.54         Up    CP        1d01h    n/a
```

```
nve1      10.1.1.74         Up    CP        4d09h    8c60.4f14.2efc
```

For verification of IP host and prefix routes, refer these commands

```
vPC-primary peer switch# show ip arp vrf all

Flags: * - Adjacencies learnt on non-active FHRP router
       + - Adjacencies synced via CFSoE
       # - Adjacencies Throttled for Glean
       D - Static Adjacencies attached to down interface

IP ARP Table for all contexts
Total number of entries: 8
Address         Age       MAC Address     Interface
10.1.1.156      00:13:04  002a.6ab2.0141  mgmt0
10.1.1.233      00:00:23  0050.569f.6c61  mgmt0
1.1.1.53        00:12:51  002a.6a85.a5bc  Ethernet1/24
38.38.38.56     00:02:55  002a.6ab2.0181  Vlan123
200.0.0.10      00:09:02  2010.0000.0010  Vlan200
200.0.0.11      00:06:37  2010.0000.0011  Vlan200          +
200.0.0.13      00:06:34  2010.0000.0013  Vlan200          +
200.0.0.35      00:00:28  002a.6a44.9381  Vlan200          +

vPC-secondary peer switch# show ip arp vrf all

Flags: * - Adjacencies learnt on non-active FHRP router
       + - Adjacencies synced via CFSoE
       # - Adjacencies Throttled for Glean
       D - Static Adjacencies attached to down interface

IP ARP Table for all contexts
Total number of entries: 8
Address         Age       MAC Address     Interface
10.1.1.154      00:13:11  002a.6a6e.cb81  mgmt0
10.1.1.233      00:00:30  0050.569f.6c61  mgmt0
1.1.1.53        00:04:27  002a.6a85.a5bc  Ethernet1/26
38.38.38.54     00:03:03  002a.6a6e.cbc1  Vlan123
200.0.0.10      00:09:09  2010.0000.0010  Vlan200          +
200.0.0.11      00:06:45  2010.0000.0011  Vlan200
200.0.0.13      00:06:41  2010.0000.0013  Vlan200
200.0.0.35      00:00:36  002a.6a44.9381  Vlan200

vPC-primary peer switch# show ip route vrf all

IP Route Table for VRF "default"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

1.1.1.53/32, ubest/mbest: 1/0
    *via 1.1.1.53, Eth1/24, [110/5], 16:40:48, ospf-UNDERLAY, intra
10.1.1.54/32, ubest/mbest: 2/0, attached
    *via 10.1.1.54, Lo0, [0/0], 01:59:11, local
    *via 10.1.1.54, Lo0, [0/0], 01:59:11, direct
10.1.1.56/32, ubest/mbest: 1/0
    *via 1.1.1.53, Eth1/24, [110/9], 16:40:42, ospf-UNDERLAY, intra
10.1.1.74/32, ubest/mbest: 1/0
    *via 1.1.1.53, Eth1/24, [110/9], 16:40:39, ospf-UNDERLAY, intra
2.2.2.2/32, ubest/mbest: 2/0, attached
    *via 2.2.2.2, Lo0, [0/0], 01:59:11, local
    *via 2.2.2.2, Lo0, [0/0], 01:59:11, direct
10.254.254.2/32, ubest/mbest: 1/0
    *via 1.1.1.53, Eth1/24, [110/5], 16:40:00, ospf-UNDERLAY, intra
10.254.254.66/32, ubest/mbest: 1/0
    *via 1.1.1.53, Eth1/24, [110/5], 16:39:55, ospf-UNDERLAY, intra
38.38.38.0/24, ubest/mbest: 1/0, attached
    *via 38.38.38.54, Vlan123, [0/0], 01:59:00, direct
38.38.38.54/32, ubest/mbest: 1/0, attached
    *via 38.38.38.54, Vlan123, [0/0], 01:59:00, local

IP Route Table for VRF "management"
'*' denotes best ucast next-hop
```

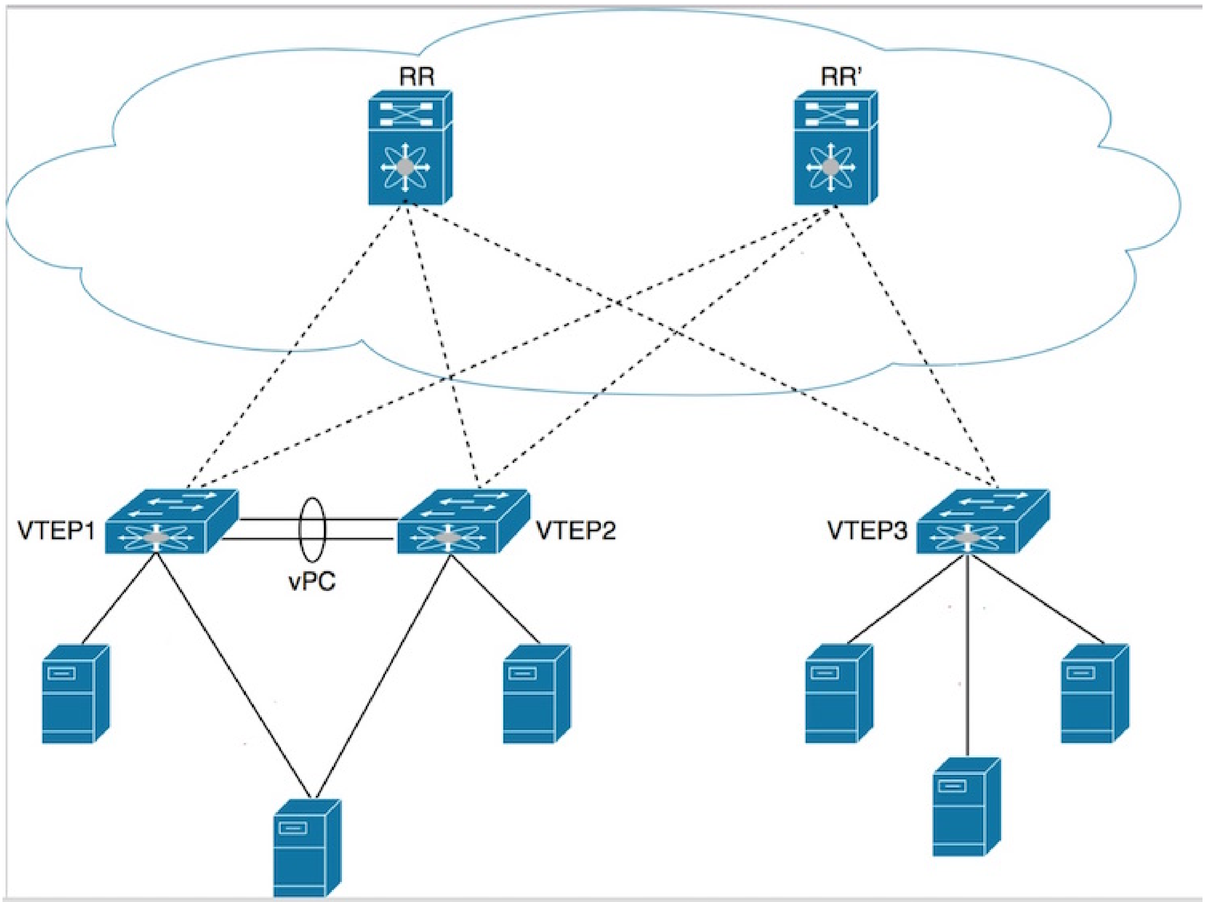
```
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

0.0.0.0/0, ubest/mbest: 1/0
    *via 10.1.1.233, [1/0], 4d08h, static
10.1.1.0/24, ubest/mbest: 1/0, attached
    *via 10.1.1.154, mgmt0, [0/0], 4d08h, direct
10.1.1.154/32, ubest/mbest: 1/0, attached
    *via 10.1.1.154, mgmt0, [0/0], 4d08h, local

IP Route Table for VRF "sml:vpn2200"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

200.0.0.0/24, ubest/mbest: 1/0, attached
    *via 200.0.0.1, Vlan200, [0/0], 01:59:00, direct, tag 12345,
200.0.0.1/32, ubest/mbest: 1/0, attached
    *via 200.0.0.1, Vlan200, [0/0], 01:59:00, local, tag 12345,
200.0.0.10/32, ubest/mbest: 1/0, attached
    *via 200.0.0.10, Vlan200, [190/0], 01:46:41, hmm
200.0.0.11/32, ubest/mbest: 1/0, attached
    *via 200.0.0.11, Vlan200, [190/0], 01:46:41, hmm
200.0.0.12/32, ubest/mbest: 1/0
    *via 10.1.1.74%default, [200/0], 01:59:04, bgp-100, internal, ta
-vpn)segid 32200 tunnel: 16843082 encap: 1

200.0.0.13/32, ubest/mbest: 1/0, attached
    *via 200.0.0.13, Vlan200, [190/0], 01:46:36, hmm
200.0.0.35/32, ubest/mbest: 1/0, attached
    *via 200.0.0.35, Vlan200, [190/0], 01:58:35, hmm
vPC-secondary peer switch# show ip route vrf all

IP Route Table for VRF "default"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

1.1.1.53/32, ubest/mbest: 1/0
    *via 1.1.1.53, Eth1/26, [110/5], 16:45:46, ospf-UNDERLAY, intra
10.1.1.54/32, ubest/mbest: 1/0
    *via 1.1.1.53, Eth1/26, [110/9], 02:04:15, ospf-UNDERLAY, intra
10.1.1.56/32, ubest/mbest: 2/0, attached
    *via 10.1.1.56, Lo0, [0/0], 4d10h, local
    *via 10.1.1.56, Lo0, [0/0], 4d10h, direct
10.1.1.74/32, ubest/mbest: 1/0
    *via 1.1.1.53, Eth1/26, [110/9], 16:45:43, ospf-UNDERLAY, intra
2.2.2.2/32, ubest/mbest: 2/0, attached
    *via 2.2.2.2, Lo0, [0/0], 4d10h, local
    *via 2.2.2.2, Lo0, [0/0], 4d10h, direct
10.254.254.2/32, ubest/mbest: 1/0
    *via 1.1.1.53, Eth1/26, [110/5], 16:45:04, ospf-UNDERLAY, intra
10.254.254.66/32, ubest/mbest: 1/0
    *via 1.1.1.53, Eth1/26, [110/5], 16:44:59, ospf-UNDERLAY, intra
38.38.38.0/24, ubest/mbest: 1/0, attached
    *via 38.38.38.56, Vlan123, [0/0], 02:04:13, direct
38.38.38.56/32, ubest/mbest: 1/0, attached
    *via 38.38.38.56, Vlan123, [0/0], 02:04:13, local

IP Route Table for VRF "management"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

0.0.0.0/0, ubest/mbest: 1/0
    *via 10.1.1.233, [1/0], 4d10h, static
10.1.1.0/24, ubest/mbest: 1/0, attached
    *via 10.1.1.156, mgmt0, [0/0], 4d10h, direct
10.1.1.156/32, ubest/mbest: 1/0, attached
```

```
    *via 10.1.1.156, mgmt0, [0/0], 4d10h, local

IP Route Table for VRF "sml:vpn2200"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

200.0.0.0/24, ubest/mbest: 1/0, attached
    *via 200.0.0.1, Vlan200, [0/0], 4d10h, direct, tag 12345,
200.0.0.1/32, ubest/mbest: 1/0, attached
    *via 200.0.0.1, Vlan200, [0/0], 4d10h, local, tag 12345,
200.0.0.10/32, ubest/mbest: 1/0, attached
    *via 200.0.0.10, Vlan200, [190/0], 01:51:46, hmm
200.0.0.11/32, ubest/mbest: 1/0, attached
    *via 200.0.0.11, Vlan200, [190/0], 01:51:46, hmm
200.0.0.12/32, ubest/mbest: 1/0
    *via 10.1.1.74%default, [200/0], 02:07:28, bgp-100, internal, tag 100,  (mpls
-vpn)segid 32200 tunnel: 16843082 encap: 1

200.0.0.13/32, ubest/mbest: 1/0, attached
    *via 200.0.0.13, Vlan200, [190/0], 01:51:40, hmm
200.0.0.35/32, ubest/mbest: 1/0, attached
    *via 200.0.0.35, Vlan200, [190/0], 02:03:39, hmm
vPC-primary peer switch# show l2route evpn mac-ip all

Topology ID Mac Address    Prod Host IP                                 Next Hop
 (s)
----------- -------------- ---- --------------------------------------- --------
200         002a.6a44.9381 HMM  200.0.0.35                              N/A

200         2010.0000.0012 BGP  200.0.0.12                              10.1.1.74

vPC-secondary peer switch# show l2route evpn mac-ip all

n6k-56-poap# show l2route evpn mac-ip all
Topology ID Mac Address    Prod Host IP                                 Next Hop
 (s)
----------- -------------- ---- --------------------------------------- --------
200         002a.6a44.9381 HMM  200.0.0.35                              N/A

200         2010.0000.0012 BGP  200.0.0.12                              10.1.1.74

vPC-primary peer switch# show bgp l2vpn evpn

Route Distinguisher: 10.1.1.54:3    (L3VNI 32200)
*>i[2]:[0]:[0]:[48]:[002a.6a44.9381]:[32]:[200.0.0.35]/272
                      2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0010]:[32]:[200.0.0.10]/272
                      2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0011]:[32]:[200.0.0.11]/272
                      2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0012]:[32]:[200.0.0.12]/272
                      10.1.1.74                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0013]:[32]:[200.0.0.13]/272
                      2.2.2.2                           100          0 i
*>l[5]:[0]:[0]:[24]:[200.0.0.0]:[0.0.0.0]/224
                      2.2.2.2                 0         100      32768 ?
* i                   2.2.2.2                 0         100          0 ?

vPC-secondary peer switch# show bgp l2vpn evpn

Route Distinguisher: 10.1.1.56:3    (L3VNI 32200)
*>i[2]:[0]:[0]:[48]:[002a.6a44.9381]:[32]:[200.0.0.35]/272
                      2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0010]:[32]:[200.0.0.10]/272
                      2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0011]:[32]:[200.0.0.11]/272
                      2.2.2.2                           100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0012]:[32]:[200.0.0.12]/272
                      10.1.1.74                          100          0 i
*>i[2]:[0]:[0]:[48]:[2010.0000.0013]:[32]:[200.0.0.13]/272
                      2.2.2.2                           100          0 i
```

```
* i[5]:[0]:[0]:[24]:[200.0.0.0]:[0.0.0.0]/224
                      2.2.2.2                  0        100          0 ?
*>l                   2.2.2.2                  0        100      32768 ?
```

vPC configuration for Cisco Nexus 9000 Series Switches

Figure 2: vPC configuration



Configure the vPC features

(config) #

```
feature vpc
```

Create a vPC domain

vPC-primary peer switch

(config) #

```
vpc domain 1
  peer-switch
  peer-keepalive destination 192.0.2.30 source 192.0.2.20
  delay restore 150
  peer-gateway
  auto-recovery
```

```
  ip arp synchronize
  ipv6 nd synchronize
```

vPC-secondary peer switch

(config) #

```
vpc domain 1
 peer-switch
 peer-keepalive destination 192.0.2.30 source 192.0.2.20
 delay restore 150 peer-gateway
 auto-recovery
 ip arp synchronize
 ipv6 nd synchronize
```

Configure the secondary IP address on the loopback. This will be used as the virtual IP address (vIP) for both vPC peers

The secondary IP address of the source VTEP interface on vPC leaf switches will be used as the source IP address in the VXLAN outer IP header. In a vPC scenario when EVPN is enabled, EVPN advertises the secondary IP address as the next hop address in the BGP update message. This is true for all route types including MAC routes, host IP routes, prefix routes, etc.

vPC-primary peer switch

(config) #

```
interface loopback1
  ip address 192.0.2.40/32
  ip address 198.51.100.10/32 secondary
  ip router isis UNDERLAY
  ip pim sparse-mode

interface nve 1
  source-interface loopback1
  host-reachability protocol bgp
```

vPC-secondary peer switch

(config) #

```
interface loopback1
  ip address 192.0.2.41/32
  ip address 198.51.100.10/32 secondary
  ip router isis UNDERLAY
  ip pim sparse-mode

interface nve 1
  source-interface loopback1
  host-reachability protocol bgp
```

Note that the secondary IP address configured on the vPC primary and vPC secondary peer switches is the same.

Create the peer-link port-channel

vPC-primary and vPC-secondary peer switches

(config) #

```
interface port-channel 10
  description "vpc-peer-link"
  switchport mode trunk
  spanning-tree port type network
  vpc peer-link
```

Configure the peer-link interface

vPC-primary and vPC-secondary peer switches

(config) #

```
interface Ethernet1/1
  description “vpc-peer-link”
  switchport mode trunk
  channel-group 10 mode active
```

Configure the backup VLAN path between vPC peer switches

Note To provide a backup path when a vPC switch loses connectivity to the spine, at least one SVI is required to be configured across the peer-link, so that traffic can be forwarded to this vPC switch from its vPC peer switch over the peer-link.

You can use IS-IS or OSPF as the routing protocol between the vPC peer switches, as mentioned below:

IS-IS

vPC-primary peer switch

(config) #

```
vlan 123
interface Vlan123
  no shutdown
  ip address 192.0.2.100/24
  ip router isis UNDERLAY
  ip pim sparse-mode
  no ip redirects
  no ipv6 redirects

system nve infra-vlan 123
```

vPC-secondary peer switch

(config) #

```
vlan 123
interface Vlan123
  no shutdown
  ip address 192.0.2.101/24
  ip router isis UNDERLAY
  ip pim sparse-mode
  no ip redirects
  no ipv6 redirects

system nve infra-vlan 123
```

OSPF

vPC-primary peer switch

(config) #

```
vlan 123
interface Vlan123
  no shutdown
  ip address 192.0.2.100/24
  ip router ospf UNDERLAY area 0.0.0.0
  ip ospf network point-to-point
  ip pim sparse-mode
```

```
  no ip redirects
  no ipv6 redirects

system nve infra-vlan 123
```

vPC-secondary peer switch

(config) #

```
vlan 123 interface vlan123
  no shutdown
  ip address 192.0.2.100/24
  ip router ospf UNDERLAY area 0.0.0.0
  ip ospf network point-to-point
  ip pim sparse-mode
  no ip redirects
  no ipv6 redirects

system nve infra-vlan 123
```

Configure the vPC host interface

As shown in the figure, an end host is (dual) attached to both vPC peer switches. Same port channel must be configured on both switches to enable end host dual attachment.

vPC-primary and vPC-secondary peer switches

(config) #

```
interface Ethernet1/2
  switchport mode trunk
  channel-group 52

interface port-channel 52
  switchport mode trunk
  vpc 52
```

PLB

Pervasive Load Balancing for the Programmable Fabric

In a programmable fabric, the servers, the virtual machines (VMs), and the containers (specific to a given service) can be distributed across the fabric, attached to different ToR/leaf switches. The Pervasive Load Balancing (PLB) feature enables load balancing to the servers that are distributed across the fabric.

In the load balancing function, a virtual IP (VIP) abstracts a service provided by a physical server farm distributed across the DC fabric. When different clients (local to fabric or from a remote location) send requests for a given service, these requests are always destined to the VIP of these servers.

On the ToR/leaf switches, PLB matches the source IP address bits/mask, the destination IP address (Virtual IP address), and relevant Layer 3/Layer 4 fields to load balance these requests among the servers.

PLB provides an infrastructure to configure a cluster of the servers (nodes) inside a device group and segregates the client traffic based on the buckets (bit mask) and the tenant SVI configured under the PLB service. Based on the defined cluster of nodes (servers) and buckets, PLB automatically creates PBR rules to match the client IP traffic into the buckets mask and redirect the matched traffic to a specific server node.

PLB provides the infrastructure to configure cluster of servers' nodes inside a device group and segregates client traffic based on the buckets (bit mask) and tenant SVI configured under PLB service. Based on the defined cluster of nodes (servers) and buckets. PLB automatically creates PBR rules to match client IP traffic into buckets mask and redirect matched traffic to specific server node.

PLB also provides the infrastructure to periodically monitor health of all server nodes and status of its application services like TCP/UDP/DNS on a given VRF.

In case, if server become non-responsive or non-operational then it provides resiliency by atomically switching the client traffic from destined non-operational node to configured standby node/s. Traffic assignment is achieved by automatically changing PBR rules to standby node/s.

PLB is fabric agnostic but currently supported with VXLAN EVPN Fabric.

PLB currently uses Direct Server Return (DSR) concept and functionality that means responses from servers directly goes to the client.

A high-level overview of Pervasive Load Balancing on the ToR switch is given below:

- Load balancing servers are identified and grouped into a device group.
- A PLB service instance is created (for the group), and the following associations are made:
    - A virtual IP address (VIP) that is created for incoming PLB traffic directed at the servers. The VIP represents the servers in the device group.
    - Other load balancing configurations are enabled.