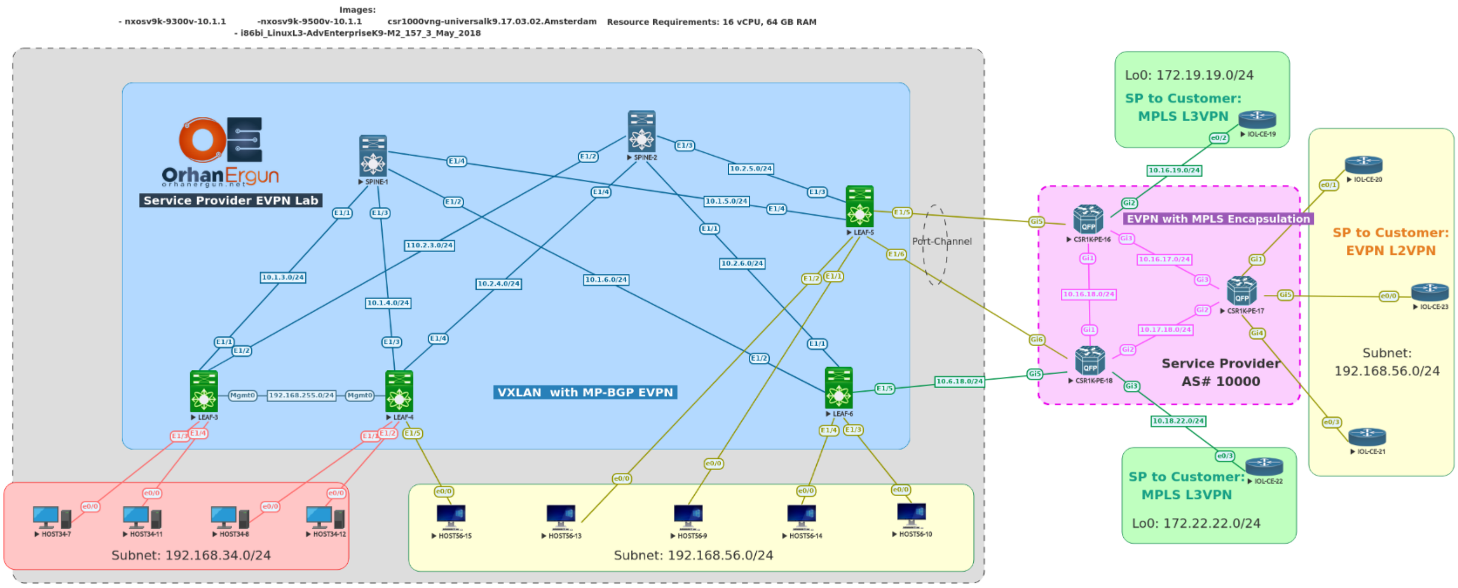
VXLAN, MPLS, BGP EVPN Lab

Topology:



BGP AF/SAF: L2VPN EVPN

Encapsulations:

- VXLAN (Single-Homed)
- MPLS (Multi-Homed)

Task 01 and 02:

- Configure SPINE switches, these switches must be Route Reflectors (Only enable L2VPN EVPN for BGP neighbors)
- Configure LEAF Switches (Only enable L2VPN EVPN for BGP neighbors)
- Enable VXLAN BGP-EVPN fabric
- For HOST34-x use L2 VNI 34
- For HOST56-x use L2 VNI 56
- L3 VNI for IRB: 1000
- BUM traffic should be forwarded using Multicast (Not Ingress Replication)
- L3 IP addresses of host should be also learned

Solution:

First of all, we are going to discuss about different encapsulation types that we use in Ethernet VPN (EVPN), EVPN by itself is going to be used as the control plane protocol in order to Discovery of LEAF or PE devices, Advertise MAC/IP of the end devices, Aliasing, etc… . we will discuss about all this thing in this lab.

There are two major types of encapsulation when it comes to BGP EVPN, the first one is using NVO (Network Virtualization Overlay) which is VXLAN in this case and the other one is MPLS.

There are some differences in EVPN information when it comes to EVPN routes detailed information. In this lab we will try to walk through all these differences which is necessary for CCIE SP exam.

The configuration of VXLAN EVPN on NX-OS devices is not necessary for CCIE SP exam, but in order to get familiar with EVPN in total, how it acts when it comes to VXLAN or MPLS is important.

Let's start with EVPN using the VXLAN as an encapsulation.

We mentioned that, BGP EVPN is going to be used as the Control Plane, it is being used as Auto Discovery (Leaf or PE devices are going to find out about each other), MAC/IP advertisement (Leaf or PE devices can find some information about end nodes such as IP address and MAC address of them and advertise to other PE or Leaf devices, so when it is necessary they can use those information in order to send the frames to the remote leaf/PE device), BUM traffic forwarding (Broadcast, Unknown Unicast and Multicast traffic can be sent to the remote leaf or PE devices when it is needed), Aliasing (Doing Load-Sharing between two remote devices that are connected to the same Ethernet Segment) and Provide Multi-Homing in order to provide Redundant links to an end host or a CE device.

And the VXLAN or MPLS are the encapsulation types that are taking the resposibility of the Data Plane. The original frame or packet is encapsulated using these encapsulation methods.

Let's start the configuration and discuss about all of them in action.

In a Data Center when it comes to VXLAN BGP EVPN, we have two kinds of devices, The SPINE and LEAF.

The SPINE devices are placed in the core of the network and the LEAF devices are connected to those SPINE devices. In this lab we have four LEAF devices which are connected to two SPINE devices in a redundant manner.

We need to configure SPINE devices to be the route-reflectors of MP-BGP EVPN (They will get the EVPN routes from Leaves of advertise them to other leaves.

NOTE: SPINEs are not directly connected to each other and they are not BGP neighbors with each other.

First of all we need to enable some features on the SPINEs NX-OS, because, in NX-OS features are not enabled by default unlike IOS-XE or IOS-XR.

So features like : OSPF, BGP EVPN, PIM needs to be enabled:

```
SPINE-1 and SPINE-2:

nv overlay evpn

feature ospf

feature bgp

feature pim
```

nv overlay evpn command is for enabling EVPN address family of BGP.

```
SPINE-1:

hostname SPINE-1


ip pim rp-address 10.255.255.100 group-list 225.0.0.0/8

ip pim anycast-rp 10.255.255.100 10.255.255.1

ip pim anycast-rp 10.255.255.100 10.255.255.2


interface Ethernet1/1

  mtu 9216

  ip address 10.1.3.1/24

  ip ospf network point-to-point

  no ip ospf passive-interface

  ip router ospf 65000 area 0.0.0.0

  ip pim sparse-mode

  no shutdown
```

```
interface Ethernet1/2
  mtu 9216
  ip address 10.1.6.1/24
  ip ospf network point-to-point
  no ip ospf passive-interface
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface Ethernet1/3
  mtu 9216
  ip address 10.1.4.1/24
  ip ospf network point-to-point
  no ip ospf passive-interface
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface Ethernet1/4
  mtu 9216
  ip address 10.1.5.1/24
  ip ospf network point-to-point
  no ip ospf passive-interface
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface loopback100
  description Multicast Anycast RP
  ip address 10.255.255.100/32
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode

interface loopback0
  description RID
  ip address 10.255.255.1/32
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode

router ospf 65000
  router-id 10.255.255.1
  log-adjacency-changes detail
  passive-interface default
router bgp 65000
  router-id 10.255.255.1
  address-family l2vpn evpn
  neighbor 10.255.255.3
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community
      send-community extended
      route-reflector-client
  neighbor 10.255.255.4
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community
      send-community extended
      route-reflector-client
  neighbor 10.255.255.5
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community
      send-community extended
      route-reflector-client
  neighbor 10.255.255.6
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community
      send-community extended
      route-reflector-client
```

All these configurations are just basic oned which we need to bring-up the underlay routing (OSPF) in our network, and configuring the multicast routing in the core of the network (Multicast is going to be used for BUM traffic forwarding between LEAF switches), later on when it comes to EVPN with MPLS encapsulation we will use Ingress Replication instead of Multicast Replication for BUM traffic forwarding (CSR1000v only supports Ingress Replication).

In this case we chose PIM Sparse mode and configured redundant RPs (Using Anycast RP).

The last step is the BGP configuration, we have defined four neighbors (four LEAF switches) and made them as route-reflector clients and also enabled L2VPN EVPN AF/SAF.

The configuration of the other SPINE switch is almost the same:

```
SPINE-2:
hostname SPINE-2

ip pim rp-address 10.255.255.100 group-list 225.0.0.0/8
ip pim anycast-rp 10.255.255.100 10.255.255.1
ip pim anycast-rp 10.255.255.100 10.255.255.2


interface Ethernet1/1
  mtu 9216
  ip address 10.2.6.2/24
  ip ospf network point-to-point
  no ip ospf passive-interface
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface Ethernet1/2
  mtu 9216
  ip address 10.2.3.2/24
  ip ospf network point-to-point
  no ip ospf passive-interface
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface Ethernet1/3
  mtu 9216
  ip address 10.2.5.2/24
  ip ospf network point-to-point
  no ip ospf passive-interface
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface Ethernet1/4
  mtu 9216
  ip address 10.2.4.2/24
  ip ospf network point-to-point
  no ip ospf passive-interface
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode
  no shutdown


interface loopback0
  description RID
  ip address 10.255.255.2/32
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode

interface loopback100
  description Multicast Anycast RP
  ip address 10.255.255.100/32
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode

router ospf 65000
  router-id 10.255.255.2
  log-adjacency-changes detail
  passive-interface default
router bgp 65000
  router-id 10.255.255.2
  address-family l2vpn evpn
  neighbor 10.255.255.3
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community
      send-community extended
      route-reflector-client
  neighbor 10.255.255.4
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community
      send-community extended
      route-reflector-client
  neighbor 10.255.255.5
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community
      send-community extended
      route-reflector-client
  neighbor 10.255.255.6
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community
      send-community extended
      route-reflector-client
```

Now everything is ready on SPINE switches, let’s configure LEAF switches also:

The same steps are needed to be done on the LEAF switches (OSPF, BGP EVPN and Multicast configuration):

```
LEAF-3:
hostname LEAF-3

nv overlay evpn
feature ospf
feature bgp
feature pim

ip pim rp-address 10.255.255.100 group-list 225.0.0.0/8

interface Ethernet1/1
  no switchport
  mtu 9216
  ip address 10.1.3.3/24
  ip ospf network point-to-point
  no ip ospf passive-interface
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface Ethernet1/2
  no switchport
  mtu 9216
  ip address 10.2.3.3/24
  ip ospf network point-to-point
  no ip ospf passive-interface
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode
  no shutdown


interface loopback0
  description RID_and_VTEP
  ip address 10.255.255.3/32
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode


router ospf 65000
  router-id 10.255.255.3
  log-adjacency-changes detail
  passive-interface default
router bgp 65000
  router-id 10.255.255.3
  address-family l2vpn evpn
  neighbor 10.255.255.1
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community
      send-community extended
  neighbor 10.255.255.2
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community
      send-community extended
      advertise l2vpn evpn
```

LEAF-4:

```
hostname LEAF-4


nv overlay evpn
feature ospf
feature bgp
feature pim


fabric forwarding anycast-gateway-mac 0000.dcdc.dcdc
ip pim rp-address 10.255.255.100 group-list 225.0.0.0/8


interface Ethernet1/3
  no switchport
  mtu 9216
  ip address 10.1.4.4/24
  ip ospf network point-to-point
  no ip ospf passive-interface
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode
  no shutdown


interface Ethernet1/4
  no switchport
  mtu 9216
  ip address 10.2.4.4/24
  ip ospf network point-to-point
  no ip ospf passive-interface
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode
  no shutdown



interface loopback0
  description RID_and_VTEP
  ip address 10.255.255.4/32
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode

router ospf 65000
  router-id 10.255.255.4
  log-adjacency-changes detail
  passive-interface default
router bgp 65000
  router-id 10.255.255.4
  address-family l2vpn evpn
  neighbor 10.255.255.1
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community
      send-community extended
  neighbor 10.255.255.2
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community
      send-community extended

LEAF-6:
hostname LEAF-6

nv overlay evpn
feature ospf
feature bgp
feature pim

ip pim rp-address 10.255.255.100 group-list 225.0.0.0/8

interface Ethernet1/1
```

```
  no switchport
  mtu 9216
  ip address 10.2.6.6/24
  ip ospf network point-to-point
  no ip ospf passive-interface
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface Ethernet1/2
  no switchport
  mtu 9216
  ip address 10.1.6.6/24
  ip ospf network point-to-point
  no ip ospf passive-interface
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface loopback0
  description RID_and_VTEP
  ip address 10.255.255.6/32
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode

router ospf 65000
  router-id 10.255.255.6
  log-adjacency-changes detail
  passive-interface default
router bgp 65000
  router-id 10.255.255.6
  address-family l2vpn evpn
  neighbor 10.255.255.1
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community
      send-community extended
  neighbor 10.255.255.2
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community
      send-community extended
```

LEAF-5:

```
hostname LEAF-5

nv overlay evpn
feature ospf
feature bgp
feature pim

ip pim rp-address 10.255.255.100 group-list 225.0.0.0/8

interface Ethernet1/3
  no switchport
  mtu 9216
  ip address 10.2.5.5/24
  ip ospf network point-to-point
  no ip ospf passive-interface
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

interface Ethernet1/4
  no switchport
  mtu 9216
  ip address 10.1.5.5/24
  ip ospf network point-to-point
  no ip ospf passive-interface
  ip router ospf 65000 area 0.0.0.0
  ip pim sparse-mode
  no shutdown

router ospf 65000
  router-id 10.255.255.5
  log-adjacency-changes detail
  passive-interface default
router bgp 65000
  router-id 10.255.255.5
  address-family l2vpn evpn
  neighbor 10.255.255.1
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community
      send-community extended
  neighbor 10.255.255.2
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community
      send-community extended
```

Now our underlay is ready.

This time we need to configure the overlay.

Again, some features needs to be enable on all LEAF switches, including VXLAN, SVI (VLAN Interfaces), and fabric forwarding which is VXLAN encapsulation for VXLAN EVPN fabric.

Let's configure them on all TOR (Top Of Rack) switches:

```
LEAF-3, LEAF-4, LEAF-5 and LEAF-6:

feature fabric forwarding

feature interface-vlan

feature vn-segment-vlan-based

feature nv overlay
```

Next step is to create local VLANs on LEAF switches and put the access ports in the forwarding of those VLANs.

NOTE: These VLANs are only localy significant to the LEAF switches, but they will be also part of a global VLAN! (which means VXLAN Network Segment), all these VLANs which are part of the same VXLAN Segment are going to be considered as being a part of a big global VLAN that is distributed between LEAF switches. For example HOST34-7 is in access mode of VLAN 3 and HOST34-8 is in access mode of VLAN 4, but both of those VLANs are part of the same VXLAN Segment (Segment 34), so in reality they are part of a same broadcast domain but spanned across tow different LEAF switches.

```
LEAF-3:

vlan 1,3

vlan 3

  vn-segment 34


LEAF-4:

vlan 1,4,456

vlan 4

  vn-segment 34

vlan 456

  vn-segment 56


LEAF-5:

vlan 1,5

vlan 5

  vn-segment 56


LEAF-6:

vlan 1,6

vlan 6

  vn-segment 56
```

Next step is to put the access ports in the forwarding state of a specific VLAN:

```
LEAF-3:
interface Ethernet1/3
  switchport access vlan 3

interface Ethernet1/4
  switchport access vlan 3

LEAF-4:
interface Ethernet1/1
  switchport access vlan 4

interface Ethernet1/2
  switchport access vlan 4

interface Ethernet1/5
  switchport access vlan 456

LEAF-5:
interface Ethernet1/1
  switchport access vlan 5

interface Ethernet1/2
  switchport access vlan 5

interface port-channel56
  switchport access vlan 5

interface Ethernet1/5
  switchport access vlan 5
  channel-group 56

interface Ethernet1/6
  switchport access vlan 5
  channel-group 56
```

```
LEAF-6:

interface Ethernet1/3

  switchport access vlan 6


interface Ethernet1/4

  switchport access vlan 6
```

Very simple steps so far!

Under the VLAN configuration we used the command: vn-segment 34 and vn-segment 56, these are the L2 VNIs (Layer 2 VXLAN Network Identifiers).

As an example, whenever HOST34-7 wants to send something to HOST34-12, the LEAF-3 switch will get that frame and encapsulates it inside a VXLAN datagram, and it will put the 34 as the VNI of VXLAN and sends it towards LEAF-4, LEAF-4 receives the packet, takes a look at the VNI and forwards it towards the destination host which is HOST34-12.

What about the L3 routing?

What if HOST34-7 sends something to HOST56-13 which is part of another VXLAN Segment (56)?

Routing should happen in this case, because they are in different Broadcast domains and they have different IP network ID (192.168.34.0/24 and 192.168.56.0/24).

This is where the IRB (Integrated Routing and Bridging) comes into play!

There are two types of IRB: Symmetric and Asymmetric.

Cisco Devices support Symmetric model.

The only thing that you need to configure on all of the LEAF switches is an L3VNI and associate the L2VNIs with it.

In the Asymmetric model you need to configure all L2 VNIs on all of the LEAF switches which is not scalable and overcomplicates the things!

This will simplify the procecss:

- HOST34-7 is part of Network Segment 34
- HOST56-13 is part of Network Segment 56
- Network Segments 34 and 56 are associated with L3VNI 1000
- When HOST34-7 want to send some packets to HOST56-13, it sends them to it's default gateway on LEAF-3, LEAF3 takes a look at the destination and realizes that the destination is on another network (192.168.56.0/24), LEAF-3 does the routing and encapsulates the packet in VXLAN and puts VNI 1000 inside the packet and sends it to

the LEAF-4, LEAF-4 gets the packet, takes a look at the VNI and sends it towards the destination host (HOST56-13).

That was all about how VXLAN EVPN works in a first glance!

The configuration is also simple:

We configure the SVIs for each of the VLANs, configure the Anycast Gateway on all SVIs, then enable the IP Forwarding on the L3 SVI:

```
LEAF-3:

fabric forwarding anycast-gateway-mac 0000.dcdc.dcdc

vlan 1000

vlan 1000

  vn-segment 1000

vrf context TEST-1000

  vni 1000

  rd auto

  address-family ipv4 unicast

    route-target both auto

    route-target both auto evpn


interface Vlan3

  no shutdown

  vrf member TEST-1000

  ip address 192.168.34.254/24

  fabric forwarding mode anycast-gateway


interface Vlan1000

  no shutdown

  vrf member TEST-1000

  ip forward

interface nve1

  no shutdown

  host-reachability protocol bgp

  source-interface loopback0

  member vni 34

    suppress-arp
    mcast-group 225.34.34.34
  member vni 1000 associate-vrf
evpn
  vni 34 l2
    rd auto
    route-target import auto
    route-target export auto
```

NX-OS has a good feature which simplifies the VXLAN EVPN configuration, that feature is Auto RD and Auto RT.

You don’t need to configure RD and RT for BGP and VRFs manually, just leave it to NX-OS, it will take care of assigning them!

Let’s also explain what those Green Highlighted commands mean:

- `fabric forwarding anycast-gateway-mac 0000.dcdc.dcdc`: MAC address configuration for Anycast Gateway
- `fabric forwarding mode anycast-gateway`: Enable the Anycast-Gateway forwarding mode for a specific VLAN
- `host-reachability protocol bgp`: Use MP-BGP EVPN for the control plane (MAC/IP) advertisement
- `suppress-arp`: Enbale Host IP Learning
- `mcast-group 225.34.34.34`: Use Multicast group of x.x.x.x for the BUM traffic multicast replication

```
LEAF-4:
fabric forwarding anycast-gateway-mac 0000.dcdc.dcdc
vlan 1000
vlan 1000
  vn-segment 1000
vrf context TEST-1000
  vni 1000
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
interface Vlan4
  no shutdown
  vrf member TEST-1000
  ip address 192.168.34.254/24
  fabric forwarding mode anycast-gateway
```

```
interface Vlan456
  no shutdown
  vrf member TEST-1000
  ip address 192.168.56.254/24
  fabric forwarding mode anycast-gateway

interface Vlan1000
  no shutdown
  vrf member TEST-1000
  ip forward
interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback0
  member vni 34
    suppress-arp
    mcast-group 225.34.34.34
  member vni 56
    suppress-arp
    mcast-group 225.56.56.56
  member vni 1000 associate-vrf
evpn
  vni 34 l2
    rd auto
    route-target import auto
    route-target export auto

LEAF-5:
fabric forwarding anycast-gateway-mac 0000.dcdc.dcdc
vlan 1000
vlan 1000
  vn-segment 1000
vrf context TEST-1000
  vni 1000
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
interface Vlan5
  no shutdown
  vrf member TEST-1000
  ip address 192.168.56.254/24
  fabric forwarding mode anycast-gateway

interface Vlan1000
  no shutdown
  vrf member TEST-1000
  ip forward

interface port-channel56
  switchport access vlan 5

interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback0
  member vni 56
    suppress-arp
    mcast-group 225.56.56.56
  member vni 1000 associate-vrf
evpn
  vni 56 l2
    rd auto
    route-target import auto
    route-target export auto
```

LEAF-6:

```
fabric forwarding anycast-gateway-mac 0000.dcdc.dcdc
vlan 1000
vlan 1000
```

```
vrf context TEST-1000
  vni 1000
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
interface Vlan6
  no shutdown
  vrf member TEST-1000
  ip address 192.168.56.254/24
  fabric forwarding mode anycast-gateway

interface Vlan1000
  no shutdown
  vrf member TEST-1000
  ip forward

interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback0
  member vni 56
    suppress-arp
    mcast-group 225.56.56.56
  member vni 1000 associate-vrf
interface Ethernet1/5
  description To-SP-MPLS
  no switchport
  vrf member TEST-1000
  ip address 10.6.18.6/24
  no shutdown
router bgp 65000
  vrf TEST-1000
    timers bgp 7 21
    address-family ipv4 unicast
    neighbor 10.6.18.18
      remote-as 10000
      address-family ipv4 unicast
evpn
  vni 56 l2
    rd auto
    route-target import auto
    route-target export auto
```

Those Green Highlighted commands are for connecting the LEAF-6 Ethernet1/5 interface to the Service Provider which is providing MPLS L3VPN service (In upcoming tasks we will configure it).

Verification:

Let's examine some show commands:

```
HOST34-7#ping 192.168.34.8
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.34.8, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 5/9/13 ms
HOST34-7#ping 192.168.34.12
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.34.12, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 4/4/7 ms

LEAF-3# show bgp l2vpn evpn vni-id 34 | begin Network
   Network            Next Hop            Metric     LocPrf     Weight Path
Route Distinguisher: 10.255.255.3:32770    (L2VNI 34)
*>l[2]:[0]:[0]:[48]:[aabb.cc00.0700]:[0]:[0.0.0.0]/216
                      10.255.255.3                      100      32768 i
*>i[2]:[0]:[0]:[48]:[aabb.cc00.0800]:[0]:[0.0.0.0]/216
                      10.255.255.4                      100          0 i
*>l[2]:[0]:[0]:[48]:[aabb.cc00.0b00]:[0]:[0.0.0.0]/216
                      10.255.255.3                      100      32768 i
```

```
*>i[2]:[0]:[0]:[48]:[aabb.cc00.0c00]:[0]:[0.0.0.0]/216
                      10.255.255.4                      100          0 i
*>l[2]:[0]:[0]:[48]:[aabb.cc00.0700]:[32]:[192.168.34.7]/272
                      10.255.255.3                      100      32768 i
*>i[2]:[0]:[0]:[48]:[aabb.cc00.0800]:[32]:[192.168.34.8]/272
                      10.255.255.4                      100          0 i
*>i[2]:[0]:[0]:[48]:[aabb.cc00.0c00]:[32]:[192.168.34.12]/272
                      10.255.255.4                      100          0 i



HOST34-8#show interface e0/0 | include bia
  Hardware is AmdP2, address is aabb.cc00.0800 (bia aabb.cc00.0800)
```

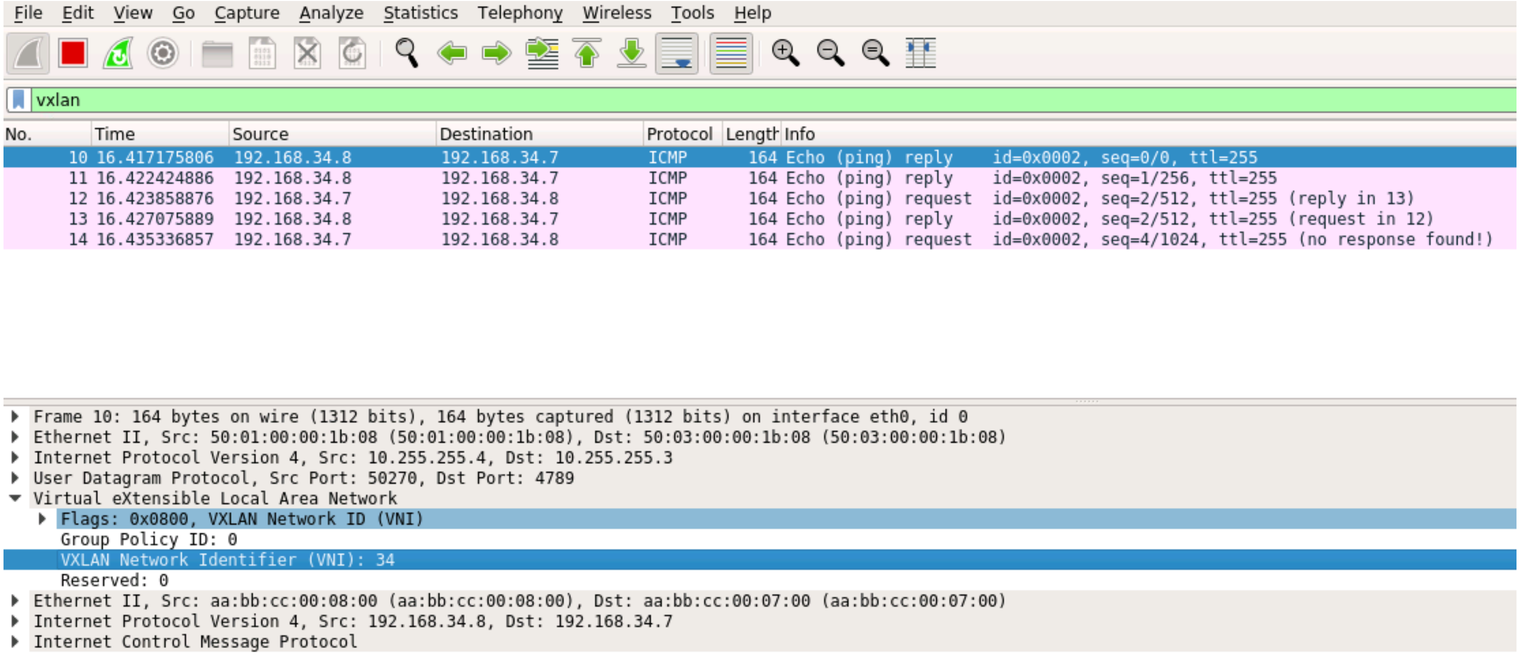
As you can realise from the above output, LEAF-3 know about the remote host MAC address thanks to the EVPN Route Type 2:

```
LEAF-3# show bgp l2vpn evpn aabb.cc00.0800
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 10.255.255.3:32770    (L2VNI 34)
BGP routing table entry for [2]:[0]:[0]:[48]:[aabb.cc00.0800]:[0]:[0.0.0.0]/216,
 version 13
Paths: (1 available, best #1)
Flags: (0x000212) (high32 00000000) on xmit-list, is in l2rib/evpn, is not in HW

  Advertised path-id 1
  Path type: internal, path is valid, is best path, no labeled nexthop, in rib
             Imported from 10.255.255.4:32771:[2]:[0]:[0]:[48]:[aabb.cc00.0800]:
[0]:[0.0.0.0]/216
  AS-Path: NONE, path sourced internal to AS
    10.255.255.4 (metric 81) from 10.255.255.1 (10.255.255.1)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 34
      Extcommunity: RT:65000:34 ENCAP:8
      Originator: 10.255.255.4 Cluster list: 10.255.255.1
  Path-id 1 not advertised to any peer

BGP routing table entry for [2]:[0]:[0]:[48]:[aabb.cc00.0800]:[32]:[192.168.34.8

]/272, version 31

Paths: (1 available, best #1)

Flags: (0x000212) (high32 00000000) on xmit-list, is in l2rib/evpn, is not in HW

--More--
```

in the output we can see:

```
10.255.255.4:32771:[2]:[0]:[0]:[48]:[aabb.cc00.0800] >>> 2: EVPN Route Type 2, 48: MAC length
Extcommunity: RT:65000:34 ENCAP:8  >>> ENCAP 8: VXLAN encapsulation
Received label 34  >>>  L2VNI 34
```

Packet Capture:



```
No.        Time          Source         Destination    Protocol Length Info
        10 16.417175806  192.168.34.8   192.168.34.7   ICMP        164 Echo (ping) reply    id=0x0002, seq=0/0, ttl=255
        11 16.422424886  192.168.34.8   192.168.34.7   ICMP        164 Echo (ping) reply    id=0x0002, seq=1/256, ttl=255
        12 16.423858876  192.168.34.7   192.168.34.8   ICMP        164 Echo (ping) request  id=0x0002, seq=2/512, ttl=255 (reply in 13)
        13 16.427075889  192.168.34.8   192.168.34.7   ICMP        164 Echo (ping) reply    id=0x0002, seq=2/512, ttl=255 (request in 12)
        14 16.435336857  192.168.34.7   192.168.34.8   ICMP        164 Echo (ping) request  id=0x0002, seq=4/1024, ttl=255 (no response found!)
```

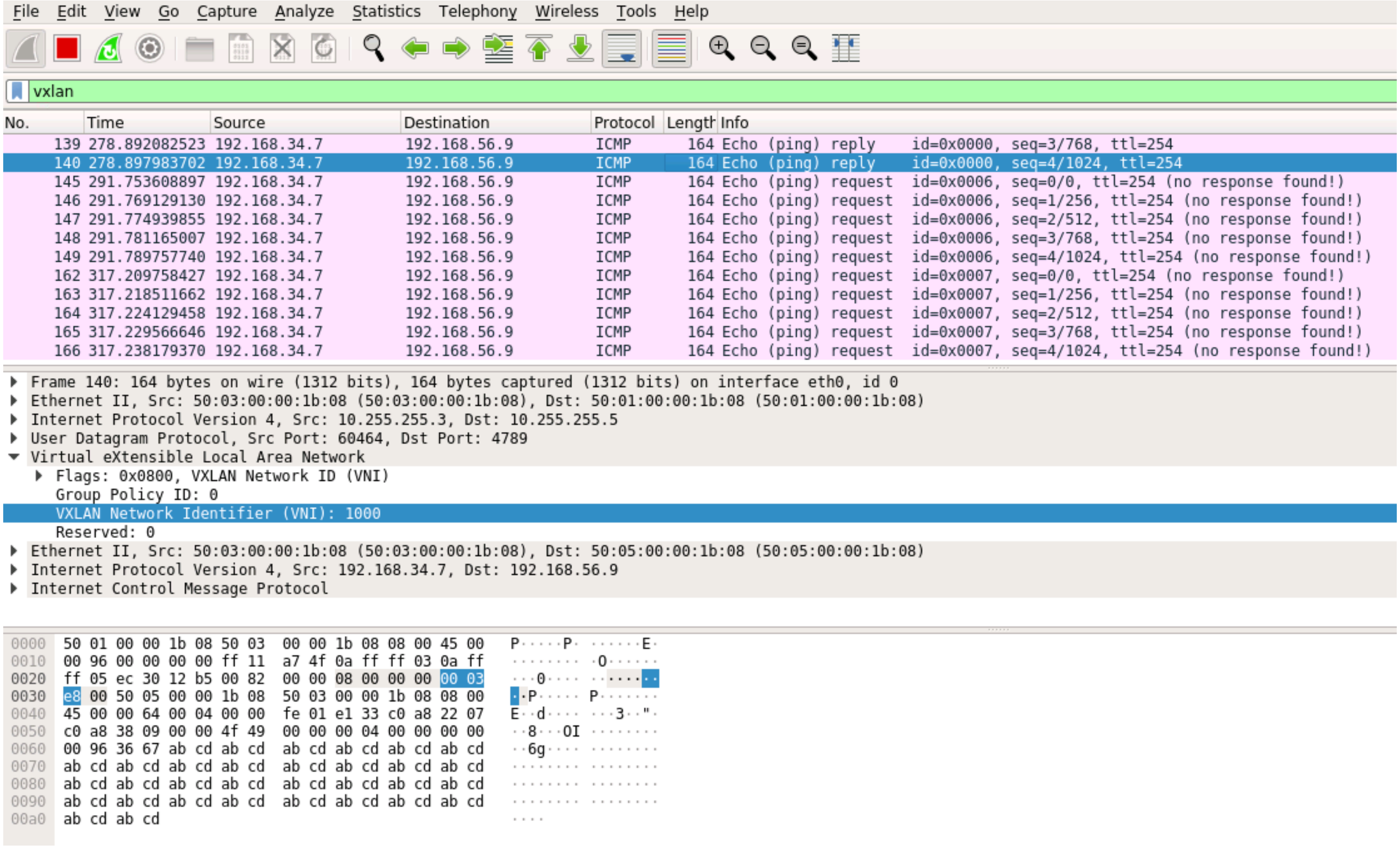
```
▸ Frame 10: 164 bytes on wire (1312 bits), 164 bytes captured (1312 bits) on interface eth0, id 0
▸ Ethernet II, Src: 50:01:00:00:1b:08 (50:01:00:00:1b:08), Dst: 50:03:00:00:1b:08 (50:03:00:00:1b:08)
▸ Internet Protocol Version 4, Src: 10.255.255.4, Dst: 10.255.255.3
▸ User Datagram Protocol, Src Port: 50270, Dst Port: 4789
▾ Virtual eXtensible Local Area Network
  ▸ Flags: 0x0800, VXLAN Network ID (VNI)
    Group Policy ID: 0
    VXLAN Network Identifier (VNI): 34
    Reserved: 0
▸ Ethernet II, Src: aa:bb:cc:00:08:00 (aa:bb:cc:00:08:00), Dst: aa:bb:cc:00:07:00 (aa:bb:cc:00:07:00)
▸ Internet Protocol Version 4, Src: 192.168.34.8, Dst: 192.168.34.7
▸ Internet Control Message Protocol
```

Take a look at the VXLAN header, there is VXLAN Network Identifier (l2VNI 34 in this case).

This time, we will ping HOST56-13 on HOST34-7:

```
HOST34-7#ping 192.168.56.9

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.56.9, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 6/8/17 ms
```

We expect to see VNI 1000 in the VXLAN header, because the packet is being routed from one subnet to another subnet (from 192.168.34.0/24 to 192.168.56.0/24):



```
vxlan
No.    Time          Source        Destination   Protocol Length Info
   139 278.892082523 192.168.34.7  192.168.56.9  ICMP     164 Echo (ping) reply    id=0x0000, seq=3/768, ttl=254
   140 278.897983702 192.168.34.7  192.168.56.9  ICMP     164 Echo (ping) reply    id=0x0000, seq=4/1024, ttl=254
   145 291.753608897 192.168.34.7  192.168.56.9  ICMP     164 Echo (ping) request  id=0x0006, seq=0/0, ttl=254 (no response found!)
   146 291.769129130 192.168.34.7  192.168.56.9  ICMP     164 Echo (ping) request  id=0x0006, seq=1/256, ttl=254 (no response found!)
   147 291.774939855 192.168.34.7  192.168.56.9  ICMP     164 Echo (ping) request  id=0x0006, seq=2/512, ttl=254 (no response found!)
   148 291.781165007 192.168.34.7  192.168.56.9  ICMP     164 Echo (ping) request  id=0x0006, seq=3/768, ttl=254 (no response found!)
   149 291.789757740 192.168.34.7  192.168.56.9  ICMP     164 Echo (ping) request  id=0x0006, seq=4/1024, ttl=254 (no response found!)
   162 317.209758427 192.168.34.7  192.168.56.9  ICMP     164 Echo (ping) request  id=0x0007, seq=0/0, ttl=254 (no response found!)
   163 317.218511662 192.168.34.7  192.168.56.9  ICMP     164 Echo (ping) request  id=0x0007, seq=1/256, ttl=254 (no response found!)
   164 317.224129458 192.168.34.7  192.168.56.9  ICMP     164 Echo (ping) request  id=0x0007, seq=2/512, ttl=254 (no response found!)
   165 317.229566646 192.168.34.7  192.168.56.9  ICMP     164 Echo (ping) request  id=0x0007, seq=3/768, ttl=254 (no response found!)
   166 317.238179370 192.168.34.7  192.168.56.9  ICMP     164 Echo (ping) request  id=0x0007, seq=4/1024, ttl=254 (no response found!)

▸ Frame 140: 164 bytes on wire (1312 bits), 164 bytes captured (1312 bits) on interface eth0, id 0
▸ Ethernet II, Src: 50:03:00:00:1b:08 (50:03:00:00:1b:08), Dst: 50:01:00:00:1b:08 (50:01:00:00:1b:08)
▸ Internet Protocol Version 4, Src: 10.255.255.3, Dst: 10.255.255.5
▸ User Datagram Protocol, Src Port: 60464, Dst Port: 4789
▾ Virtual eXtensible Local Area Network
  ▸ Flags: 0x0800, VXLAN Network ID (VNI)
    Group Policy ID: 0
    VXLAN Network Identifier (VNI): 1000
    Reserved: 0
▸ Ethernet II, Src: 50:03:00:00:1b:08 (50:03:00:00:1b:08), Dst: 50:05:00:00:1b:08 (50:05:00:00:1b:08)
▸ Internet Protocol Version 4, Src: 192.168.34.7, Dst: 192.168.56.9
▸ Internet Control Message Protocol

0000  50 01 00 00 1b 08 50 03  00 00 1b 08 08 00 45 00   P·····P· ······E·
0010  00 96 00 00 00 00 ff 11  a7 4f 0a ff ff 03 0a ff   ········ ·O······
0020  ff 05 ec 30 12 b5 00 82  00 00 08 00 00 00 00 03   ···0···· ········
0030  e8 00 50 05 00 00 1b 08  50 03 00 00 1b 08 08 00   ··P····· P·······
0040  45 00 00 64 00 04 00 00  fe 01 e1 33 c0 a8 22 07   E··d···· ···3··"·
0050  c0 a8 38 09 00 00 4f 49  00 00 00 04 00 00 00 00   ··8···OI ········
0060  00 96 36 67 ab cd ab cd  ab cd ab cd ab cd ab cd   ··6g···· ········
0070  ab cd ab cd ab cd ab cd  ab cd ab cd ab cd ab cd   ········ ········
0080  ab cd ab cd ab cd ab cd  ab cd ab cd ab cd ab cd   ········ ········
0090  ab cd ab cd ab cd ab cd  ab cd ab cd ab cd ab cd   ········ ········
00a0  ab cd ab cd                                        ····
```

And this time we can see the IP address information in the EVPN Route Type 2:

```
LEAF-3# show bgp l2vpn evpn vni-id 1000  | begin Netw

   Network            Next Hop            Metric     LocPrf     Weight Path

Route Distinguisher: 10.255.255.3:3    (L3VNI 1000)

*>i[2]:[0]:[0]:[48]:[aabb.cc00.0800]:[32]:[192.168.34.8]/272

                      10.255.255.4                      100          0 i

*>i[2]:[0]:[0]:[48]:[aabb.cc00.0900]:[32]:[192.168.56.9]/272

                      10.255.255.5                      100          0 i

*>i[2]:[0]:[0]:[48]:[aabb.cc00.0c00]:[32]:[192.168.34.12]/272

                      10.255.255.4                      100          0 i
```

32 shows the IPv4 length in bits followed by the IP address itself.

If it was IPv6 we should see 128 instead of 32.

In the above capture, we see the response not found! It is because of the load sharing, the reply packet comes from another link (another spine switch for example).

Let's also take a look at the detailed information about this route:

```
LEAF-3# show bgp l2vpn evpn 192.168.56.9
BGP routing table information for VRF default, address family L2VPN EVPN
Route Distinguisher: 10.255.255.5:32772
BGP routing table entry for [2]:[0]:[0]:[48]:[aabb.cc00.0900]:[32]:[192.168.56.9
]/272, version 39
Paths: (2 available, best #2)
Flags: (0x000202) (high32 00000000) on xmit-list, is not in l2rib/evpn, is not i
n HW

  Path type: internal, path is valid, not best reason: Neighbor Address, no labe
led nexthop
  AS-Path: NONE, path sourced internal to AS
    10.255.255.5 (metric 81) from 10.255.255.2 (10.255.255.2)
      Origin IGP, MED not set, localpref 100, weight 0
      Received label 56 1000
      Extcommunity: RT:65000:56 RT:65000:1000 ENCAP:8 Router MAC:5005.0000.1b08
      Originator: 10.255.255.5 Cluster list: 10.255.255.2
```

This time we see two labes: 56 and 1000:

- 56 is the L2VNI and 1000 is L3VNI.

Router MAC: 5005.0000.1b08:

- This is the MAC address of the remote LEAF switch's SVI or VTEP connected to the HOST56-9 which is LEAF-5

```
LEAF-5# show interface vlan 5
Vlan5 is up, line protocol is up, autostate enabled
  Hardware is EtherSVI, address is  5005.0000.1b08
  Internet Address is 192.168.56.254/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
   reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive not supported
  ARP type: ARPA
  Last clearing of "show interface" counters never
!!!!!!!!!!
```

Task 03:

- LEAF-6 should be configured as a Border Leaf node (in order to connect to the L3VPN service that SP is providing)
- PE-CE routing protocol is BGP
- LEAF-5 should connect to the Service Providedr PE devices using an ethernet bundle (Access VLAN 56)

Solution:

We already configured this task in the previous tasks!

```
LEAF-6:

interface Ethernet1/5

  description To-SP-MPLS

  no switchport

  vrf member TEST-1000

  ip address 10.6.18.6/24

  no shutdown

router bgp 65000

  vrf TEST-1000

   timers bgp 7 21

   address-family ipv4 unicast

   neighbor 10.6.18.18

     remote-as 10000

     address-family ipv4 unicast

LEAF-5:

interface port-channel56

  switchport access vlan 5

interface Ethernet1/5

  switchport access vlan 5

  channel-group 56


interface Ethernet1/6

  switchport access vlan 5

  channel-group 56
```

Task 04:

- Configure Service Provider PE devices as per the diagram
- VPNv4 Unicast and L2VPN EVPN for BGP neighbors should only be enabled (Full Mesh)
- Provide EVPN (with multi-homing) service to the customer as per the diagram (Use ingress replication for the BUM traffic forwarding)
- Use MPLS as the encapsulation for L2VPN service
- Provide L3VPN service to the customer as per the diagram (PE-CE routing protocol: BGP)
- All customer devices must have reachability to any host

Solution:

As we mentioned earlier, IOS-XE device in this lab (CSR1000v) has some limitation when it comes to EVPN, let's check them:

```
CSR1K-PE-16#show l2vpn evpn capabilities

EVPN Platform Capabilities

 VLAN-based EVPN Instance: supported

 VLAN-bundle EVPN Instance: supported

 VLAN-aware EVPN Instance: supported

 Ingress replication type: supported

 Point-to-multipoint replication type: not supported

 Multipoint-to-multipoint replication type: not supported

 Static replication type: not supported

 Per-BD MPLS label allocation mode: supported

 Per-CE MPLS label allocation mode: supported

 Per-EVI MPLS label allocation mode: not supported

 Address resolution flooding suppression: not supported

 VLAN configuration mode: not supported

 MPLS encapsulation: supported

 VxLAN encapsulation: not supported

 Multi-homing aliasing: supported

 VPLS stitching: not supported

 VPLS seamless integration: not supported
```

```
Multi-homing all active redundancy mode: supported

Multi-homing single active redundancy mode: not supported

Ethernet Segment old config model: supported

IP local learning: not supported

VPLS stitching single-active dual-homing: not supported
```

We can realize that, it only supports MPLS as the encapsulation, All-Active Multi-Homing is only supperted, IP learning not supported and Multicast Replication not supported.

First, Full mesh BGP is required, we need to only enable VPNv4 Unicast and L2VPN EVPN on all PE devices:

```
CSR1K-PE-16:

router bgp 10000

 bgp router-id 10.255.255.16

 bgp log-neighbor-changes

 no bgp default ipv4-unicast

 neighbor 10.255.255.17 remote-as 10000

 neighbor 10.255.255.17 update-source Loopback0

 neighbor 10.255.255.18 remote-as 10000

 neighbor 10.255.255.18 update-source Loopback0

 !

 address-family ipv4

 exit-address-family

 !

 address-family vpnv4

  neighbor 10.255.255.17 activate

  neighbor 10.255.255.17 send-community extended

  neighbor 10.255.255.18 activate

  neighbor 10.255.255.18 send-community extended

 exit-address-family

 !

 address-family l2vpn evpn

  neighbor 10.255.255.17 activate

  neighbor 10.255.255.17 send-community both

  neighbor 10.255.255.18 activate

  neighbor 10.255.255.18 send-community both
 exit-address-family
 !
 address-family ipv4 vrf Cust-DC-MPLS
  redistribute connected
  neighbor 10.16.19.19 remote-as 65001
  neighbor 10.16.19.19 activate
 exit-address-family
```

CSR1K-PE-17:

```
router bgp 10000
 bgp router-id 10.255.255.17
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.255.255.16 remote-as 10000
 neighbor 10.255.255.16 update-source Loopback0
 neighbor 10.255.255.18 remote-as 10000
 neighbor 10.255.255.18 update-source Loopback0
 !
 address-family ipv4
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.255.255.16 activate
  neighbor 10.255.255.16 send-community extended
  neighbor 10.255.255.18 activate
  neighbor 10.255.255.18 send-community extended
 exit-address-family
 !
 address-family l2vpn evpn
  neighbor 10.255.255.16 activate
  neighbor 10.255.255.16 send-community both
  neighbor 10.255.255.18 activate
  neighbor 10.255.255.18 send-community both
 exit-address-family
```

CSR1K-PE-18:

```
router bgp 10000
 bgp router-id 10.255.255.18
 bgp log-neighbor-changes
 no bgp default ipv4-unicast
 neighbor 10.255.255.16 remote-as 10000
 neighbor 10.255.255.16 update-source Loopback0
 neighbor 10.255.255.17 remote-as 10000
 neighbor 10.255.255.17 update-source Loopback0
 !
 address-family ipv4
 exit-address-family
 !
 address-family vpnv4
  neighbor 10.255.255.16 activate
  neighbor 10.255.255.16 send-community extended
  neighbor 10.255.255.17 activate
  neighbor 10.255.255.17 send-community extended
 exit-address-family
 !
 address-family l2vpn evpn
  neighbor 10.255.255.16 activate
  neighbor 10.255.255.16 send-community both
  neighbor 10.255.255.17 activate
  neighbor 10.255.255.17 send-community both
 exit-address-family
 !
 address-family ipv4 vrf Cust-DC-MPLS
  redistribute connected
  neighbor 10.6.18.6 remote-as 65000
  neighbor 10.6.18.6 timers 7 21
  neighbor 10.6.18.6 activate
  neighbor 10.18.22.22 remote-as 65002
  neighbor 10.18.22.22 activate
 exit-address-family
```

These are just basic configuration to get L2VPN and L3VPN ready.

```
CSR1K-PE-16:
vrf definition Cust-DC-MPLS
 rd 10000:1
 route-target export 10000:1
 route-target import 10000:1
 !
 address-family ipv4
 exit-address-family
!
!
interface GigabitEthernet2
 vrf forwarding Cust-DC-MPLS
 ip address 10.16.19.16 255.255.255.0
 negotiation auto
 no mop enabled
 no mop sysid
!
router bgp 10000
 !
 address-family ipv4 vrf Cust-DC-MPLS
  redistribute connected
  neighbor 10.16.19.19 remote-as 65001
  neighbor 10.16.19.19 activate
 exit-address-family
!
CSR1K-PE-18:
vrf definition Cust-DC-MPLS
 rd 10000:1
 route-target export 10000:1
 route-target import 10000:1
 !
 address-family ipv4
 exit-address-family
!
```

```
!
interface GigabitEthernet3
 vrf forwarding Cust-DC-MPLS
 ip address 10.18.22.18 255.255.255.0
 negotiation auto
 no mop enabled
 no mop sysid
!
interface GigabitEthernet5
 vrf forwarding Cust-DC-MPLS
 ip address 10.6.18.18 255.255.255.0
 negotiation auto
 no mop enabled
 no mop sysid
!
router bgp 10000
 !
 address-family ipv4 vrf Cust-DC-MPLS
  redistribute connected
  neighbor 10.6.18.6 remote-as 65000
  neighbor 10.6.18.6 timers 7 21
  neighbor 10.6.18.6 activate
  neighbor 10.18.22.22 remote-as 65002
  neighbor 10.18.22.22 activate
 exit-address-family
!
```

Everything is ready to begin configuring EVPN service:

We start with the simplest one (CSR1K-PE-17) which is single-homed to the customer devices.

EVPN global configuration is needed to specify the router-id and replication type, after that, we should configure an evpn instance for each of the customers. In this case the evpn instance ID will be 23 and it’s type is vlan-aware, what is vlan-aware, vlan-based… etc? they will be explained soon!

```
CSR1K-PE-17:
l2vpn evpn
 replication-type ingress
 mpls label mode per-ce
 router-id Loopback0
l2vpn evpn instance 23 vlan-aware
 rd 10000:23
```

There are 4 types of EVPN instance in this platform:

```
CSR1K-PE-17(config)#l2vpn evpn instance 1000 ?
  point-to-point  EVPN Virtual Private Wire Service (EVPN VPWS)
  vlan-aware      VLAN-Aware service interface
  vlan-based      VLAN-Based service interface
  vlan-bundle     VLAN Bundle service interface
```

- Point-to-point is for VPWS service
- VLAN-based is an EVPN instance that only supports a single broadcast domain when it comes to EVPN Route Types information
- VLAN-Aware is an EVPN instance that supports multiple broadcast domains when it comes to EVPN Route Types information, it gives us the ability to natively support multiple ethernet broadcast domains which is going to be specified by an ethernet-tag, so the customer can have multiple separate broadcast domains
- VLAN-Boundle is an EVPN instance that can support handle multiple VLANs with a single broadcast domain, which means, the customer can send the service provider device (PE device) multiple VLAN tagged frames and the service provider behaves with them as a single broadcast domain.

In this example there was no need to use VLAN-Aware instance, only for demonstration purpose we used it!

The next step is to configure service instances under the physical interfaces that are connected to the CE devices:

```
CSR1K-PE-17:
interface GigabitEthernet1
 no shutdown
 service instance 23 ethernet
  encapsulation default
 !
!
```

```
interface GigabitEthernet4
 no shutdown
 service instance 23 ethernet
  encapsulation default
 !
!
interface GigabitEthernet5
 no shutdown
 service instance 23 ethernet
  encapsulation default
 !
!
```

And, configure a bridge-domain (think of a bridge domain just like a virtual switch that can have local physical or remote ports (on another PE devices)):

```
CSR1K-PE-17:
bridge-domain 23
 member GigabitEthernet1 service-instance 23
 member GigabitEthernet4 service-instance 23
 member GigabitEthernet5 service-instance 23
 member evpn-instance 23 ethernet-tag 1819
!
```

3 of the members are physical ports, the other one is connected to the EVPN-Instance.

NOTE: The customer can have multiple broadcast domains which are specified by ethernet-tags. In this case we used only a single broadcast domain 1819.

That is all needs to be done on CSR1K-PE-17 router!

Let's configure the other two PE devices. The configuration is almost the same except Multi-Homing configuration.

In this case we need to define an Ethernet Segment.

If you take a closer look at the topology, CSR1K-PE-16 and CSR1K-PE-18 are connected to the LEAF-5.

LEAF-5 forms a Port-Channel with these two PE devices, which means both of those links are part of the same Ethernet Segment.

Thanks to the Route Type 4, EVPN can provide this kind of setup (unlike VPLS which has it's limitations when it comes to the loop prevention etc…).

First step is to setup an MCLAG between two PE devices! How it is possible?! EVPN will handle it don't worry ☺

```
CSR1K-PE16:

interface Port-channel23

 no shutdown

 evpn ethernet-segment 1618

  identifier type 0 AA.BB.CC.DD.EE.FF.16.18.00

  redundancy all-active

 service instance 23 ethernet

  encapsulation default

 !

interface GigabitEthernet5

 no shutdown

 channel-group 23

!


CSR1K-PE18:

interface Port-channel23

 no shutdown

 evpn ethernet-segment 1618

  identifier type 0 AA.BB.CC.DD.EE.FF.16.18.00

  redundancy all-active

 service instance 23 ethernet

  encapsulation default

 !

interface GigabitEthernet6

 no shutdown

 channel-group 23

!
```

That's all! We created a port-channel, we put the same ESI value (Ethernet Segment ID) on both of the router's port-channels, and the physical interfaces are being configured as a part of that port-channel.

The rest of the configuration is the same as CSR1K-PE-17:

```
CSR1K-PE16 and CSR1K-PE18:
l2vpn evpn
 replication-type ingress
 mpls label mode per-ce
 router-id Loopback0
!
l2vpn evpn instance 23 vlan-aware
 rd 10000:23
!
bridge-domain 23
 member Port-channel23 service-instance 23
 member evpn-instance 23 ethernet-tag 1819
!
```

We are done!

Verification:

```
CSR1K-PE-16#show bgp l2vpn evpn summary  | begin Ne
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.255.255.17   4        10000     233     258      112    0    0 03:26:31        4
10.255.255.18   4        10000     279     284      112    0    0 03:28:39        4


IOL-CE-23#ping 192.168.56.255 re 1
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 192.168.56.255, timeout is 2 seconds:


Reply to request 0 from 192.168.56.20, 1 ms
Reply to request 0 from 192.168.56.14, 24 ms
Reply to request 0 from 192.168.56.10, 21 ms
Reply to request 0 from 192.168.56.15, 11 ms
Reply to request 0 from 192.168.56.13, 6 ms
Reply to request 0 from 192.168.56.9, 6 ms
Reply to request 0 from 192.168.56.21, 1 ms
```

BUM Traffic forwarding works fine, Let’s also check Known unicast:

```
IOL-CE-23#ping 192.168.56.14 re 5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.56.14, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 5/7/11 ms

IOL-CE-23#ping 192.168.56.9 re 5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.56.9, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/11 ms
```

This time we don't have VXLAN encapsulation, there is no L2VNI or L3VNI.

Instead we have MPLS as the encapsulation, MPLS must be enabled behind the scenes (don't forget to enable it in the core).

Let's check EVPN route types and discuss about them:

```
CSR1K-PE-17#show bgp l2vpn evpn  | begin Netw

     Network          Next Hop            Metric LocPrf Weight Path

Route Distinguisher: 10000:23

 *>i  [1][10000:23][00AABBCCDDEEFF161800][1819]/23

                      10.255.255.18            0    100      0 ?

 *mi                   10.255.255.16            0    100      0 ?

Route Distinguisher: 10.255.255.16:1

 *>i  [1][10.255.255.16:1][00AABBCCDDEEFF161800][4294967295]/23

                      10.255.255.16            0    100      0 ?

Route Distinguisher: 10.255.255.18:1

 *>i  [1][10.255.255.18:1][00AABBCCDDEEFF161800][4294967295]/23

                      10.255.255.18            0    100      0 ?

Route Distinguisher: 10000:23

 *mi  [2][10000:23][1819][48][AABBCC000900][0][*]/20

                      10.255.255.18            0    100      0 ?

 *>i                   10.255.255.16            0    100      0 ?

 *>i  [2][10000:23][1819][48][AABBCC000A00][0][*]/20

                      10.255.255.18            0    100      0 ?

 *>i  [2][10000:23][1819][48][AABBCC000D00][0][*]/20

                      10.255.255.16            0    100      0 ?
 *>i  [2][10000:23][1819][48][AABBCC000E00][0][*]/20
                      10.255.255.16            0    100      0 ?
 *>i  [2][10000:23][1819][48][AABBCC000F00][0][*]/20
                      10.255.255.18            0    100      0 ?
 *>   [2][10000:23][1819][48][AABBCC001410][0][*]/20
                      ::                                 32768 ?
 *>   [2][10000:23][1819][48][AABBCC001530][0][*]/20
                      ::                                 32768 ?
 *>   [2][10000:23][1819][48][AABBCC001700][0][*]/20
                      ::                                 32768 ?
 *>i  [3][10000:23][1819][32][10.255.255.16]/17
                      10.255.255.16            0    100      0 ?
 *>   [3][10000:23][1819][32][10.255.255.17]/17
                      ::                                 32768 ?
 *>i  [3][10000:23][1819][32][10.255.255.18]/17
                      10.255.255.18            0    100      0 ?
```

On CSR1K-PE-17 we have 3 types of EVPN Routes (Type 1, 2 and 3).

There are two kinds of Route Type 1 in this case:

```
 *>i  [1][10000:23][00AABBCCDDEEFF161800][1819]/23
                      10.255.255.18            0    100      0 ?
  *mi                  10.255.255.16            0    100      0 ?
```

The first one is for Aliasing, Which means this router can do load sharing between two remote PE devices connected to the same Ethernet Segment.

In this route we have:

- 1: Route Type 1
- 10000:23: Route Distinguisher
- 00AABBCCDDEEFF161800: Ethernet Segment ID
- 1819: Ethernet-Tag

Let’s also take a look at the detailed information:

```
CSR1K-PE-17#show bgp l2vpn evpn  route-type 1
BGP routing table entry for [1][10000:23][00AABBCCDDEEFF161800][1819]/23, version 12
Paths: (2 available, best #1, table evi_23)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    10.255.255.18 (metric 2) (via default) from 10.255.255.18 (10.255.255.18)
      Origin incomplete, metric 0, localpref 100, valid, internal, multipath, best
      Rcvd Label: 18000, Local Label: None
      Extended Community: RT:10000:23
      rx pathid: 0, tx pathid: 0x0
      net: 0x7F10CB4F0C98, path: 0x7F1139EDF780, pathext: 0x7F10CB5F5AF8
      flags: net: 0x10, path: 0x4000000000043, pathext: 0x81
      attribute: 0x7F10CB500EE8, ref: 4
      Updated on Apr 10 2021 21:00:14 UTC
  Refresh Epoch 2
  Local
    10.255.255.16 (metric 2) (via default) from 10.255.255.16 (10.255.255.16)
      Origin incomplete, metric 0, localpref 100, valid, internal, multipath(oldest)
      Rcvd Label: 16000, Local Label: None
      Extended Community: RT:10000:23
      rx pathid: 0, tx pathid: 0
      net: 0x7F10CB4F0C98, path: 0x7F1139EDFA38, pathext: 0x0
      flags: net: 0x10, path: 0x4000000000043, pathext: 0x0
      attribute: 0x7F10CB500EE8, ref: 4
      Updated on Apr 10 2021 21:00:10 UTC
```

CSR1K-PE-17 received two labels from two remote PE devices (R16 and R18), when it wants to do aliasing, it will use these labels as the buttom of the stack label (when Route Type 2 information about some specific hos is missing from one of the PEs, if it is not missing so PE will use the Route-Type 2 received label instead).

There is also another EVPN Route Type 1 in this example:

```
*>i  [1][10.255.255.16:1][00AABBCCDDEEFF161800][4294967295]/23
                      10.255.255.16            0    100      0 ?
Route Distinguisher: 10.255.255.18:1
 *>i  [1][10.255.255.18:1][00AABBCCDDEEFF161800][4294967295]/23
```

This route is for split horizon label advertisement for a specific Ethernet Segment.

- 1: EVPN Route Type 1
- 10.255.255.16:1: Route Distinguisher derived automaticaly from EVPN RID
- 00AABBCCDDEEFF161800: Ethernet Segment
- 4294967295: Ethernet Tag set to maximum value which menas this route is for Split Horizon label advertisement purpose

```
BGP routing table entry for [1][10.255.255.16:1][00AABBCCDDEEFF161800][4294967295]/23, version 3
Paths: (1 available, best #1, table EVPN-BGP-Table)
  Not advertised to any peer
  Refresh Epoch 2
  Local
    10.255.255.16 (metric 2) (via default) from 10.255.255.16 (10.255.255.16)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Rcvd Label: 0, Local Label: None
      Extended Community: RT:10000:23 EVPN LABEL:0x0:Label-26128
      rx pathid: 0, tx pathid: 0x0
      net: 0x7F10CB4F0FA8, path: 0x7F1139EDFC08, pathext: 0x7F10CB5F5A98
      flags: net: 0x0, path: 0x3, pathext: 0x81
      attribute: 0x7F10CB501158, ref: 3
      Updated on Apr 10 2021 21:00:10 UTC
CSR1K-PE-16#show l2vpn evpn ethernet-segment detail
EVPN Ethernet Segment ID: 00AA.BBCC.DDEE.FF16.1800
  Interface:              Po23
  Redundancy mode:        all-active
  DF election wait time:  3 seconds
  Split Horizon label:    1633
  State:                  Ready
  Encapsulation:          mpls
  Ordinal:                0
  RD:                     10.255.255.16:1
    Export-RTs:           10000:23
  Forwarder List:         10.255.255.16 10.255.255.18
```

The Split Horizon label 1633 is encoded inside the route as an extended community value:

LABEL:0x0:Label-26128, There is no information in the RFC 7432 about how exactly it’s encoded!

In this example, there is also Route-Type 2 which is for MAC/IP advertisement, this platform does not support IP learning so that part is missing:

```
CSR1K-PE-16#show bgp l2vpn evpn | begin Net
     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 10000:23
 *mi  [1][10000:23][00AABBCCDDEEFF161800][1819]/23
                      10.255.255.18            0    100      0 ?
 *>                    ::                                 32768 ?
Route Distinguisher: 10.255.255.16:1
 *>   [1][10.255.255.16:1][00AABBCCDDEEFF161800][4294967295]/23
                      ::                                 32768 ?
Route Distinguisher: 10.255.255.18:1
 *>i  [1][10.255.255.18:1][00AABBCCDDEEFF161800][4294967295]/23
                      10.255.255.18            0    100      0 ?
Route Distinguisher: 10000:23
 *>i  [2][10000:23][1819][48][AABBCC000D00][0][*]/20
                      10.255.255.18            0    100      0 ?
 *>i  [2][10000:23][1819][48][AABBCC001410][0][*]/20
                      10.255.255.17            0    100      0 ?
 *>i  [2][10000:23][1819][48][AABBCC001530][0][*]/20
                      10.255.255.17            0    100      0 ?
 *>i  [2][10000:23][1819][48][AABBCC001700][0][*]/20
                      10.255.255.17            0    100      0 ?
 *>   [3][10000:23][1819][32][10.255.255.16]/17
                      ::                                 32768 ?
 *>i  [3][10000:23][1819][32][10.255.255.17]/17
                      10.255.255.17            0    100      0 ?
 *>i  [3][10000:23][1819][32][10.255.255.18]/17
     Network          Next Hop            Metric LocPrf Weight Path
                      10.255.255.18            0    100      0 ?
```

For each of the hosts there is an entry, let’s interpret the first one:

- 2: EVPN Route-Type 2
- 10000:23: Route Distinguisher value
- 1819: Ethernet-Tag value
- 48: MAC address length in bits

- AABBCC000D00: MAC address of the host
- 0: Length of IP address is 0 because it is missing
- *: IP address information is missing (CSR1000v does not support IP address Learning)

Let’s talk about EVPN Route Type 3:

```
CSR1K-PE-16#show bgp l2vpn evpn | begin Net
     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 10000:23
 *mi  [1][10000:23][00AABBCCDDEEFF161800][1819]/23
                      10.255.255.18            0    100      0 ?
 *>                    ::                                 32768 ?
Route Distinguisher: 10.255.255.16:1
 *>   [1][10.255.255.16:1][00AABBCCDDEEFF161800][4294967295]/23
                      ::                                 32768 ?
Route Distinguisher: 10.255.255.18:1
 *>i  [1][10.255.255.18:1][00AABBCCDDEEFF161800][4294967295]/23
                      10.255.255.18            0    100      0 ?
Route Distinguisher: 10000:23
 *>   [2][10000:23][1819][48][AABBCC000900][0][*]/20
                      ::                                 32768 ?
 *>i  [2][10000:23][1819][48][AABBCC000D00][0][*]/20
                      10.255.255.18            0    100      0 ?
 *>i  [2][10000:23][1819][48][AABBCC001410][0][*]/20
                      10.255.255.17            0    100      0 ?
 *>i  [2][10000:23][1819][48][AABBCC001530][0][*]/20
                      10.255.255.17            0    100      0 ?
 *>i  [2][10000:23][1819][48][AABBCC001700][0][*]/20
                      10.255.255.17            0    100      0 ?
 *>   [3][10000:23][1819][32][10.255.255.16]/17
                      ::                                 32768 ?
 *>i  [3][10000:23][1819][32][10.255.255.17]/17
                      10.255.255.17            0    100      0 ?
 *>i  [3][10000:23][1819][32][10.255.255.18]/17
                      10.255.255.18            0    100      0 ?
```

This route is for Ingress Replication Purpose, the BUM traffic is going to be handled by the label received from this route. Each PE device advertises it’s own label for BUM traffic encapsulation purpose, as an example, when IOL-CE-23 sends some Multicast traffic destined to HOST56-14, The CSR1K-PE-17 encapsulates this multicast traffic and put's the BUM label received from PE-16 or PE-18 at the buttom of the stack and sends it towards the remote PE.

```
CSR1K-PE-16#show bgp l2vpn evpn route-type 3 | begin 10.255.255.17
BGP routing table entry for [3][10000:23][1819][32][10.255.255.17]/17, version 27
Paths: (1 available, best #1, table evi_23)
  Flag: 0x100
  Not advertised to any peer
  Refresh Epoch 1
  Local
    10.255.255.17 (metric 2) (via default) from 10.255.255.17 (10.255.255.17)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:10000:23
      PMSI Attribute: Flags:0x0, Tunnel type:IR, length 4, label:17003 tunnel identifier: < Tunnel Endpoint:
10.255.255.17 >
      rx pathid: 0, tx pathid: 0x0
      net: 0x7F56B17FD828, path: 0x7F56B181DF30, pathext: 0x7F56B18C86F8
      flags: net: 0x100, path: 0x4000000000003, pathext: 0xA1
      attribute: 0x7F56B180DEC8, ref: 2
      Updated on Apr 10 2021 21:01:09 UTC
```

The Tunnel Type is IR (Ingress Replication), the label received from 10.255.255.17 is 17003.

Let's also interpret the route details:

- 3: EVPN Route-Type 3
- 10000:23: Route Distinguisher
- 1819: Ethernet-Tag
- 32: IP address Length in bits
- 10.255.255.17: The tunnel endpoint EVPN Router ID

There is another route type on CSR1K-PE-16 and CSR1K-PE-18 which is Route-Type 4.

These two PE devices are connected to the same Ethernet Segment, they need to find out about each other and also there should be a Designated Forwarder (DF) election between them.

Only one of those PE devices that are connected to the same Ethernet Segment should be in the forwarding state of the BUM traffic towards the CE device (LEAF-5) otherwise loop can happen.

Let's check the details of route type 4:

```
CSR1K-PE-16#show bgp l2vpn evpn | begin Netw
     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 10000:23
 *mi  [1][10000:23][00AABBCCDDEEFF161800][1819]/23
                      10.255.255.18            0    100      0 ?
 *>                    ::                                 32768 ?
Route Distinguisher: 10.255.255.16:1
 *>   [1][10.255.255.16:1][00AABBCCDDEEFF161800][4294967295]/23
                      ::                                 32768 ?
Route Distinguisher: 10.255.255.18:1
 *>i  [1][10.255.255.18:1][00AABBCCDDEEFF161800][4294967295]/23
                      10.255.255.18            0    100      0 ?
Route Distinguisher: 10000:23
 *>   [2][10000:23][1819][48][AABBCC000900][0][*]/20
                      ::                                 32768 ?
 *>i  [2][10000:23][1819][48][AABBCC000D00][0][*]/20
                      10.255.255.18            0    100      0 ?
 *>i  [2][10000:23][1819][48][AABBCC001410][0][*]/20
                      10.255.255.17            0    100      0 ?
 *>i  [2][10000:23][1819][48][AABBCC001530][0][*]/20
                      10.255.255.17            0    100      0 ?
 *>i  [2][10000:23][1819][48][AABBCC001700][0][*]/20
                      10.255.255.17            0    100      0 ?
 *>   [3][10000:23][1819][32][10.255.255.16]/17
                      ::                                 32768 ?
 *>i  [3][10000:23][1819][32][10.255.255.17]/17
                      10.255.255.17            0    100      0 ?
 *>i  [3][10000:23][1819][32][10.255.255.18]/17
                      10.255.255.18            0    100      0 ?
Route Distinguisher: 10.255.255.16:6144
 *>   [4][10.255.255.16:6144][00AABBCCDDEEFF161800][32][10.255.255.16]/23
                      ::                                 32768 ?
Route Distinguisher: 10.255.255.18:6144
 *>i  [4][10.255.255.18:6144][00AABBCCDDEEFF161800][32][10.255.255.18]/23
                      10.255.255.18            0    100      0 ?
```

- 4: EVPN Route-Type 4
- 10.255.255.16:6144: Route-Distinguisher automaticaly derived from EVPN RID
- 00AABBCCDDEEFF161800: Ethernet Segment ID
- 32: IP address length in bits
- 10.255.255.16: Router that is advertising this route

```
CSR1K-PE-16#show bgp l2vpn evpn route-type 4

BGP routing table entry for [4][10.255.255.16:6144][00AABBCCDDEEFF161800][32][10.255.255.16]/23, version 16

Paths: (1 available, best #1, table EVPN-BGP-Table)

  Advertised to update-groups:

     1

  Refresh Epoch 1

  Local

    :: (via default) from 0.0.0.0 (10.255.255.16)

      Origin incomplete, localpref 100, weight 32768, valid, sourced, local, best

      Extended Community: EVPN ES-IMPORT:0xAABB:0xCCDD:0xEEFF

      rx pathid: 0, tx pathid: 0x0

      net: 0x7F56B17FE158, path: 0x7F56B181E670, pathext: 0x7F56B18C8998

      flags: net: 0x0, path: 0x28000003, pathext: 0x81

      attribute: 0x7F56B180E270, ref: 2

      Updated on Apr 10 2021 20:59:54 UTC


CSR1K-PE-16#show l2vpn evpn evi detail

EVPN instance:       23 (VLAN Aware)

  RD:                10000:23 (cfg)

  Import-RTs:        10000:23

  Export-RTs:        10000:23

  Per-EVI Label:     none

  State:             Established

  Replication Type:  Ingress

  Encapsulation:     mpls

  Bridge Domain:     23

    Ethernet-Tag:    1819

    BUM Label:       16001

    Per-BD Label:    none

    State:           Established
    Access If:
    Pseudoports (Labels):
      Port-channel23 service instance 23 (16000) (DF state: forwarding)
        Routes: 0 MAC, 0 MAC/IP
    Peers:
      10.255.255.17
        Routes: 3 MAC, 0 MAC/IP, 1 IMET, 0 EAD
      10.255.255.18
        Routes: 0 MAC, 0 MAC/IP, 1 IMET, 1 EAD
```

CSR1K-PE-16 is in the forwarding state for BUM traffic. The other device should be in the blocked state:

```
CSR1K-PE-18#show l2vpn evpn evi de
EVPN instance:       23 (VLAN Aware)
  RD:                10000:23 (cfg)
  Import-RTs:        10000:23
  Export-RTs:        10000:23
  Per-EVI Label:     none
  State:             Established
  Replication Type:  Ingress
  Encapsulation:     mpls
  Bridge Domain:     23
    Ethernet-Tag:    1819
    BUM Label:       18001
    Per-BD Label:    none
    State:           Established
    Access If:
    Pseudoports (Labels):
      Port-channel23 service instance 23 (18000) (DF state: PE-to-CE BUM blocked)
        Routes: 1 MAC, 0 MAC/IP
    Peers:
      10.255.255.16
        Routes: 0 MAC, 0 MAC/IP, 1 IMET, 1 EAD
      10.255.255.17
        Routes: 3 MAC, 0 MAC/IP, 1 IMET, 0 EAD
```

The last Route Type that we want to discuss about is EVPN Route Type 5.

Take a look at the left side of the topology, in the DC site, we have LEAF-6 which is connected to the Service Provider PE device (CSR1K-PE-18) which is providing MPLS L3VPN service.

How LEAF-6 deals with the routes received from the service provider.

EVPN Route Type 5 is the key!

```
LEAF-6# show bgp l2vpn evpn | begin Netw

#####!!!!!

*>l[5]:[0]:[0]:[24]:[10.6.18.0]/224

                      10.255.255.6             0                     0 10000 ?

*>l[5]:[0]:[0]:[24]:[10.16.19.0]/224

                      10.255.255.6                                   0 10000 ?

*>l[5]:[0]:[0]:[24]:[10.18.22.0]/224

                      10.255.255.6             0                     0 10000 ?

*>l[5]:[0]:[0]:[24]:[172.19.19.0]/224

                      10.255.255.6                                   0 10000 650

01 i

*>l[5]:[0]:[0]:[24]:[172.22.22.0]/224

                      10.255.255.6                                   0 10000 650

02 i
```

Fore each of the individual routes received from Service Provider through BGP neighborship we have an entry, the routing between the VXLAN Segments is going to be handled by the L3VNI which is 1000:

```
LEAF-6# show bgp l2vpn evpn route-type 5

BGP routing table information for VRF default, address family L2VPN EVPN

Route Distinguisher: 10.255.255.6:3    (L3VNI 1000)

BGP routing table entry for [5]:[0]:[0]:[24]:[10.6.18.0]/224, version 7

Paths: (1 available, best #1)

Flags: (0x000002) (high32 00000000) on xmit-list, is not in l2rib/evpn


  Advertised path-id 1

  Path type: local, path is valid, is best path, no labeled nexthop

  Gateway IP: 0.0.0.0

  AS-Path: 10000 , path sourced external to AS
  10.255.255.6 (metric 0) from 0.0.0.0 (10.255.255.6)
    Origin incomplete, MED 0, localpref 100, weight 0
    Received label 1000
    Extcommunity: RT:65000:1000 ENCAP:8 Router MAC:5006.0000.1b08
```

This time we don't have any ESI or Ethernet-tag, the values are 0.

- 5: EVPN Route Type 5
- 24: Prefix Length in bits
- 10.6.18.0: Network ID
- Received Label 1000: L3VNI which is handling the routing