EVPN-VXLAN DC IP FABRIC MAC-VRF L2 services

Engineering
Simplicity

Table of Contents

Introduction

Leveraging virtualization and multitenancy in a data center network has historically been complex to implement with multiple touch points required before fully connecting the new workloads. With EVPN-VXLAN and MAC-VRFs, the new server VLAN/port connect provisioning tasks are greatly simplified as they are enabled at the point of server attachment only – at the leaf/ToR level. Additionally, the overlay segmentation of services at the L2 level using MAC-VRF is improved for individual VLAN(VNI) or a set of VLAN(VNI) as they can be allocated to a separate EVPN instance (EVI) for additional isolation or grouped under a common instance.

Customer use case

With the new MAC-VRF (multiple EVPN instances – EVIs) approach provisioning the L2 Ethernet in the data center use-case is expanded to support additional service-type and VLAN to VNI mapping options. These features support several useful combinations of L2 Ethernet to L3 instances mappings that help facilitate the end-to-end segmentation and security strategy in the data center network. This is illustrated in Figure 1 where tenant44 (mac-vrf44), tenant55 (mac-vrf55) and tenant66 (mac-vrf66) are fully isolated with dedicated L2 routing instance at the top of the rack (TOR) switch level.

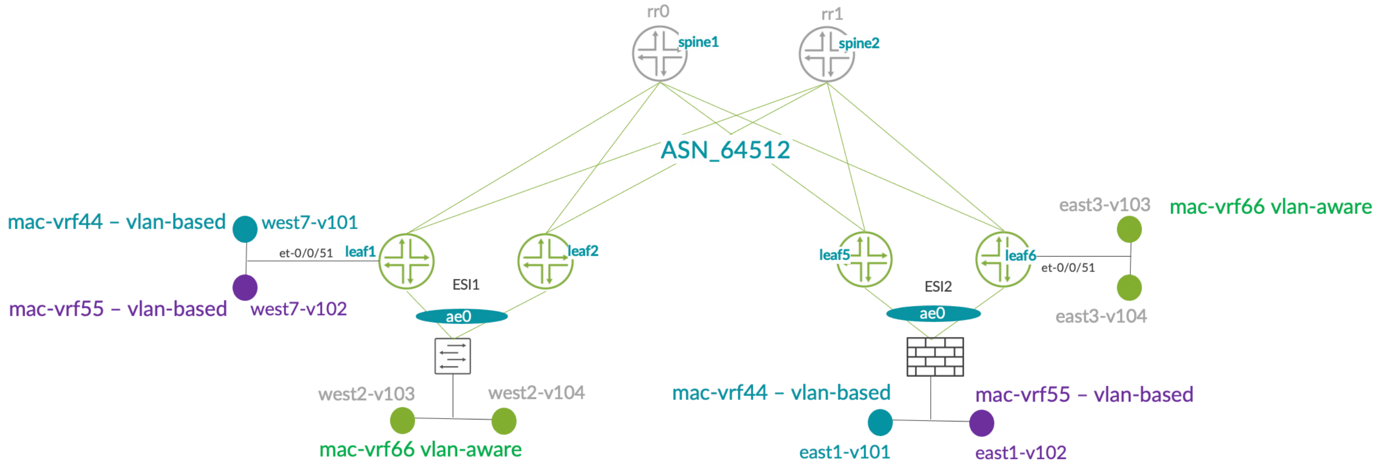


Figure 1 EVPN-VXLAN IP Fabric with Multiple MAC-VRF (Multi EVI) and Multiple L2 Service Types

In the customer use-case shown above servers connected to west2 and east3 ports, were deployed in the DC with the similar security characteristics, including same zone and same policing at the FW level and are just enabled with different IP subnets so they were grouped under the same mac-vrf66 using the vlan-aware service-type and corresponding type-5 VRF. In contrast, other servers like west7 connected) have a firewall with different security zones and specific security policies. As a result, these servers are not able to communicate to west2/east3 services. At the TOR switch level, they are provisioned with dedicated EVPN L2 MAC-VRFs (mac-vrf44 and mac-vrf55), and their first hop IP gateway is directly enabled at the firewall (east1 connected). This arrangement provides consistent services separation and control. These basic requirements are used to show the way MAC-VRF can be enabled in an EVPN-VXLAN DC fabric. The L2 port attachment shown in this paper is implemented at the TOR level, which is closer to the server port and does not require any changes at the spine nodes. In this example the spines are “lean”, which is to say they only provide transit IP forwarding in the underlay and iBGP route reflection in the overlay. Because the spines don’t provide VXLAN tunnel termination any changes for servers connect VLAN provisioning is only required at the leaf QFX5120 or QFX5110 devices. In case the preferred way of BGP overlay peering was eBGP, this is also possible and in this case each leaf would be using a unique overlay ASN (same as underlay for example) while both spines in the topology shown above would be in a common overlay ASN. In case of the eBGP overlay the *multihop no-nexthop*-change knob is required at the spines at the overlay peering level.

MAC-VRF and EVPN-VXLAN in DC

EVPN-VXLAN reference architectures for MAC-VRF

The EVPN-VXLAN MAC-VRF capabilities enabled at the Juniper QFX switches functioning as part of an EVPN-VXLAN IP Clos architecture extend the number of L2 virtualization options available to fabric administrators for connecting new server workloads; for example, being able to fully isolate the servers or group them under the same L2 VRF.
These new virtualization options are part of the Network Virtualization Overlay – NVO architecture, which means it’s built on solid foundations of an IP routed underlay with a BGP based EVPN overlay for MAC advertisement and VXLAN tunnel signaling. The use of EVPN-VXLAN offers significant scaling and performance improvements over the traditional flooding-based MAC address learning used in other architectures.

Before we focus on a specific configuration example, it’s important to highlight that the MAC-VRF capabilities introduced in Junos 20.4 for QFX5110/5120/QFX10K switches can be enabled in one of the two main Juniper reference DC architectures - Edge-routed bridging (ERB) or Bridged Overlay (BO). These reference architectures are shown in Figure 2.

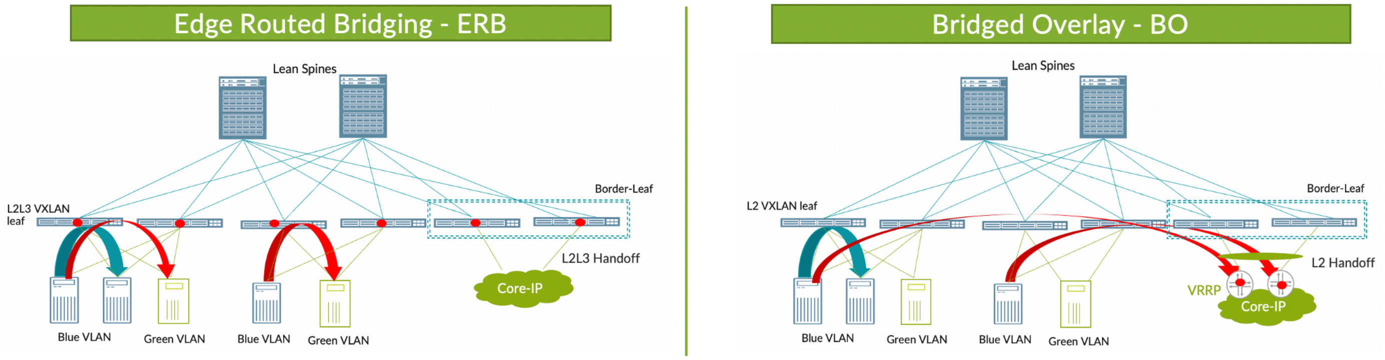


Figure 2 EVPN-VXLAN main reference DC architectures

MAC-VRF L2 services are supported in all reference architectures regardless of the location of the first hop IP gateway. That is, whether it’s located at the leaf devices (ERB) or outside of the fabric (BO). The main difference between the two IP Clos architectures is related to the placement of the first hop gateway – distributed IP anycast gateways within the fabric in case of ERB (Edge Routed Bridging) or centralized outside of the fabric, in case of BO (Bridged Overlay). Some additional design differentiators are highlighted below:

Why ERB design?	Why BO design?
• Reduced blast radius and high tenant segmentation • Distributed tenant IP management	• No IP gateway migration/ IP gateways are managed outside of the fabric • vlan-id overlapping/vlan-bundling services

Both reference designs can be extended to a multipod 5-stage IP Fabric design with the addition of super-spines.

MAC-VRF L2 services options

One of the reasons for considering multiple MAC-VRFs within an EVPN-VXLAN fabric is the added flexibility of enabling different service-types at the same leaf node depending on the requirements of the customer. The data center operator now has the option of provisioning within the EVPN MAC-VRF instances various service-types, to either group or fully isolate the tenant's workloads, at the fabric leaf switch.

QFX5K's have supported vlan-aware service-types in a default-switch EVI but starting Junos 20.4 all QFX5110/5120/QFX10K switches offer support for all the following three EVPN service type options, as defined in rfc8365 and rfc7432. These options relate to how VLANs and EVPN-VXLAN Virtual Network Identifiers (VNIs) are handled:

- vlan-based with each VLAN/VNI gets a dedicated mac-vrf
- vlan-aware with multiple VLANs/VNIs under the same mac-vrf
- vlan-bundle with multiple VLANs under the same mac-vrf using the same VNI

Figure 3 and 4 shows the main differences between the vlan-based and vlan-aware service types that can be enabled within different MAC-VRF instances in a TOR switch:

vlan-based

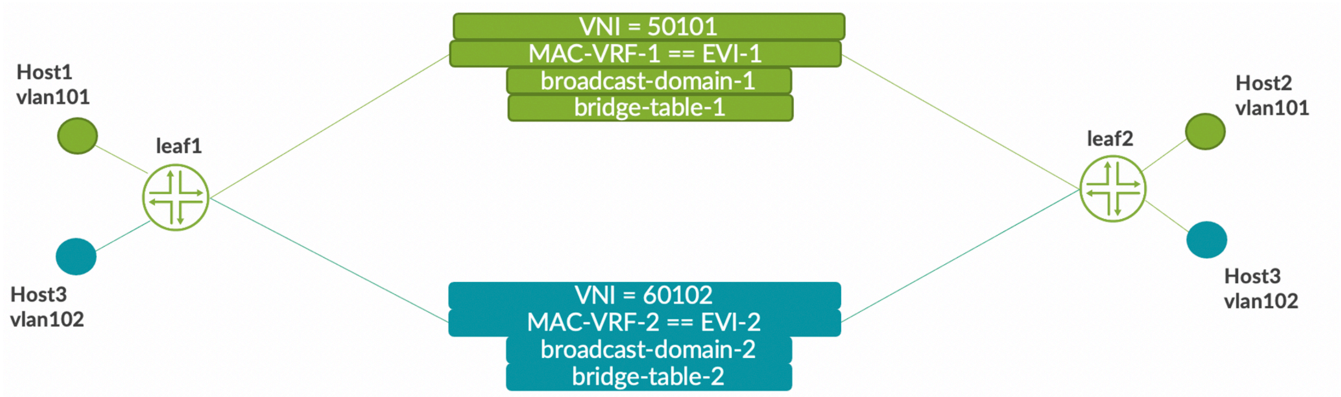


Figure 3 VLAN-based EVPN-VXLAN

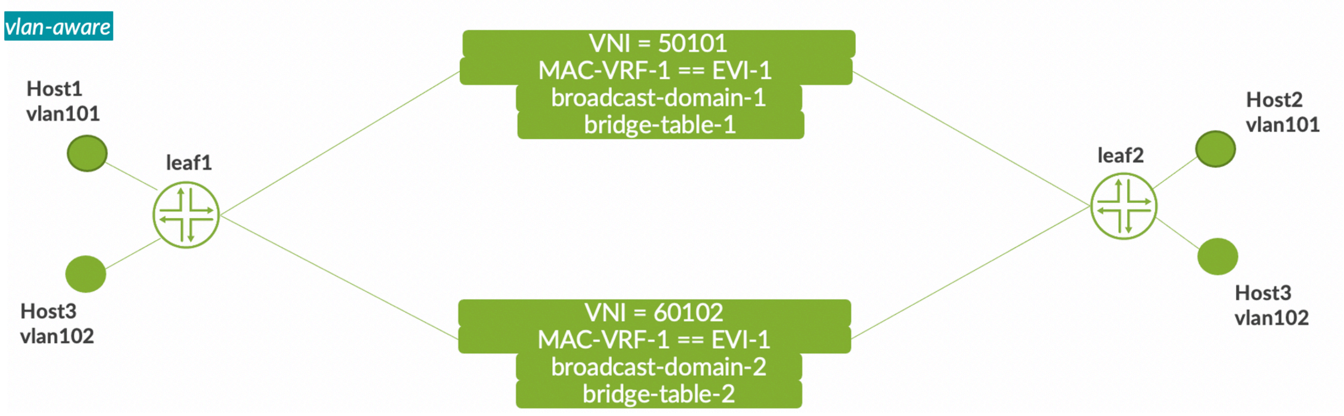


Figure 4 VLAN-aware EVPN-VXLAN

In a vlan-based service each vlan-vni is associated with a unique MAC-VRF EVI so when only a subset of vlans is enabled on different leaf nodes, then the vlan-based service type can reduce the number of dynamically established VXLAN tunnels because each MAC-VRF will originate a unique auto-discovery AD route-target (RT). However, when the fabric has a consistent allocation of the VLAN/VNIs then the vlan-aware service type, with all VLAN/VNIs enabled under the same MAC-VRF, can simplify the way server connect operations are delivered. When provisioning a new VLAN in in a vlan-based service the operator must provision a new mac-vrf instance with a unique route-target/route-distinguisher. In contrast, with a vlan-aware service only the new VLAN/VNI needs to be added to the existing MAC-VRF instance.

The vlan-bundle service-type does not support VLAN normalization at the egress leaf. Normalization allows multiple customer VLAN IDs are mapped to a common provider VNI value which reduces the number of bridge tables the operator must deal with at the switch level.

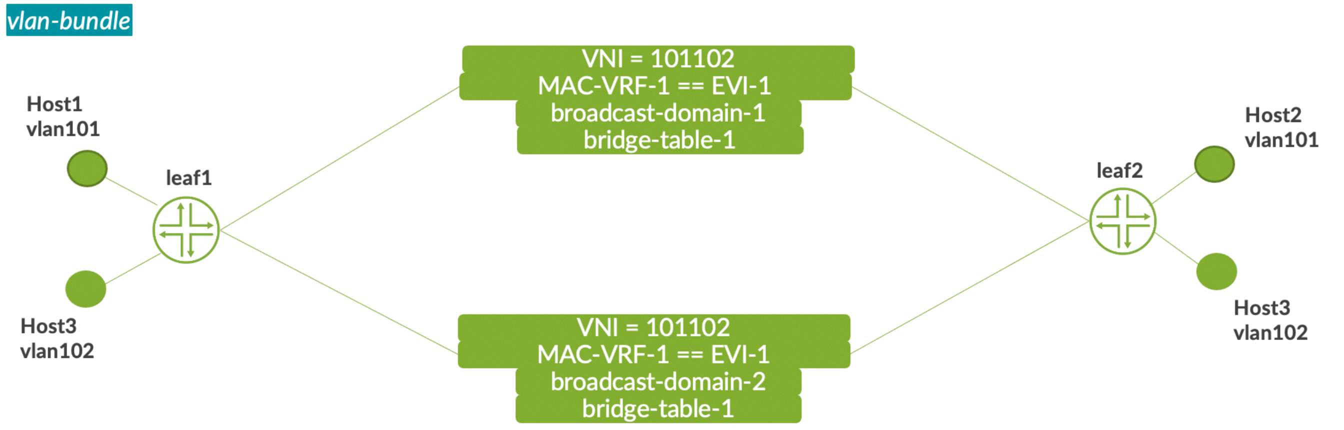


Figure 5 The VLAN Bundle Service Type

The vlan-bundle service type is supported on bridged-overlay (BO) architecture and in this case an external first hop IP gateway is to be considered.

The following table additionally highlights the main differences between the three service types supported as part of the MAC-VRF implementation at the QFX switches, dedicated to EVPN-VXLAN data center fabrics.

vlan-based	vlan-aware	vlan-bundle
▪ 1:1 VLAN ID to EVI ▪ RT per VLAN-VNI ▪ Supports VLAN normalization (optional) ▪ Efficient flooding (broadcast domain per EVI) ▪ VLAN-id overlapping within the node using SP-Style interfaces ▪ Each VLAN-VNI gets a dedicated bridge-table ▪ Type-5 VRF virtualization per edge leaf as separate complementary design option (ERB and BO design)	▪ N:1 (N-VLAN-ID to EVI) ▪ RT per VRF ▪ Supports VLAN normalization (optional) ▪ Efficient flooding ▪ Lower number of EVPN AD routes ▪ VLAN-id overlapping within the node using SP-Style interfaces ▪ Easier to add new VLANs ▪ Each VLAN-VNI gets a dedicated bridge-table ▪ Type-5 VRF virtualization per edge leaf as separate complementary design option (ERB and BO design)	▪ N:1 (N-VLAN-ID to EVI) ▪ RT per VRF ▪ Lower number of EVPN AD routes ▪ Multiple VLAN-id overlapping within the node using SP-Style interfaces ▪ No vlan normalization ▪ All VLANs from given EVI on same bridge-table ▪ Bridged-overlay architecture

MAC-VRF data center fabric provisioning

In this section we walk you through the main steps required to deploy an EVPN-VXLAN MAC-VRF fabric. The focus here is to highlight the main building blocks involved in the implementation of the new L2 virtualization capabilities and to show how proper operation of the same is verified on the QFX5120 leaf devices.

MAC-VRF example topology

Before a MAC-VRF can be provisioned the fabric underlay EBGP and overlay IBGP (for EVPN signaling) should be configured and operational between the leaf and spines. This is illustrated in Figure 6, which provides the example topology. Note that the leaf devices (QFX5120/QFX5110) are attached to both spines (QFX10K). This example is based on an ERB architecture so the (lean) spines only provide forwarding in the underlay and IBGP route reflection in the overlay. As a result, only the leaf nodes provide the VTEP function to originate and terminate VXLAN tunnels based on EVPN signaling.

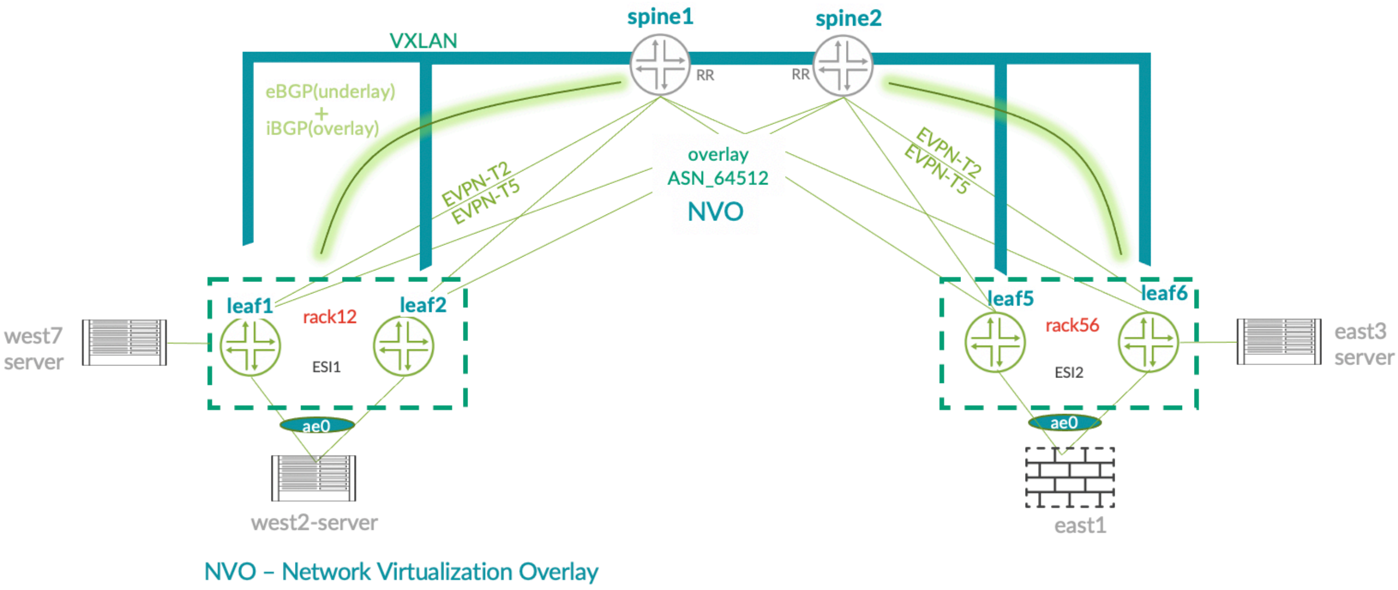


Figure 6 The Fabric Underlay and Overlay (NVO)

The goal of the example design is to fully isolate specific services at the L2 level using mac-vrf44/mac-vrf55 and to use the firewall as their first hop gateway while other services are grouped under the same L2 instance (MAC-VRF) and stay within the fabric as they are less secure (mac-vrf66). Based on the service requirements of this example the mac-vrf66 enabled workloads are not able to communicate with mac-vrf44/mac-vrf55 connected workloads and they stay fully isolated at the top of rack switch level.

The detailed topology depicted in Figure 7 shows the west7/east1 connections enabled with two vlan-based mac-vrfs and the west2/east3 grouped under the same mac-vrf using a vlan-aware service-type.

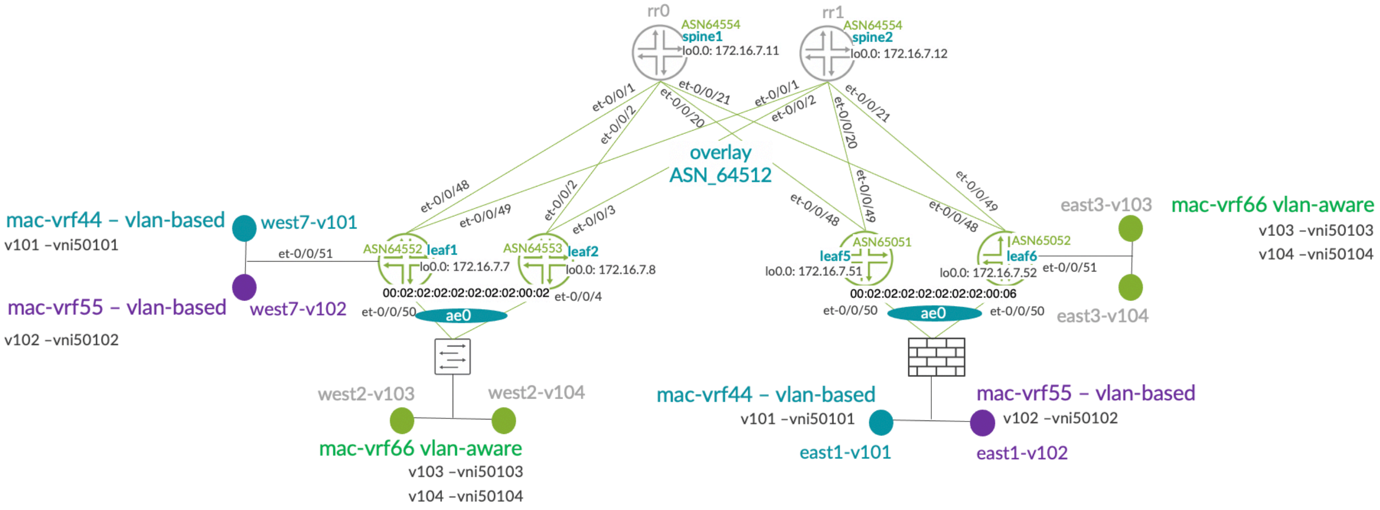


Figure 7 EVPN-VXLAN MAC-VRF Example Topology

In the next section we detail the main MAC-VRF, server interfaces, and BGP building blocks needed to deploy a MAC-VRF based EVPN-VXLAN data center fabric.

MAC-VRF instances and L2 services provisioning

The block diagram shown below summarizes the steps needed to provision a new MAC-VRF based server connect:

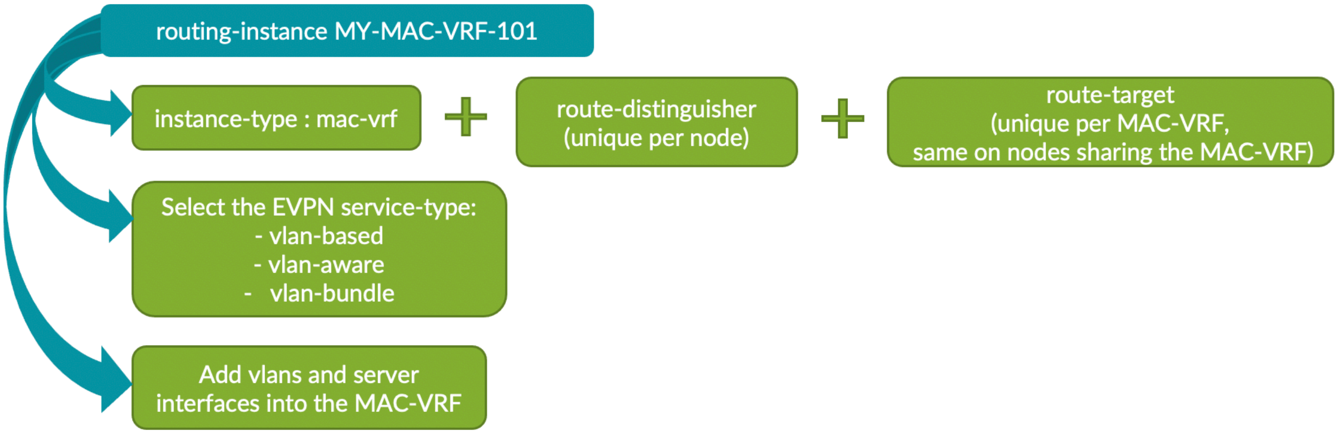

routing-instance MY-MAC-VRF-101
instance-type : mac-vrf
+
route-distinguisher (unique per node)
+
route-target (unique per MAC-VRF, same on nodes sharing the MAC-VRF)
Select the EVPN service-type: - vlan-based - vlan-aware - vlan-bundle
Add vlans and server interfaces into the MAC-VRF

Figure 8 The Primary MAC-VRF Provisioning Blocks

The detailed configuration needed to provision the MAC-VRFs at QFX5120 leaf devices from the diagram shown in figure 7 are shared below:
Leaf1's MAC-VRF configuration:

```
## the macvrf44 at Leaf1 is for the west7 v101 L2 connectivity to be extended to east1 firewall, one vlan per MAC-VRF vlan-based
set routing-instances macvrf44-vlan-based protocols evpn extended-vni-list 50101
set routing-instances macvrf44-vlan-based protocols evpn encapsulation vxlan
set routing-instances macvrf44-vlan-based protocols evpn default-gateway no-gateway-community
set routing-instances macvrf44-vlan-based vtep-source-interface lo0.0
set routing-instances macvrf44-vlan-based instance-type mac-vrf
set routing-instances macvrf44-vlan-based service-type vlan-based
set routing-instances macvrf44-vlan-based route-distinguisher 172.16.7.7:2 ← unique RD per node
set routing-instances macvrf44-vlan-based vrf-target target:1:7777 ← shared on leaf nodes enabled with the given MAC-VRF
set routing-instances macvrf44-vlan-based vlans v101 vlan-id 101
set routing-instances macvrf44-vlan-based vlans v101 interface et-0/0/51.101 ← enabled when interface is using SP-style
set routing-instances macvrf44-vlan-based vlans v101 vxlan vni 50101

## the macvrf55 at Leaf1 is for the west7 v102 L2 connectivity to be extended to east1 firewall
set routing-instances macvrf55-vlan-based protocols evpn extended-vni-list 50102
set routing-instances macvrf55-vlan-based protocols evpn encapsulation vxlan
set routing-instances macvrf55-vlan-based protocols evpn default-gateway no-gateway-community
set routing-instances macvrf55-vlan-based vtep-source-interface lo0.0
set routing-instances macvrf55-vlan-based instance-type mac-vrf
set routing-instances macvrf55-vlan-based service-type vlan-based
set routing-instances macvrf55-vlan-based route-distinguisher 172.16.7.7:3
set routing-instances macvrf55-vlan-based vrf-target target:1:7778
set routing-instances macvrf55-vlan-based vlans v102 vlan-id 102
set routing-instances macvrf55-vlan-based vlans v102 interface et-0/0/51.102
set routing-instances macvrf55-vlan-based vlans v102 vxlan vni 50102
```

```
## the macvrf66 is for the west2 server connectivity L2 extended to east3 and includes two VLAN-VNIs under the same instance
set routing-instances macvrf66-vlan-aware protocols evpn extended-vni-list 50103
set routing-instances macvrf66-vlan-aware protocols evpn extended-vni-list 50104
set routing-instances macvrf66-vlan-aware protocols evpn encapsulation vxlan
set routing-instances macvrf66-vlan-aware protocols evpn default-gateway no-gateway-community
set routing-instances macvrf66-vlan-aware vtep-source-interface lo0.0
set routing-instances macvrf66-vlan-aware instance-type mac-vrf
set routing-instances macvrf66-vlan-aware service-type vlan-aware
set routing-instances macvrf66-vlan-aware interface ae0.0 ← west2 server interface enabled with in enterprise-style
set routing-instances macvrf66-vlan-aware route-distinguisher 172.16.7.7:4 ← unique RD
set routing-instances macvrf66-vlan-aware vrf-target target:1:7779 ← RT common value on nodes enabled with that MAC-VRF
set routing-instances macvrf66-vlan-aware vlans v103 vlan-id 103
set routing-instances macvrf66-vlan-aware vlans v103 l3-interface irb.103
set routing-instances macvrf66-vlan-aware vlans v103 vxlan vni 50103
set routing-instances macvrf66-vlan-aware vlans v104 vlan-id 104
set routing-instances macvrf66-vlan-aware vlans v104 l3-interface irb.104
set routing-instances macvrf66-vlan-aware vlans v104 vxlan vni 50104
```

Worth noting is that when the interface is enabled in an enterprise-style configuration (as used for west2) the interface is declared directly at the MAC-VRF level. In a service provider (SP) style configuration (as used for west7) the interface is specified in the instance under the vlan name hierarchy.

Leaf5's MAC-VRF configuration:

```
### macvrf44 v101 used to connect east1 firewall with west7 using L2 EVPN-VXLAN
set routing-instances macvrf44-vlan-based protocols evpn extended-vni-list 50101
set routing-instances macvrf44-vlan-based protocols evpn encapsulation vxlan
set routing-instances macvrf44-vlan-based protocols evpn default-gateway no-gateway-community
set routing-instances macvrf44-vlan-based vtep-source-interface lo0.0
set routing-instances macvrf44-vlan-based instance-type mac-vrf
set routing-instances macvrf44-vlan-based service-type vlan-based
set routing-instances macvrf44-vlan-based route-distinguisher 172.16.7.51:2 ← unique RD
set routing-instances macvrf44-vlan-based vrf-target target:1:7777
set routing-instances macvrf44-vlan-based vlans v101 vlan-id 101
set routing-instances macvrf44-vlan-based vlans v101 interface ae0.101 ← aggregated interface multihomed with leaf6
set routing-instances macvrf44-vlan-based vlans v101 vxlan vni 50101

### macvrf55 v102 used to connect east1 firewall with west7 using L2 EVPN-VXLAN
set routing-instances macvrf55-vlan-based protocols evpn extended-vni-list 50102
set routing-instances macvrf55-vlan-based protocols evpn encapsulation vxlan
set routing-instances macvrf55-vlan-based protocols evpn default-gateway no-gateway-community
set routing-instances macvrf55-vlan-based vtep-source-interface lo0.0
set routing-instances macvrf55-vlan-based instance-type mac-vrf
set routing-instances macvrf55-vlan-based service-type vlan-based
set routing-instances macvrf55-vlan-based route-distinguisher 172.16.7.51:3
set routing-instances macvrf55-vlan-based vrf-target target:1:7778
set routing-instances macvrf55-vlan-based vlans v102 vlan-id 102
set routing-instances macvrf55-vlan-based vlans v102 interface ae0.102 ← aggregated interface multihomed with leaf6
set routing-instances macvrf55-vlan-based vlans v102 vxlan vni 50102
```

Leaf6 MAC-VRF configuration:

```
### macvrf44 v101 used to connect east1 firewall with west7 using L2 EVPN-VXLAN
set routing-instances macvrf44-vlan-based protocols evpn extended-vni-list 50101
set routing-instances macvrf44-vlan-based protocols evpn encapsulation vxlan
set routing-instances macvrf44-vlan-based protocols evpn default-gateway no-gateway-community
set routing-instances macvrf44-vlan-based vtep-source-interface lo0.0
set routing-instances macvrf44-vlan-based instance-type mac-vrf
set routing-instances macvrf44-vlan-based service-type vlan-based
set routing-instances macvrf44-vlan-based route-distinguisher 172.16.7.52:2
set routing-instances macvrf44-vlan-based vrf-target target:1:7777
set routing-instances macvrf44-vlan-based vlans v101 vlan-id 101
set routing-instances macvrf44-vlan-based vlans v101 interface ae0.101 ← aggregated interface multihomed with leaf5
set routing-instances macvrf44-vlan-based vlans v101 vxlan vni 50101

### macvrf55 v102 used to connect east1 firewall with west7 using L2 EVPN-VXLAN
set routing-instances macvrf55-vlan-based protocols evpn extended-vni-list 50102
set routing-instances macvrf55-vlan-based protocols evpn encapsulation vxlan
set routing-instances macvrf55-vlan-based protocols evpn default-gateway no-gateway-community
set routing-instances macvrf55-vlan-based vtep-source-interface lo0.0
set routing-instances macvrf55-vlan-based instance-type mac-vrf
set routing-instances macvrf55-vlan-based service-type vlan-based
set routing-instances macvrf55-vlan-based route-distinguisher 172.16.7.52:3
set routing-instances macvrf55-vlan-based vrf-target target:1:7778
set routing-instances macvrf55-vlan-based vlans v102 vlan-id 102
set routing-instances macvrf55-vlan-based vlans v102 interface ae0.102 ← aggregated interface multihomed with leaf5
set routing-instances macvrf55-vlan-based vlans v102 vxlan vni 50102

### when connecting the east3 servers with v103/v104 the following macvrf66 configuration was used
set routing-instances macvrf66-vlan-aware protocols evpn extended-vni-list 50103
set routing-instances macvrf66-vlan-aware protocols evpn extended-vni-list 50104
set routing-instances macvrf66-vlan-aware protocols evpn encapsulation vxlan
set routing-instances macvrf66-vlan-aware protocols evpn default-gateway no-gateway-community
set routing-instances macvrf66-vlan-aware vtep-source-interface lo0.0
set routing-instances macvrf66-vlan-aware instance-type mac-vrf
set routing-instances macvrf66-vlan-aware service-type vlan-aware
set routing-instances macvrf66-vlan-aware interface et-0/0/51.0 ← enterprise-style interface east3 servers
set routing-instances macvrf66-vlan-aware route-distinguisher 172.16.7.52:4
set routing-instances macvrf66-vlan-aware vrf-target target:1:7779
set routing-instances macvrf66-vlan-aware vlans v103 vlan-id 103
set routing-instances macvrf66-vlan-aware vlans v103 l3-interface irb.103
set routing-instances macvrf66-vlan-aware vlans v103 vxlan vni 50103
set routing-instances macvrf66-vlan-aware vlans v104 vlan-id 104
set routing-instances macvrf66-vlan-aware vlans v104 l3-interface irb.104
set routing-instances macvrf66-vlan-aware vlans v104 vxlan vni 50104
```

In the topology used in the present document, the leaf2 was still running the legacy default-switch vlan-aware EVPN instance available on QFX5120/QFX5110/5100 and integrated with the other leaf devices running the Junos 20.4 (or higher) release, by enabling the same per VLAN/VNI route-target – this is detailed later in the present paper in the miscellaneous topics dedicated section.

MAC-VRF server interfaces provisioning

This section covers the main configurations used when provisioning the server connect interfaces. The configurations use the service-provider style, with multiple units per physical or aggregated port, for vlan-based services. When the interface is not aggregated the ESI configuration is not needed. By default, such interfaces get the ESI value of "0" in the EVPN control plane.

ESI-LAG interface requires the same ESI type-0 hard coded values on all leaf devices connected to the same multihomed server or switch. In this example the leaf1/leaf2 pair share the same ESI when connecting to west2 switch while the leaf5/6 pair use a different ESI value for their LAG connection to the east1 firewall.

Note: *In the context of a MAC-VRF, interfaces using the sp-style are listed inside the instance at the vlan level and when using enterprise style the interface is listed directly inside the mac-vrf and not at the VLAN hierarchy.*

Leaf1 server facing interfaces:

```
## west7 server connected – sp-style (service-provider) interface config
set interfaces et-0/0/51 description sp-style
set interfaces et-0/0/51 flexible-vlan-tagging
set interfaces et-0/0/51 mtu 9100
set interfaces et-0/0/51 encapsulation extended-vlan-bridge
set interfaces et-0/0/51 unit 101 description mac-vrf44
set interfaces et-0/0/51 unit 101 vlan-id 101
set interfaces et-0/0/51 unit 102 description mac-vrf55
set interfaces et-0/0/51 unit 102 vlan-id 102

## west2 server connected ep-style (enterprise) interface config
set interfaces ae0 description ep-style
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 mtu 9100
set interfaces ae0 encapsulation extended-vlan-bridge
set interfaces ae0 esi 00:02:02:02:02:02:02:02:00:02 ← 10 bytes EVPN ESI segment value shared with leaf2 connected to
west2
set interfaces ae0 esi all-active
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp system-id 00:00:00:02:00:02 ← LACP value shared with leaf2 when
multihomed
set interfaces ae0 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae0 unit 0 family ethernet-switching vlan members 103-104

set interfaces et-0/0/50 ether-options 802.3ad ae0 ← associating ESI-LAG AE0 aggregate member interface
set chassis aggregated-devices ethernet device-count 6 ← higher number of AE interfaces can be set (for example 48)

## IRB.VGA anycast gateway IP interfaces for v103/v104 – can also use the IRB.VGA style if VIP and unique IP is needed
per IRB
set interfaces irb unit 103 family inet address 10.10.103.254/24
set interfaces irb unit 103 mac 00:00:5e:00:53:aa
set interfaces irb unit 104 family inet address 10.10.104.254/24
set interfaces irb unit 104 mac 00:00:5e:00:53:aa
```

Leaf2 interface configuration

```
## west2 servers connected interface configuration , enterprise-style
set interfaces ae0 mtu 9100
set interfaces ae0 esi 00:02:02:02:02:02:02:02:00:02
set interfaces ae0 esi all-active
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp system-id 00:00:00:02:00:02
set interfaces ae0 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae0 unit 0 family ethernet-switching vlan members 103-104




### the LAG member interface configuration


set interfaces et-0/0/4 ether-options 802.3ad ae0 🡸 associate the member physical interface with the LAG interface AE0
set chassis aggregated-devices ethernet device-count 6

## IP anycast gateway IRB configuration for v103/v104 from mac-vrf66

set interfaces irb unit 103 family inet address 10.10.103.254/24
set interfaces irb unit 103 mac 00:00:5e:00:53:aa
set interfaces irb unit 104 family inet address 10.10.104.254/24
set interfaces irb unit 104 mac 00:00:5e:00:53:aa
```

Leaf5 interface configuration to connect to the firewall and offer the first hop gateway for west7 servers:

```
## the east1 firewall connected LAG interface using sp-style configuration with multiple units

set interfaces ae0 description sp-style
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 mtu 9100
set interfaces ae0 encapsulation extended-vlan-bridge
set interfaces ae0 esi 00:02:02:02:02:02:02:02:00:06 🡸 ESI 10 bytes value shared with leaf6 connected to external GW
set interfaces ae0 esi all-active
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp system-id 00:00:02:02:00:06 🡸 same value shared with leaf6
set interfaces ae0 unit 101 description mac-vrf44
set interfaces ae0 unit 101 vlan-id 101
set interfaces ae0 unit 102 description mac-vrf55
set interfaces ae0 unit 102 vlan-id 102

### the LAG member interface configuration
set interfaces et-0/0/50 ether-options 802.3ad ae0 🡸 associating the physical interface with the ESI-LAG
set chassis aggregated-devices ethernet device-count 6
```

Worth noting is that the type-0 EVPN ESI value must always start with 00:0x:xx:xx:xx:xx:xx:xx:xx:xx and should not be all zeros as this is a reserved value for single homed interfaces.

Leaf6 interface configuration:

```
## the east1 firewall connected LAG interface
set interfaces ae0 description sp-style
set interfaces ae0 flexible-vlan-tagging
set interfaces ae0 mtu 9100
set interfaces ae0 encapsulation extended-vlan-bridge
set interfaces ae0 esi 00:02:02:02:02:02:02:02:00:06 ← ESI value shared with the leaf5 connected to firewall east1
set interfaces ae0 esi all-active
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp system-id 00:00:02:02:00:06 ← same value shared with leaf5
set interfaces ae0 unit 101 description mac-vrf44
set interfaces ae0 unit 101 vlan-id 101
set interfaces ae0 unit 102 description mac-vrf55
set interfaces ae0 unit 102 vlan-id 102

## east1 LAG interface member physical interface configuration
set interfaces et-0/0/50 ether-options 802.3ad ae0
set chassis aggregated-devices ethernet device-count 6

## Single homed east3 servers interface configuration

set interfaces et-0/0/51 description mac-vrf66-ep-style
set interfaces et-0/0/51 unit 0 family ethernet-switching interface-mode trunk
set interfaces et-0/0/51 unit 0 family ethernet-switching vlan members 103-104

## the anycast first hop IP gateway configuration
set interfaces irb unit 103 family inet address 10.10.103.254/24
set interfaces irb unit 103 mac 00:00:5e:00:53:aa
set interfaces irb unit 104 family inet address 10.10.104.254/24
set interfaces irb unit 104 mac 00:00:5e:00:53:aa
```

BGP Underlay and Overlay configuration

For reference purposes the leaf1 and spine1 device underlay and overlay BGP peering configurations are provided. Note that for the spine 1 device the *cluster* option is needed to enable route-reflection in the overlay.

Leaf1 BGP peering used for underlay (to advertise and learn fabric device loopback addresses) and the overlay peering that supports EVPN route exchange.

```
##  eBGP peering used for the advertisement of local loopback and reception of remote loopbacks (used for VXLAN RVTEP
## connect )
set protocols bgp group underlay type external
set protocols bgp group underlay export my_underlay_export ← exports local loopback IP /32 prefix
set protocols bgp group underlay local-as 64552 ← unique ASN per node for underlay peering
set protocols bgp group underlay multipath multiple-as ← for underlay IP-ECMP purposes
set protocols bgp group underlay neighbor 192.168.15.1 peer-as 64554 ← underlay peerings are using eBGP
set protocols bgp group underlay neighbor 192.168.16.1 peer-as 64555 ← underlay peerings are using eBGP

## iBGP peering with spines route reflectors used for EVPN signalization
set protocols bgp group overlay type internal
set protocols bgp group overlay local-address 172.16.7.7 ← local loopback address from global routing-table
set protocols bgp group overlay family evpn signaling
set protocols bgp group overlay local-as 64512 ← overlay peerings common BGP ASN on all nodes at the fabric
set protocols bgp group overlay multipath
set protocols bgp group overlay bfd-liveness-detection minimum-interval 350
set protocols bgp group overlay bfd-liveness-detection multiplier 3
set protocols bgp group overlay neighbor 172.16.7.11
set protocols bgp group overlay neighbor 172.16.7.12

## policy-statement used in BGP to advertise the local loopback address into the underlay routing

set policy-options policy-statement my_underlay_export term term1 from route-filter 172.16.7.0/24 prefix-length-range /32-/32
set policy-options policy-statement my_underlay_export term term1 then accept

## interfaces used for the establishment of the BGP peerings
## uplink interfaces connected to the spines spine1/spine2 used for underlay eBGP peering
set interfaces et-0/0/48 mtu 9216
set interfaces et-0/0/48 unit 0 family inet address 192.168.15.2/24
set interfaces et-0/0/49 mtu 9216
set interfaces et-0/0/49 unit 0 family inet address 192.168.16.2/24

## loopback interface from the global routing table used for overlay iBGP peering with spines route-reflectors
set interfaces lo0 unit 0 family inet address 172.16.7.7/32 primary
set interfaces lo0 unit 0 family inet address 172.16.7.7/32 preferred
```

The spine1 BGP peerings and route policy needed is shown below. Note the use of the cluster option to evoke route reflection functionality for the overlay peering group is needed only if iBGP overlay is used. Use the *multihop no-nexthop-change* option in case of the overlay eBGP peering.

```
## here’s the configuration used for the underlay eBGP peering at the spine1 level with all the leaf devices from the
## fabric
set protocols bgp group myunderlay type external
set protocols bgp group myunderlay export my_underlay_export ← to advertise the underlay loopback IP@
set protocols bgp group myunderlay local-as 64554 ← unique local underlay ASN
set protocols bgp group myunderlay multipath multiple-as ← enabled for IP ECMP purposes of the underlay
set protocols bgp group myunderlay neighbor 192.168.15.2 peer-as 64552 ← leaf1
set protocols bgp group myunderlay neighbor 192.168.220.2 peer-as 65051 ← leaf5
set protocols bgp group myunderlay neighbor 192.168.221.2 peer-as 65052 ← leaf6
set protocols bgp group myunderlay neighbor 192.168.17.2 peer-as 64553 ← leaf2

set policy-options policy-statement my_underlay_export term term1 from route-filter 172.16.7.0/24 prefix-length-range /32-/32
set policy-options policy-statement my_underlay_export term term1 then accept

## the spine level overlay iBGP config is including all leaf devices loopback IP peering addresses and clustering for
## reflection

set protocols bgp group myoverlay2 type internal
set protocols bgp group myoverlay2 local-address 172.16.7.11
set protocols bgp group myoverlay2 family evpn signaling
set protocols bgp group myoverlay2 cluster 172.16.7.11 ← in order to reflect the EVPN routes between the leaf
set protocols bgp group myoverlay2 local-as 64512 ← same overlay ASN for iBGP peering with leafs
set protocols bgp group myoverlay2 multipath
set protocols bgp group myoverlay2 neighbor 172.16.7.7 ← leaf1
set protocols bgp group myoverlay2 neighbor 172.16.7.51 ← leaf5
set protocols bgp group myoverlay2 neighbor 172.16.7.52 ← leaf6
set protocols bgp group myoverlay2 neighbor 172.16.7.8 ← leaf2
set protocols bgp group myoverlay2 vpn-apply-export ← used mainly when any export policy-statement is added later for
## example during the maintenance of the given spine
```

The spine2 BGP configuration is very similar. It uses a unique underlay autonomous system number and the same overlay autonomous system number. The cluster ID is set to the local loopback address of spine2.

Verification of the EVPN MAC-VRF

This section provides information on how to verify proper MAC-VRF operation for the example discussed in this paper as shown in Figure 9.

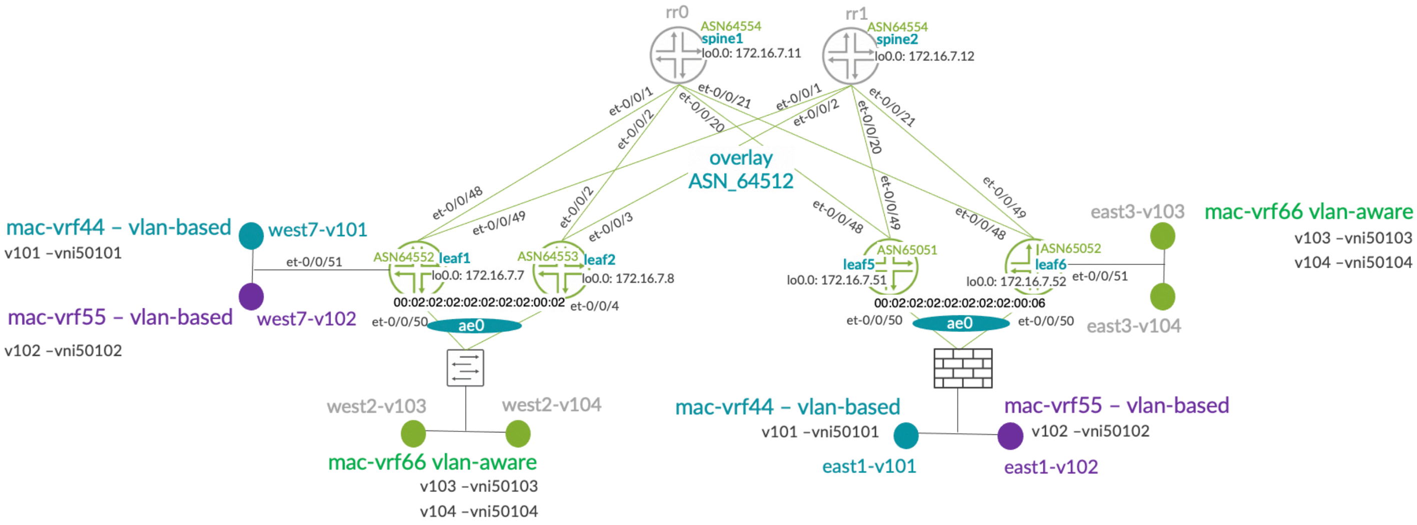


Figure 9 EVPN-VXLAN MAC-VRF Reference Topology

The first step in confirming EVPN-VXLAN operation is the verification of the BGP underlay and overlay peerings. Note that the overlay peerings relay on working underlay peering, and that problems in the overlay peerings result in missing EVPN route information that prevents proper VXLAN tunnel establishment:

```
root@leaf1# run show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 2 Peers: 4 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.evpn.0
                     108        54          0          0          0          0
inet.0
                      11         6          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/Received/Accepted/Damped...
172.16.7.11           64512       2451       1760       0       0    13:01:07 Establ 🡐 overlay iBGP peering to spine1
  bgp.evpn.0: 54/54/54/0
  macvrf44-vlan-based.evpn.0: 14/14/14/0 🡐 each MAC-VRF will be reported with the total number of EVPN routes
  macvrf55-vlan-based.evpn.0: 12/12/12/0
  macvrf66-vlan-aware.evpn.0: 27/27/27/0
  __default_evpn__.evpn.0: 3/3/3/0
172.16.7.12           64512         49         23       0       1        1:14 Establ 🡐 overlay iBGP peering to spine2
  bgp.evpn.0: 0/54/54/0
  macvrf44-vlan-based.evpn.0: 0/14/14/0
  macvrf55-vlan-based.evpn.0: 0/12/12/0
  macvrf66-vlan-aware.evpn.0: 0/27/27/0
  __default_evpn__.evpn.0: 0/3/3/0
192.168.15.1          64554       1697       1735       0       0    13:01:09 Establ 🡐 underlay eBGP peering to spine1
  inet.0: 5/6/6/0
192.168.16.1          64555         15         12       0       0        1:35 Establ 🡐 underlay eBGP peering to spine2
  inet.0: 1/5/5/0
root@leaf1#
```

With proper BGP peering confirmed the VXLAN tunnel endpoint information can be verified. Output from leaf6 confirms the expected remote VTEP connectivity with per MAC-VRF forwarding.

```
{master:0}[edit]
root@leaf6# run show mac-vrf forwarding vxlan-tunnel-end-point remote
Logical System Name       Id  SVTEP-IP         IFL    L3-Idx    SVTEP-Mode
<default>                 0   172.16.7.52      lo0.0    0
 RVTEP-IP      IFL-Idx     Interface    NH-Id   RVTEP-Mode Flags
 172.16.7.7      582       vtep.32774   1810    RNVE    ← leaf1 shared tunnel
 172.16.7.8      581       vtep.32773   1798    RNVE    ← leaf2 shared tunnel
 172.16.7.51     580       vtep.32772   1797    RNVE    ← leaf5 shared tunnel
 RVTEP-IP         L2-RTT                   IFL-Idx   Interface    NH-Id   RVTEP-Mode Flags
 172.16.7.7       macvrf44-vlan-based      671645702 vtep.671645702 1810  RNVE  ← local per MAC-VRF tunnel IFL
    VNID          MC-Group-IP
    50101         0.0.0.0
 RVTEP-IP         L2-RTT                   IFL-Idx   Interface    NH-Id   RVTEP-Mode Flags
 172.16.7.51      macvrf44-vlan-based      671645700 vtep.671645700 1797  RNVE
    VNID          MC-Group-IP
    50101         0.0.0.0
 RVTEP-IP         L2-RTT                   IFL-Idx   Interface    NH-Id   RVTEP-Mode Flags
 172.16.7.7       macvrf55-vlan-based      671711238 vtep.671711238 1810  RNVE
    VNID          MC-Group-IP
    50102         0.0.0.0
 RVTEP-IP         L2-RTT                   IFL-Idx   Interface    NH-Id   RVTEP-Mode Flags
 172.16.7.51      macvrf55-vlan-based      671711236 vtep.671711236 1797  RNVE
    VNID          MC-Group-IP
    50102         0.0.0.0
 RVTEP-IP         L2-RTT                   IFL-Idx   Interface    NH-Id   RVTEP-Mode Flags
 172.16.7.7       macvrf66-vlan-aware      671776774 vtep.671776774 1810  RNVE
    VNID          MC-Group-IP
    50104         0.0.0.0
    50103         0.0.0.0
 RVTEP-IP         L2-RTT                   IFL-Idx   Interface    NH-Id   RVTEP-Mode Flags
 172.16.7.8       macvrf66-vlan-aware      671776773 vtep.671776773 1798  RNVE
    VNID          MC-Group-IP
    50104         0.0.0.0
    50103         0.0.0.0
 RVTEP-IP         L2-RTT                   IFL-Idx   Interface    NH-Id   RVTEP-Mode Flags
 172.16.7.51      macvrf66-vlan-aware      671776772 vtep.671776772 1797  RNVE
    VNID          MC-Group-IP
    50103         0.0.0.0
    50104         0.0.0.0
```

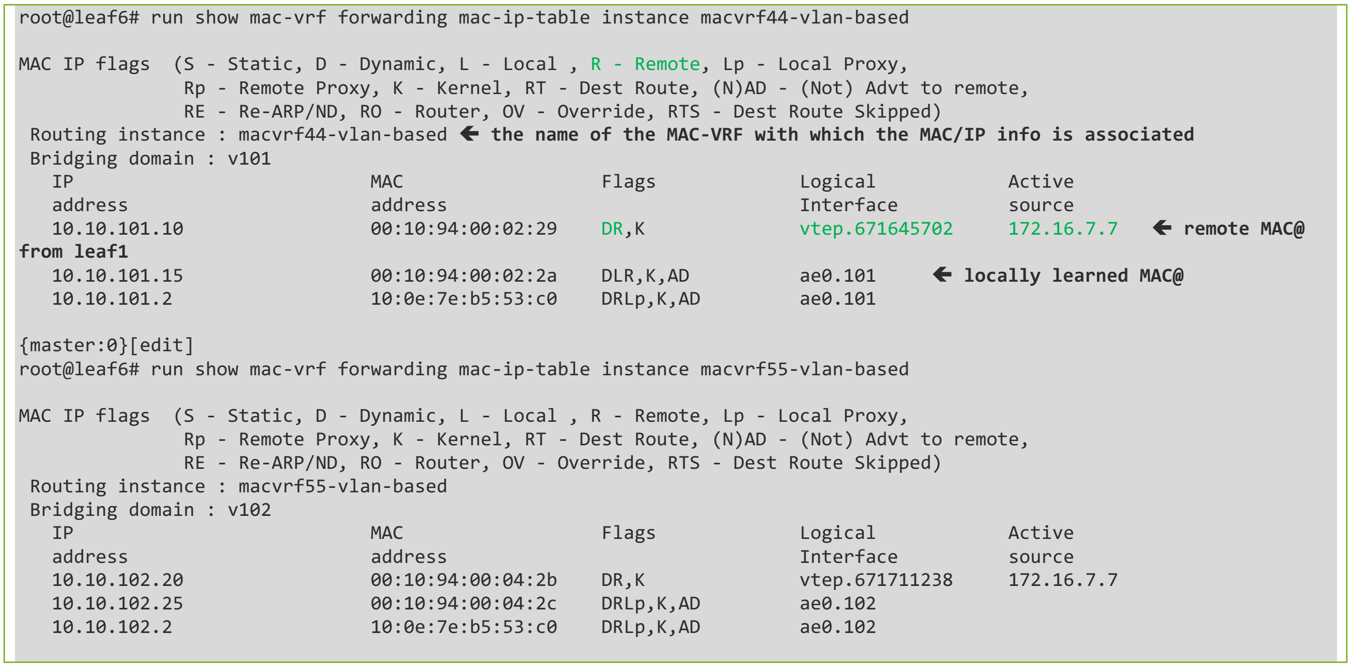
We move on to verify the L2 interface state for the MAC-VRF to ensure it has the expected forwarding state:

```
{master:0}[edit]
root@leaf6# run show mac-vrf forwarding interface ae0.101 ← verification is done per sp-style logical interface (IFL)
Routing Instance Name : macvrf44-vlan-based ← the name of the MAC-VRF is referenced for the given interface
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down,
                         MMAS - Mac-move action shutdown,  AS - Autostate-exclude enabled,
                         SCTL - shutdown by Storm-control, MI - MAC+IP limit hit)

Logical         Vlan                    TAG   MAC    MAC+IP STP         Logical         Tagging
interface       members                       limit  limit  state       interface flags
ae0.101                                       0      0                                  tagged
                v101                    101   294912 0      Forwarding                  tagged
```

```
{master:0}[edit]
root@leaf6# run show mac-vrf forwarding interface ae0.102
Routing Instance Name : macvrf55-vlan-based
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down,
                         MMAS - Mac-move action shutdown,  AS - Autostate-exclude enabled,
                         SCTL - shutdown by Storm-control, MI - MAC+IP limit hit)

Logical         Vlan                   TAG   MAC    MAC+IP STP         Logical          Tagging
interface       members                      limit  limit  state       interface flags
ae0.102                                      0      0                                   tagged
                v102                   102   294912 0      Forwarding                   tagged

{master:0}[edit]
root@leaf6# run show mac-vrf forwarding interface et-0/0/51
Routing Instance Name : macvrf66-vlan-aware
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down,
                         MMAS - Mac-move action shutdown,  AS - Autostate-exclude enabled,
                         SCTL - shutdown by Storm-control, MI - MAC+IP limit hit)

Logical         Vlan                   TAG   MAC    MAC+IP STP         Logical          Tagging
interface       members                      limit  limit  state       interface flags
et-0/0/51.0                                  294912 0                                   tagged
                v103                   103   294912 0      Forwarding                   tagged
                v104                   104   294912 0      Forwarding                   tagged

{master:0}[edit]
root@leaf6#
```

Use the mac-ip-table argument for more information about the dynamically learned local and remote MAC/IP entries in the MAC-VRF. The entries have various flags that can be used for more advanced troubleshooting.

Refer to the following documentation for additional information on the meaning of each flags:



```
root@leaf6# run show mac-vrf forwarding mac-ip-table instance macvrf44-vlan-based

MAC IP flags  (S - Static, D - Dynamic, L - Local , R - Remote, Lp - Local Proxy,
               Rp - Remote Proxy, K - Kernel, RT - Dest Route, (N)AD - (Not) Advt to remote,
               RE - Re-ARP/ND, RO - Router, OV - Override, RTS - Dest Route Skipped)
 Routing instance : macvrf44-vlan-based ← the name of the MAC-VRF with which the MAC/IP info is associated
 Bridging domain : v101
   IP                    MAC                  Flags             Logical            Active
   address               address                                Interface          source
   10.10.101.10          00:10:94:00:02:29    DR,K              vtep.671645702     172.16.7.7    ← remote MAC@
from leaf1
   10.10.101.15          00:10:94:00:02:2a    DLR,K,AD          ae0.101      ← locally learned MAC@
   10.10.101.2           10:0e:7e:b5:53:c0    DRLp,K,AD         ae0.101

{master:0}[edit]
root@leaf6# run show mac-vrf forwarding mac-ip-table instance macvrf55-vlan-based

MAC IP flags  (S - Static, D - Dynamic, L - Local , R - Remote, Lp - Local Proxy,
               Rp - Remote Proxy, K - Kernel, RT - Dest Route, (N)AD - (Not) Advt to remote,
               RE - Re-ARP/ND, RO - Router, OV - Override, RTS - Dest Route Skipped)
 Routing instance : macvrf55-vlan-based
 Bridging domain : v102
   IP                    MAC                  Flags             Logical            Active
   address               address                                Interface          source
   10.10.102.20          00:10:94:00:04:2b    DR,K              vtep.671711238     172.16.7.7
   10.10.102.25          00:10:94:00:04:2c    DRLp,K,AD         ae0.102
   10.10.102.2           10:0e:7e:b5:53:c0    DRLp,K,AD         ae0.102
```

```
{master:0}[edit]
root@leaf6#

##. Because the macvrf66 is having two vlans both of them will show the MAC address locally and remotely learned

root@leaf6# run show mac-vrf forwarding mac-ip-table instance macvrf66-vlan-aware
Dec 28 14:35:47

MAC IP flags  (S - Static, D - Dynamic, L - Local , R - Remote, Lp - Local Proxy,
               Rp - Remote Proxy, K - Kernel, RT - Dest Route, (N)AD - (Not) Advt to remote,
               RE - Re-ARP/ND, RO - Router, OV - Override, RTS - Dest Route Skipped)
 Routing instance : macvrf66-vlan-aware
 Bridging domain : v103
   IP                          MAC                  Flags             Logical           Active
   address                     address                                Interface         source
   10.10.103.254               00:00:5e:00:53:aa    DR,K,RT           vtep.671776774    172.16.7.7
   10.10.103.52                00:00:5e:00:53:bb    S,K               irb.103
   10.10.103.5                 00:01:99:00:01:05    DR,K,RT           esi.1801          00:02:02:02:02:02:02:02:00:02
   10.10.103.80                00:10:94:04:07:4a    DL,K,RT,AD        et-0/0/51.0
   10.10.103.70                00:10:94:07:07:2a    DR,K,RT           esi.1801          00:02:02:02:02:02:02:02:00:02

MAC IP flags  (S - Static, D - Dynamic, L - Local , R - Remote, Lp - Local Proxy,
               Rp - Remote Proxy, K - Kernel, RT - Dest Route, (N)AD - (Not) Advt to remote,
               RE - Re-ARP/ND, RO - Router, OV - Override, RTS - Dest Route Skipped)
 Routing instance : macvrf66-vlan-aware
 Bridging domain : v104
   IP                          MAC                  Flags             Logical           Active
   address                     address                                Interface         source
   10.10.104.254               00:00:5e:00:53:aa    DR,K,RT           vtep.671776774    172.16.7.7
   10.10.104.52                00:00:5e:00:53:bb    S,K               irb.104
   10.10.104.5                 00:01:99:00:02:05    DR,K,RT           esi.1801          00:02:02:02:02:02:02:02:00:02
   10.10.104.81                00:10:94:04:07:4b    DL,K,RT,AD        et-0/0/51.0
   10.10.104.71                00:10:94:08:08:2b    DR,K,RT           esi.1801          00:02:02:02:02:02:02:02:00:02

{master:0}[edit]
root@leaf6#

## To check the remoote VTEPs associated with the esi.1801
## Note there are two RVTEPs associated because the MAC address was learned via the ESI-LAG on leaf1/leaf2
root@leaf6# run show mac-vrf forwarding vxlan-tunnel-end-point esi | find 1801
00:02:02:02:02:02:02:02:00:02 macvrf66-vlan-aware      1801  524290  esi.1801           2      Aliasing
    RVTEP-IP             RVTEP-IFL      VENH     MASK-ID   FLAGS        MAC-COUNT
    172.16.7.7           vtep.671776774 1810     1         2            2 ← leaf1
    172.16.7.8           vtep.671776773 1798     0         2            4 ← leaf2
```

EVPN database verification is also useful to confirm if there's a difference between the EVPN database and the routing information Base (RIB) for a particular MAC-VRF. You can check the timestamps to know when a remote MAC address was learned.

```
root@leaf6# run show evpn database instance macvrf44-vlan-based
Instance: macvrf44-vlan-based
VLAN  DomainId  MAC address        Active source                  Timestamp        IP address
     50101      00:10:94:00:02:29  172.16.7.7                     Dec 28 02:07:34  10.10.101.10
     50101      00:10:94:00:02:2a  00:02:02:02:02:02:02:02:00:06  Dec 28 02:07:34  10.10.101.15
     50101      10:0e:7e:b5:53:c0  00:02:02:02:02:02:02:02:00:06  Dec 28 02:41:53  10.10.101.2

{master:0}[edit]
root@leaf6# run show evpn database instance macvrf55-vlan-based
Instance: macvrf55-vlan-based
VLAN  DomainId  MAC address        Active source                  Timestamp        IP address
     50102      00:10:94:00:04:2b  172.16.7.7                     Dec 28 02:07:34  10.10.102.20
     50102      00:10:94:00:04:2c  00:02:02:02:02:02:02:02:00:06  Dec 28 02:36:52  10.10.102.25
     50102      10:0e:7e:b5:53:c0  00:02:02:02:02:02:02:02:00:06  Dec 28 02:41:53  10.10.102.2

{master:0}[edit]
root@leaf6# run show evpn database instance macvrf66-vlan-aware
Instance: macvrf66-vlan-aware
VLAN  DomainId  MAC address        Active source                  Timestamp        IP address
     50103      00:00:5e:00:53:aa  172.16.7.7                     Dec 28 01:05:43  10.10.103.254
     50103      00:00:5e:00:53:bb  irb.103                        Dec 28 01:20:33  10.10.103.52
     50103      00:01:99:00:01:05  00:02:02:02:02:02:02:02:00:02  Dec 28 01:31:26  10.10.103.5
     50103      00:10:94:04:07:4a  et-0/0/51.0                    Dec 28 02:06:42
     50103      00:10:94:07:07:2a  00:02:02:02:02:02:02:02:00:02  Dec 28 02:06:39
     50104      00:00:5e:00:53:aa  172.16.7.7                     Dec 28 01:05:43  10.10.104.254
     50104      00:00:5e:00:53:bb  irb.104                        Dec 28 01:20:33  10.10.104.52
     50104      00:01:99:00:02:05  00:02:02:02:02:02:02:02:00:02  Dec 28 01:31:26  10.10.104.5
     50104      00:10:94:04:07:4b  et-0/0/51.0                    Dec 28 02:06:42
     50104      00:10:94:08:08:2b  00:02:02:02:02:02:02:02:00:02  Dec 28 02:06:39

{master:0}[edit]
root@leaf6#
```

We next check the mac-vrf44-vlan-based MAC-VRF route table to verify if a specific MAC address is present:

```
root@leaf6# run show route table macvrf44-vlan-based.evpn.0 evpn-mac-address 00:10:94:00:02:29
Dec 28 14:49:38

macvrf44-vlan-based.evpn.0: 16 destinations, 26 routes (16 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2:172.16.7.7:2::0::00:10:94:00:02:29/304 MAC/IP     🡸 corresponding MAC Type-2 route
                   *[BGP/170] 13:08:35, localpref 100, from 172.16.7.11. 🡸 received from spine1
                      AS path: I, validation-state: unverified
                    >  to 192.168.221.1 via et-0/0/48.0
                       to 192.168.223.1 via et-0/0/49.0
                    [BGP/170] 00:03:09, localpref 100, from 172.16.7.12 🡸 received from spine2
                      AS path: I, validation-state: unverified
                    >  to 192.168.221.1 via et-0/0/48.0
                       to 192.168.223.1 via et-0/0/49.0
2:172.16.7.7:2::0::00:10:94:00:02:29::10.10.101.10/304 MAC/IP
                   *[BGP/170] 12:42:04, localpref 100, from 172.16.7.11
                      AS path: I, validation-state: unverified
                    >  to 192.168.221.1 via et-0/0/48.0
                       to 192.168.223.1 via et-0/0/49.0
                    [BGP/170] 00:03:09, localpref 100, from 172.16.7.12
                      AS path: I, validation-state: unverified
                    >  to 192.168.221.1 via et-0/0/48.0 🡸 spine1
                       to 192.168.223.1 via et-0/0/49.0 🡸 spine2

root@leaf6#

## this can be filtered based on the local active-path – we can see the spine1 reflected EVPN routes are
installed

root@leaf6# run show route table macvrf44-vlan-based.evpn.0 evpn-mac-address 00:10:94:00:02:29 active-path
macvrf44-vlan-based.evpn.0: 16 destinations, 26 routes (16 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2:172.16.7.7:2::0::00:10:94:00:02:29/304 MAC/IP    🡸 EVPN type2 MAC route for the mac@ 00:10:94:00:02:29
                   *[BGP/170] 13:13:39, localpref 100, from 172.16.7.11
                      AS path: I, validation-state: unverified
                    >  to 192.168.221.1 via et-0/0/48.0 🡸 via spine1
                       to 192.168.223.1 via et-0/0/49.0 🡸 via spine2
```

```
2:172.16.7.7:2::0::00:10:94:00:02:29::10.10.101.10/304 MAC/IP ← EVPN type2 MAC-IP route for the
mac@00:10:94:00:02:29
                   *[BGP/170] 12:47:08, localpref 100, from 172.16.7.11
                      AS path: I, validation-state: unverified
                    >  to 192.168.221.1 via et-0/0/48.0
                       to 192.168.223.1 via et-0/0/49.0
```

Use the extensive argument to display additional statistics for a MAC-VRF instance. This allows you to confirm the EVPN route count, or to confirm the designated forwarder (DF) election status per segment:

```
root@leaf6# run show evpn instance extensive macvrf44-vlan-based
Instance: macvrf44-vlan-based
  Route Distinguisher: 172.16.7.52:2
  VLAN ID: 101
  Encapsulation type: VXLAN
  Duplicate MAC detection threshold: 5
  Duplicate MAC detection window: 180
  MAC database status                     Local  Remote
    MAC advertisements:                       2       3
    MAC+IP advertisements:                    2       3
    Default gateway MAC advertisements:       0       0
  Number of local interfaces: 2 (2 up)
    Interface name  ESI                            Mode             Status     AC-Role
    .local..8       00:00:00:00:00:00:00:00:00:00  single-homed     Up         Root
    ae0.101         00:02:02:02:02:02:02:02:00:06  all-active       Up         Root
  Number of IRB interfaces: 0 (0 up)
  Number of protect interfaces: 0
  Number of bridge domains: 1
    VLAN  Domain-ID Intfs/up   IRB-intf  Mode            MAC-sync IM-label  v4-SG-sync IM-core-NH v6-SG-sync IM-
core-NH Trans-ID
    101   50101        1  1              Extended        Enabled  50101     Disabled              Disabled
50101
  Number of neighbors: 2
    Address               MAC    MAC+IP        AD        IM        ES Leaf-label Remote-DCI-Peer
    172.16.7.7              1         1         0         1         0
    172.16.7.51             2         2         2         1         0
  Number of ethernet segments: 1
    ESI: 00:02:02:02:02:02:02:02:00:06
      Status: Resolved by IFL ae0.101
      Local interface: ae0.101, Status: Up/Forwarding
      Number of remote PEs connected: 1
        Remote-PE        MAC-label  Aliasing-label  Mode
        172.16.7.51      50101      0               all-active   ←  leaf5 shared segment
      DF Election Algorithm: MOD based
      Designated forwarder: 172.16.7.52
      Backup forwarder: 172.16.7.51
      Last designated forwarder update: Dec 28 01:41:12
  Router-ID: 172.16.7.52
  Source VTEP interface IP: 172.16.7.52
  SMET Forwarding: Disabled
```

EVPN-VXLAN MAC-VRF Miscellaneous Topics

This section details additional topics related to the MAC-VRF implementation. Specifically, VXLAN tunnel scaling, co-existence with the legacy default-switch EVI implementation, or the mapping of EVPN-VXLAN MAC-VRF to Type-5 instances. These topics are not discussed in conjunction with the example MAC-VRF topology. They are covered here for the sake of completeness.

MAC-VRF and VXLAN tunnel scaling

When multiple mac-vrf instances (AKA multiple EVPN instances) are provisioned on the same node the 20.4 Junos release can optimize the shared tunnel configuration by consolidating the number of next-hop entries needed to reach the same RVTEP. Use the below configuration option, supported as of Junos release 20.4, to optimize VXLAN tunnel scaling in the fabric:

```
set forwarding-options evpn-vxlan shared-tunnels
```

Consider a leaf QFX5120 device that is configured with multiple MAC-VRFs and is connected to a remote leaf that shares the same set of MAC-VRFs. By using *shared* VXLAN tunnels the set of MAC-VRFs associated with the RVTEP only requires a single underlay next-hop to be installed in the PFE of the leaf switch. This is shown in Figure 10. Here, even if Leaf3 (L3) advertises 3 different MAC-VRF Auto-Discovery (AD) routes to Leaf1 (L1), only one shared next hop is installed in the top of the rack switch’s PFE:

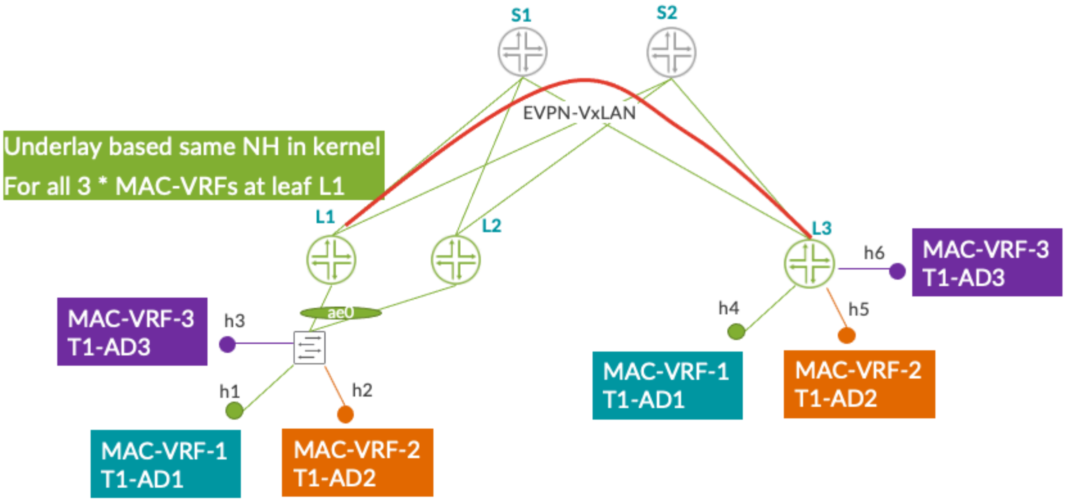


Figure 10 MAC-VRF Multi-EVI With Shared VXLAN Tunnels

Regarding the scaling capabilities, in some fabric scenarios distributing tenants and the associated MAC-VRFs at the edge of the fabric can consume additional switch TCAM resources – instead of using same EVI on all of the leaf devices, we can distribute the tenants on various leaf devices and optimize the way next-hops are installed for the remote VTEPs (RVTEPs) at the ToR switches. This is done by installing only the next hops for MAC-VRFs /VLAN-VNIs that are enabled at the specific top of rack switch.

Consider the case where leaf1 is not provisioned with VLAN BD55 in the context of a vlan-aware service-type. While not needed, the switch automatically installs a VXLAN tunnel to leaf2 because both switches share the same

EVPN AD route-targets for both VLAN-VNIs under the same MAC-VRF. This is not the case for mac-vrf's configured for a vlan-based service type, because in this case each instance originates a unique EVPN AD route, which prevents the establishment of an unneeded VXLAN tunnel between leaf1 and leaf2. This is shown in Figure 11.

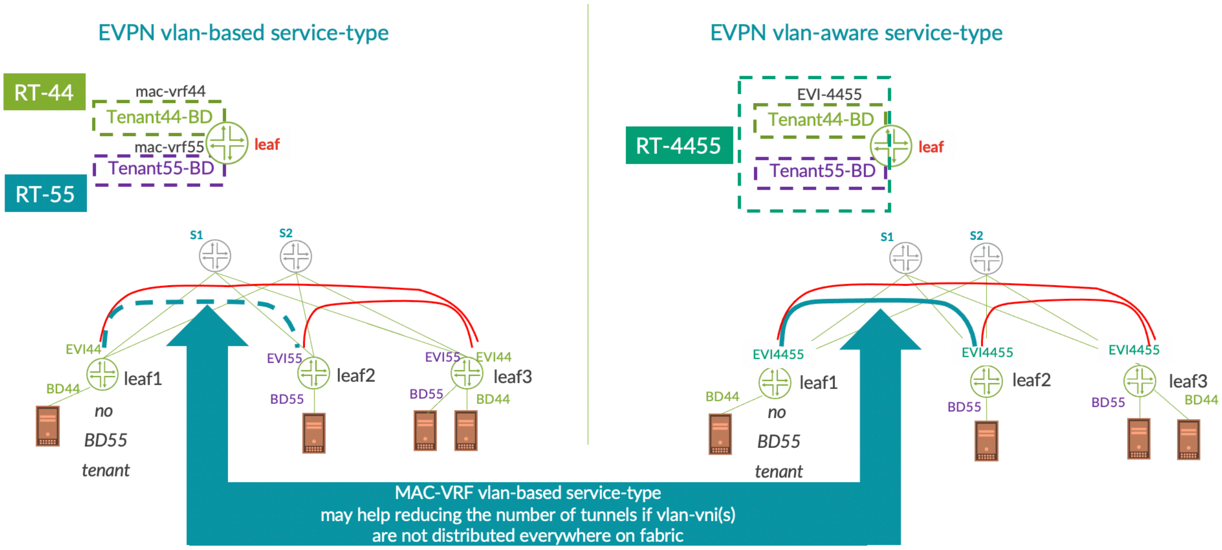


Figure 11 VLAN Based vs. VLAN Aware and VXLAN Tunnels

However, if both leaf1 and leaf2 are provisioned with same number of VLAN/VNIs, i.e., BD44 and BD55 are present on both leaves, then the same number of VXLAN tunnels are expected in both type of services – vlan-based and vlan-aware.

MAC-VRF and 'default-switch' EVI co-existence

When using the MAC-VRF feature supported in Junos Software release 20.4 there may be a requirement for interoperability with the QFX nodes running a default-switch EVI vlan-aware service in the same fabric.
For example, a QFX5120 leaf node enabled with MAC-VRF can support new workloads using a VLAN-based service type. At the same time, it can interoperate with the legacy default-switch infrastructure by using a different MAC-VRF name with a VLAN-aware service type. In our example topology, leaf2 is still running the older 18.4r2sX release and can interop with the other leaves that are enabled with mac-vrf66 using a VLAN-aware service-type as shown in Figure 12.

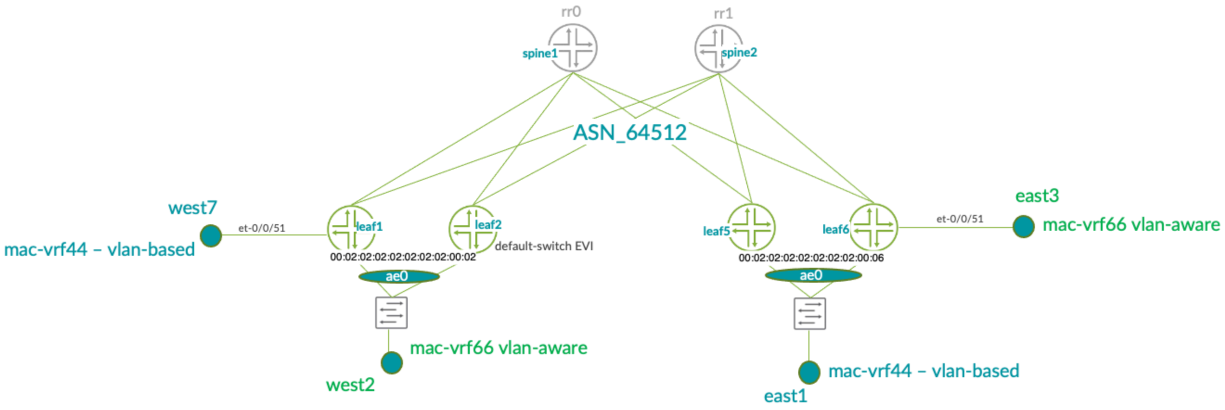


Figure 12 MAC-VRF Interoperability with the Default-Switch EVI

To interoperate with the other nodes running the MAC-VRF vlan-aware service-type, all the default-switch EVI VLAN/VNIs will be enabled with the same per VNI route-target at the protocols evpn hierarchy that matches the EVPN AD route-target.

Here's the configuration used at leaf2, which is running a previous Junos Software release and therefore supports only the legacy default-switch EVI configuration. With his configuration Leaf2 can interoperate with the rest of the fabric nodes that are enabled with the mac-vrf66 configuration shown in the previous section.

```
set protocols evpn vni-options vni 50103 vrf-target target:1:7779
set protocols evpn vni-options vni 50104 vrf-target target:1:7779
set protocols evpn encapsulation vxlan
set protocols evpn multicast-mode ingress-replication
set protocols evpn default-gateway no-gateway-community
set protocols evpn extended-vni-list all
set switch-options vtep-source-interface lo0.0
set switch-options route-distinguisher 172.16.7.8:1
set switch-options vrf-target target:1:7779
set vlans v103 vlan-id 103
set vlans v103 l3-interface irb.103
set vlans v103 vxlan vni 50103
set vlans v104 vlan-id 104
set vlans v104 l3-interface irb.104
set vlans v104 vxlan vni 50104
set interfaces ae0 mtu 9100
set interfaces ae0 esi 00:02:02:02:02:02:02:02:00:02
set interfaces ae0 esi all-active
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp system-id 00:00:02:02:00:04
set interfaces ae0 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae0 unit 0 family ethernet-switching vlan members 103-104
set interfaces ae0 unit 0 family ethernet-switching vlan members 251
```

EVPN L2 MAC-VRF Mapping to EVPN L3 Type5-VRF

When deploying new L2 MAC-VRF services various mapping options are supported when you also can provision EVPN Type-5 routing instances in case of ERB reference architecture.

In such cases a 1:1 mapping of each MAC-VRF L2 instance to a dedicated Type-5 L3 EVPN instance is recommended to maximize security. However, having multiple MAC-VRFs pointing to a common Type-5 instance can be still considered if the workloads need to be isolated only at the L2 level with the Type-5 instance used for Layer 3 connectivity. The four options of virtualization and mappings are highlighted in Figure 13 for the MAC-VRF service-type vlan-based and pure type-5 instances:

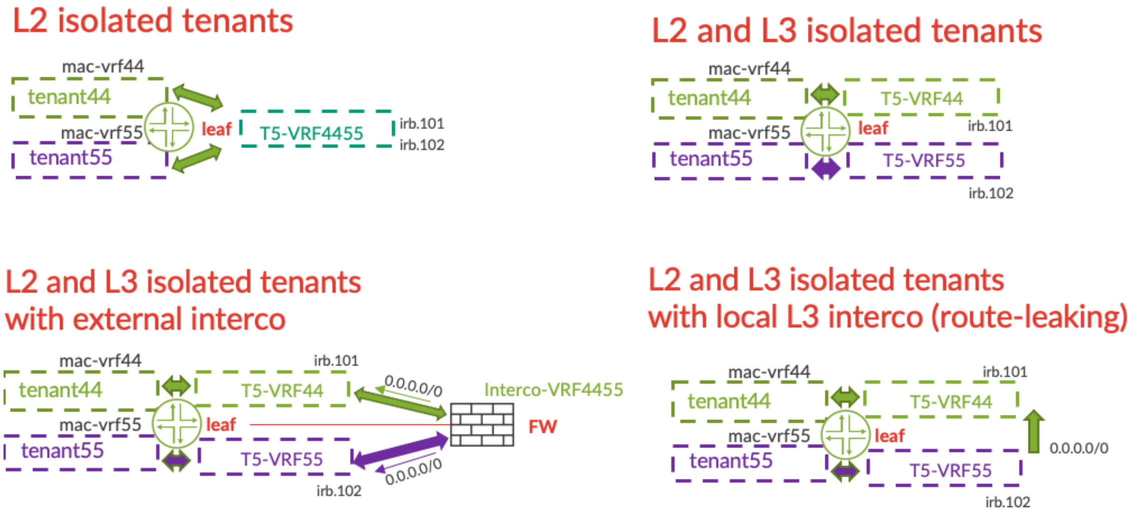


Figure 13 L2 MAC-VRF to L3 Type-5 VRF Mapping Options

A configuration for a pure Type-5 VRF L3 instance that is enabled for mac-vrf66 from the reference architecture is provided. The configuration enables Layer 3 prefix-advertisement and symmetric inter-IRB routing using type-5 instances:

```
set routing-instances T5-VRF1 routing-options static route 10.10.100.7/32 discard
set routing-instances T5-VRF1 routing-options multipath
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes advertise direct-nexthop
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes encapsulation vxlan
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes vni 1100
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes export my-t5-export-vrf1
set routing-instances T5-VRF1 instance-type vrf
set routing-instances T5-VRF1 interface irb.103
set routing-instances T5-VRF1 interface irb.104
set routing-instances T5-VRF1 interface lo0.1
set routing-instances T5-VRF1 route-distinguisher 172.16.7.7:100
set routing-instances T5-VRF1 vrf-target target:1100:1000
set routing-instances T5-VRF1 vrf-table-label

set policy-options policy-statement my-t5-export-vrf1 term term1 from route-filter 172.16.100.7/32 exact
set policy-options policy-statement my-t5-export-vrf1 term term1 from route-filter 10.10.100.7/32 exact
set policy-options policy-statement my-t5-export-vrf1 term term1 then accept
set policy-options policy-statement my-t5-export-vrf1 term term2 from route-filter 10.10.103.0/24 orlonger
set policy-options policy-statement my-t5-export-vrf1 term term2 from route-filter 10.10.104.0/24 orlonger
set policy-options policy-statement my-t5-export-vrf1 term term2 then accept
```

The VNI 1100 value in the type-5 instance is used for symmetric inter-irb routing, and for routing towards the subnets enabled at the other nodes in the fabric.
In case the leaf node is enabled with type-5 EVPN instance shown above, then the following forwarding-options are required to enhance scaling by allocating more TCAM resources to the overlay and to also enable IP-ECMP in the overlay:

```
set routing-options forwarding-table chained-composite-next-hop ingress evpn
set routing-options forwarding-table export LB
set forwarding-options evpn-vxlan shared-tunnels
set forwarding-options vxlan-routing overlay-ecmp
set forwarding-options vxlan-routing next-hop 32768
set forwarding-options vxlan-routing interface-num 8192

set policy-options policy-statement LB term term1 from protocol evpn
set policy-options policy-statement LB term term1 then load-balance per-packet
set policy-options policy-statement LB term term1 then accept
set policy-options policy-statement LB term term2 then load-balance per-packet
set policy-options policy-statement LB term term2 then accept
```

Sidenotes:

- Multiple VLAN-based MAC-VRFs support using the same physical port within an Edge-routed bridging (ERB) architecture is part of the Junos roadmap for QFX5120. When different server connect ports are mapped to the same or different vlan-based MAC-VRFs, then the ERB design can be still used. When deploying an Edge-routed bridging architecture (ERB) with first hop IP gateway at the leaf level with the 20.4 Junos release, the MAC-VRFs with a VLAN-based service type can be enabled in multiple MAC-VRFs, but on different server attachment ports using enterprise-style configuration. As a result, it's recommended that you use different server interfaces, or consider using a VLAN-aware service-type when same server port needs to be provisioned with multiple VLANs with ERB.
- Regarding the multicast support in the context of MAC-VRF and 20.4r2, it's limited to the default ingress-replication (IR). The official support for more advanced L2 multicast features such as SMET (RT-6) and Assisted-Replication (AR) is planned in the upcoming releases.
- To support the vlan-bundle EVPN-VXLAN service model on QFX5120 or QFX10K, the recommended design using 20.4r2 Junos release is Bridged-Overlay (BO).

Conclusion

The Juniper Networks MAC-VRF implementation for an EVPN-VXLAN data center fabric provides the following benefits:

- Additional network virtualization and multi-tenancy options.
  – aggregating or isolating VNIs at the control plane and data plane for better server workload control.
- Easier fabric automation with the unified Junos MAC-VRF EVPN configurations templates across multiple platforms (QFX, MX, PTX).
- Better interoperability with other vendors for various service-types. The same fabric can have switching devices from multiple vendors. When another vendor supports only a single service type then a dedicated virtual routing forwarding instance MAC-VRF can be enabled for interoperability.
- Better VXLAN tunnel distribution and flooding optimizations – when selected VLAN-VNIs are enabled at the leaf nodes the EVPN AD routes are only exchanged with the nodes provisioned with a shared set of VLANs.

The proposed configuration examples for MAC-VRF provides also a more modular and structured approach when planning new server connect operations in the data center.