EVPN Deployment Guide

Table of contents

Introduction

The intended audience of this guide is those who are planning for, deploying, or maintaining a Data Center network leveraging a VXLAN data-plane with an EVPN control-plane.

The Overview and Nomenclature sections of this guide are intended to serve as a reference for, and cover in detail, the VXLAN data-plane and EVPN control-plane protocols. It is recommended that the reader has a sound comprehension of these two technologies prior to planning and deployment.

The content found within the topology and deployment sections assumes that the reader is comfortable with VXLAN and EVPN concepts. As such, detail around configuration, deployment recommendations, and validation will be provided. If the reader encounters a topic or concept not well understood within the topology and deployment sections, it is recommended that they refer back to the Overview and Nomenclature sections of this document.

Software Version and Hardware Platforms

All configuration, verification output, and functionality detailed within this guide is based on the following software version and hardware platforms:

- EOS 4.21.5F
- 7280SR (R Series Platforms)

VXLAN Overview and Nomenclature

Virtual Extensible LAN (VXLAN) is a IP/UDP encapsulation methodology that enables the extension of Layer-2 broadcast domains over IP transport. This is achieved by effectively emulating that all endpoints within the broadcast domain are Layer 2 adjacent, with no intermediate hops between them. This is accomplished through the mapping of VLANs to VXLAN Network Identifiers (VNIs), and by defining a method to distribute Broadcast, Unknown Unicast and Multicast (BUM) traffic.

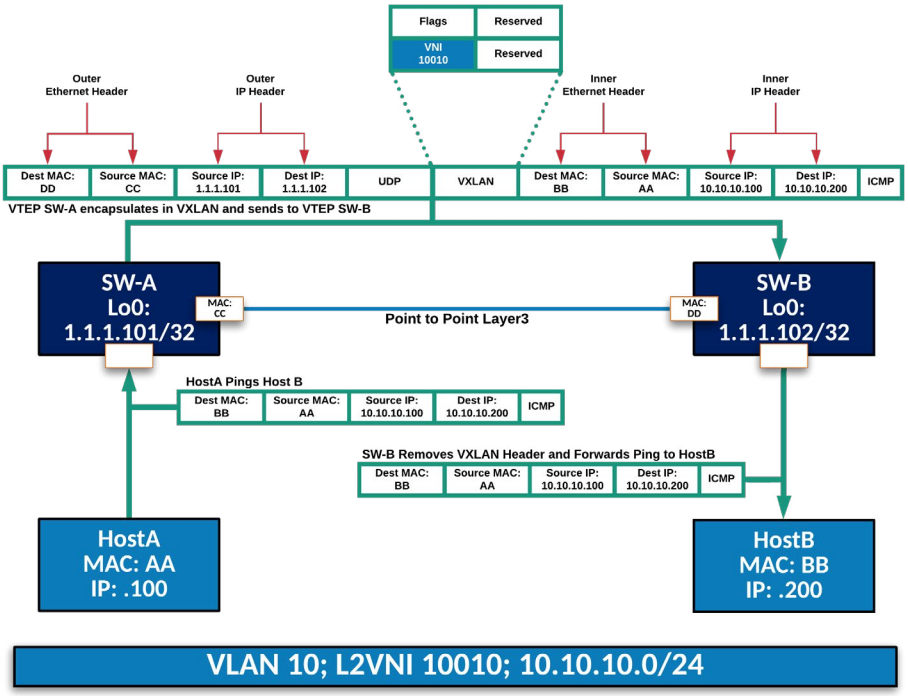
VXLAN is an industry standard protocol defined within RFC 7348. It requires IP transport with a sufficient MTU size between VXLAN Tunnel Endpoints (VTEPs). When VXLAN encapsulation is implemented, up to 54 bytes of overhead will be added to the original Ethernet frame. VTEPs must not fragment VXLAN packets. In order to facilitate this, VTEPs running EOS will set the Don't Fragment (DF) bit in the Outer IP header of the VXLAN packet. In doing so, it is recommended to have a minimum MTU of 1554 bytes enabled on the IP transport between VTEPs to support an MTU of 1500 bytes in the overlay. To support jumbo frames in the overlay, set the MTU to 9214 bytes.

Table 1: VXLAN nomenclature that will be used throughout this guide

VXLAN	Virtual Extensible LAN
VTEP	VXLAN Tunnel Endpoint
VNI / VNID	VXLAN Network Identifier
BUM Traffic	Broadcast, Unknown Unicast, Multicast Traffic
HER	Head End Replication
NVO	Network Virtualization Overlay
DCI	Data Center Interconnect

A VNI is a 24-bit value within the VXLAN header that enables the use of up to 16 million unique values to identify broadcast domain (L2VNI) or VRF (L3VNI) membership. For this example, we are focused on the L2VNI.

Consider the example below, which illustrates VXLAN Bridging Operations:



VXLAN Bridging:

1. HostA, located in VLAN 10 on SW-A, pings HostB
2. VTEP SW-A has VLAN 10 locally mapped to VNI 10010
3. VTEP SW-A performs VXLAN encapsulation, defining VNI 10010 within the VXLAN header
4. VTEP SW-B receives the VXLAN packet, inspects the header, and sees VNI 10010
5. VTEP SW-B has VNI 10010 locally mapped to VLAN 10
6. VTEP SW-B removes the VXLAN header, performs the lookup and forwarding operation within VLAN 10, and sends the packet to its destination of HostB

   The above example is meant to show the VXLAN encapsulation process, and illustrate how the VNI is a globally significant value. The VNI enables VTEPs to signal to other VTEPs which L2 (or L3) VPN to perform the lookup and forwarding operation in, once the VXLAN header is removed.

In the preceding example, how did HostA know HostB's MAC address? Additionally, how did SW-A know that Host-B was behind SW-B? This is where the distribution of Broadcast, Unknown Unicast and Multicast (BUM) traffic becomes important.

In order for HostA to resolve HostB's MAC address, it sends an ARP request. Once a VTEP receives a broadcast frame (in this case SW-A), it must ensure that all VTEPs with endpoints within the respective broadcast domain receive a copy of that frame.

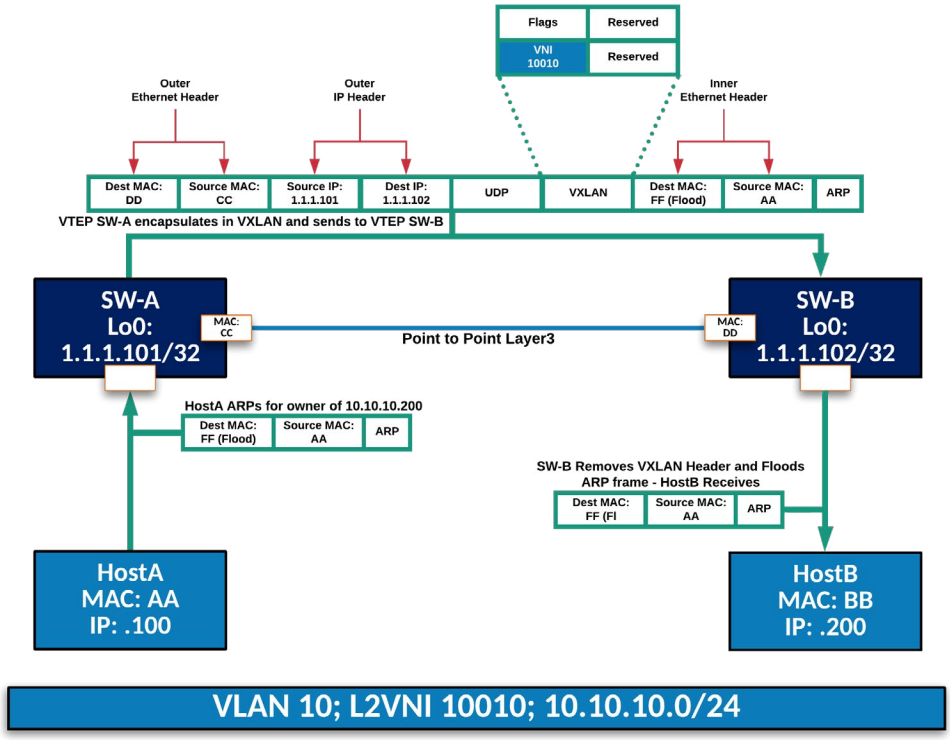
One method of BUM traffic distribution is multicast, in which VNIs are mapped to multicast groups on either a 1:1 or N:1 basis. VTEPs then become both Senders and Receivers for the multicast group(s) associated with their locally defined VNI(s). The multicast method of BUM traffic distribution will not be covered in further detail within this guide, as it is not supported in Arista EOS within the context of the EVPN control-plane. Additional information regarding multicast for VXLAN BUM Traffic distribution can be found within the VXLAN RFC 7348, Section 4.2: https://tools.ietf.org/html/rfc7348#section-4.2

Another method of BUM traffic distribution (and the focus of this guide) is Head End Replication (HER). With HER, every VTEP maintains a list of all other VTEPs that are interested in receiving BUM traffic for a given VNI. This is referred to as a 'Flood List", and can be either manually maintained via static entries, or dynamically populated via the EVPN control-plane.

In the HER model, once a BUM frame is received by a VTEP, it will unicast a unique copy of that frame to each respective VTEP within the flood list.

Consider the example below:

VXLAN BUM Traffic Distribution:



1. HostA, in VLAN 10, sends an ARP request asking what MAC address should be used as the Destination MAC when communicating with the 10.10.10.200 IP address
2. VTEP SW-A has VLAN 10 locally mapped to VNI 10010
3. VTEP SW-A sees that this is a broadcast frame that must be sent to all appropriate VTEPs
4. VTEP SW-A finds VTEP SW-B in the list of interested VTEPs (In this example, manually defined)
5. VTEP SW-A performs VXLAN encapsulation, defining VNI 10010 within the VXLAN header
6. VTEP SW-B receives the VXLAN packet, inspects the header, and sees VNI 10010
7. VTEP SW-B has VNI 10010 locally mapped to VLAN 10
8. VTEP SW-B removes the VXLAN header, and floods it within VLAN 10
9. HostB receives the frame and unicast replies to HostA, indicating that MAC address "BB" should be used for communication with IP 10.10.10.200

A key takeaway from the above example is that SW-A and SW-B only know about the location of HostA and HostB if those hosts are actively communicating. In this way, VXLAN follows traditional bridging semantics. Knowledge of endpoint location is entirely data-plane driven, with no control-plane to proactively learn host information and location. The need for a control-plane is where EVPN comes in.

EVPN Overview and Nomenclature

Ethernet Virtual Private Network (EVPN) is an industry standard protocol, defined in RFC 7432 (MPLS Encapsulation) and RFC 8365 (VXLAN, NVGRE and GENEVE Encapsulations).

EVPN is an address-family within BGP (AFI: 25, SAFI: 70), and provides a control-plane to enable L2VPN and L3VPN services, as well as additional features such as Active/Active L2VPN Multihoming. The protocol is data-plane encapsulation agnostic, with most implementations currently supporting either MPLS or VXLAN data-plane encapsulation.

When paired with a VXLAN data-plane encapsulation, EVPN allows operators to leverage the 24-bit VNID field in the VXLAN header to signal either L2VPN or L3VPN membership via an L2VNI or L3VNI, respectively.

The EVPN control-plane enables VTEPs to signal to all other VTEPs which VNIs they are interested in receiving BUM traffic for. This removes the need for manual intervention relating to BUM traffic distribution. Additionally, EVPN will enable VTEPs to proactively learn information about endpoints within the environment, including:

- MAC Address
- IP Address (/32 Host Route)
- Layer2 VNI (VLAN) Membership
- Layer3 VNI (VRF) Membership
- Which VTEP the endpoint resides behind
- Mobility tracking number (If endpoint moves behind different VTEPs)

All the information above, and more, is advertised via BGP to all other VTEPs as soon as a frame/packet from an endpoint is seen by a VTEP. Having this information about all endpoints enables techniques such as ARP suppression, which allows VTEPs to reply to ARP requests on behalf of the destination endpoint. VTEPs can perform this action because they have all of the latest information about hosts within the broadcast domain, learned via the EVPN control-plane.

Finally, EVPN also allows VTEPs to originate native IPv4 Unicast Prefixes, and signal to other VTEPs, via the L3VNI, which VRF (L3VPN) the IP prefix is a member of. There are five EVPN route-types currently defined within the standard, with additional types under development for future use cases. The focus of this guide will be Route-Types 2, 3 and 5. Each are listed below for reference, and are covered in greater detail within the deployment and validation sections.

Route-Type	Name	Purpose
2	MAC-IP	Layer 2 VPN: End-Host Information (MAC, IP, etc.)
3	IMET (Inclusive Multicast Ethernet Tag)	Signal desire to receive BUM traffic for a VNI
5	IP Prefix	Layer 3 VPN: IP Prefix and VRF Membership

Note that the multihoming solution described in this guide is based on Arista MLAG. EOS does support Route-Types 1 and 4 for EVPN based multihoming deployments, which will be covered in future versions of the deployment guide.

There are many terms frequently used when discussing EVPN control-plane design and deployment. Below are some of the most common terms, along with their associated descriptions. These will be used as a reference throughout the course of this guide:

VRF

- Layer3 Construct used to provide L3VPN services
- Enables Multi-Tenancy at Layer 3: Dedicated routing table per VRF
- Multiple VRFs can exist on a single physical device
- Inter-VRF communication (between VRFs on the same device) is not possible without additional configuration, such as route leaking or route-target import/export manipulation
- Synonymous with Routing Table

MAC-VRF

- Layer2 Construct used to provide L2VPN services
- Enables Multi-Tenancy at Layer 2: Dedicated control-plane and data-plane resources
- Allows for the creation of a distributed MAC Address table, where all VTEPs participating within the MAC-VRF learn the MAC addresses of all nodes within that MAC-VRF (VLAN)
- Synonymous with VLAN / Bridge Domain / Broadcast Domain

Route-Distinguisher (RD)

- Control-Plane mechanism that ensures all EVPN routes can be uniquely identified
- Globally significant value within the EVPN domain
    - Recommend to be unique per VRF (or MAC-VRF) on each VTEP
- *Important Note:* It is recommended that RDs be set to a globally unique value per-VRF on each VTEP, even if the VTEP is part of an MLAG domain. A globally unique Route-Distinguisher will contribute to improved convergence time, ensure proper ECMP, and assist in validation of route origination when troubleshooting.

Route-Target

- Control-Plane mechanism used to:
    - Signal, on *export* (origination of EVPN route), the value that should be used by the receiving VTEP when determining two things:
        - If the EVPN advertisement will be accepted
        - Which VRF (or MAC-VRF) the contents of the EVPN update should be imported into
    - Used, on *import* (receipt of an EVPN advertisement), to control which VRF (or MAC-VRF) to import the contents of a received EVPN update
- *Important Note:* Route-Targets are a globally significant value. In most cases, the import/export Route-Targets will match per VRF on all VTEPs. Multiple import/export Route-Targets can also be configured per VRF.

L2VNI

- Unique per MAC-VRF
- Data-Plane mechanism, signaled via the Control-Plane
- Encoded into all EVPN Type-2 (MAC/MAC-IP) and Type-3 (IMET) updates
- VNI that is present in the VXLAN header of the packet on-the-wire when performing VXLAN bridging between VTEPs
- Signals, via the Data-Plane to the receiving VTEP, which MAC Address Table (MAC-VRF) it should perform the lookup and forwarding operation in when looking at the inner-Ethernet header

L3VNI

- Unique per VRF
- Data-Plane mechanism, signaled via the Control-Plane
- Encoded into all EVPN Type-2 (MAC-IP) updates when operating in Symmetric IRB mode
- Encoded into all Type-5 (IP Prefix) updates
- VNI that is present in the VXLAN header of the packet on-the-wire when performing VXLAN routing between VTEPs
- Signals, via the Data-Plane to the receiving VTEP, which Routing Table (VRF) it should perform the lookup and forwarding operation in when looking at the inner-IP headers

Symmetric and Asymmetric Integrated Routing and Bridging (IRB), and the differences between them, are important concepts to understand in the process of planning and deploying VXLAN/EVPN in the Data Center. The characteristics of each respective IRB methodology can be found below:

Asymmetric IRB

- Does not require VRF(s), or L3VNI-to-VRF Mappings. L3 Multi-tenancy enabled through VRF-Lite
- Inter-Vlan routing occurs on the first hop router (local VTEP), followed by VXLAN bridging towards the destination using L2VNI
- Routing into destination subnet occurs on ingress VTEP
- To ensure reachability between subnets, all VLANs/Subnets must exist on all VTEPs (higher state requirements)
- VTEPs will not maintain host routes for advertised hosts within the EVPN domain
- Remote ARP entries maintained in hardware*

Symmetric IRB

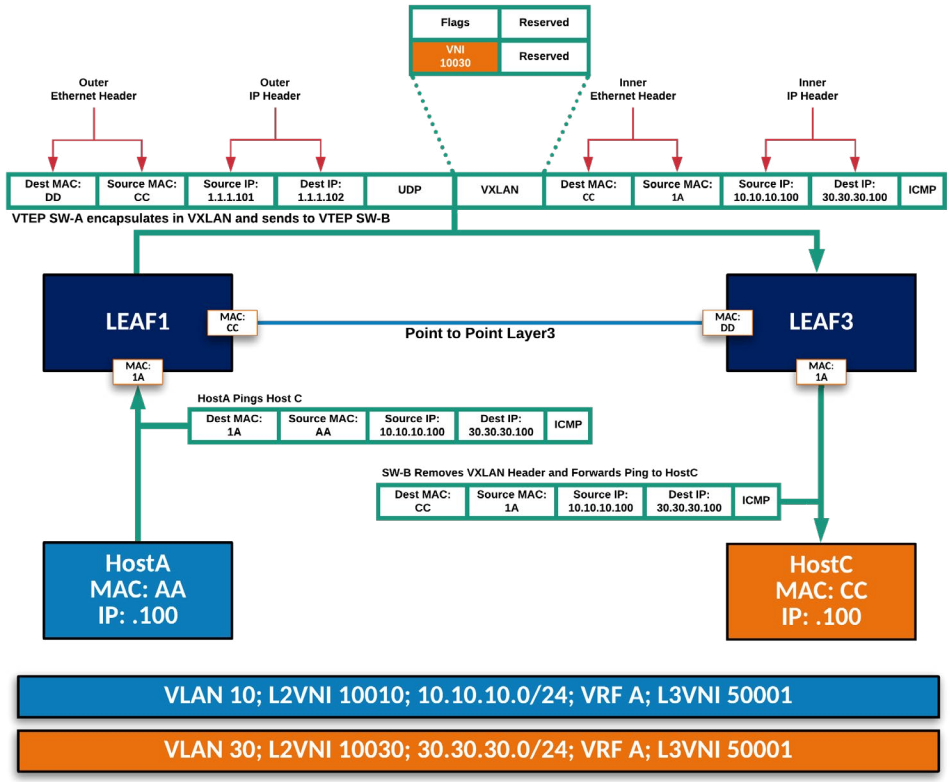
- Requires VRF(s), and L3VNI-to-VRF Mappings
- L3VNI used for VXLAN routing between VTEPs
- Makes use of "EvpnRouterMac" in MAC-IP (Type-2) and IP-Prefix (Type-5) routes
- Routing into destination subnet occurs on egress VTEP
- Placement of VLANs/Subnets can be scoped accordingly (lower state requirements)
- VTEPs will maintain host routes for all advertised hosts within the EVPN domain
- Optimized hardware scale - All remote ARP entries maintained in software

```
* Scale improvements to Asymmetric IRB can be obtained via selective ARP install:
https://eos.arista.com/eos-4-21-3f/provide-user-control-of-selective-arp/
```

Note: In a Symmetric IRB model, the origination of host routes can be disabled on a per MAC-VRF basis through the use of the 'no redistribute host-route' command under the MAC-VRF within BGP. This disables the automatic origination of Dual-VNI (L2VNI/L3VNI) Type-2 Routes for discovered hosts. Note that disabling could result in sub-optimal traffic forwarding traversing the DC Leaf-Spine fabric twice.

To better illustrate the concepts and operations of Asymmetric and Symmetric IRB, consider the examples below:

Asymmetric IRB



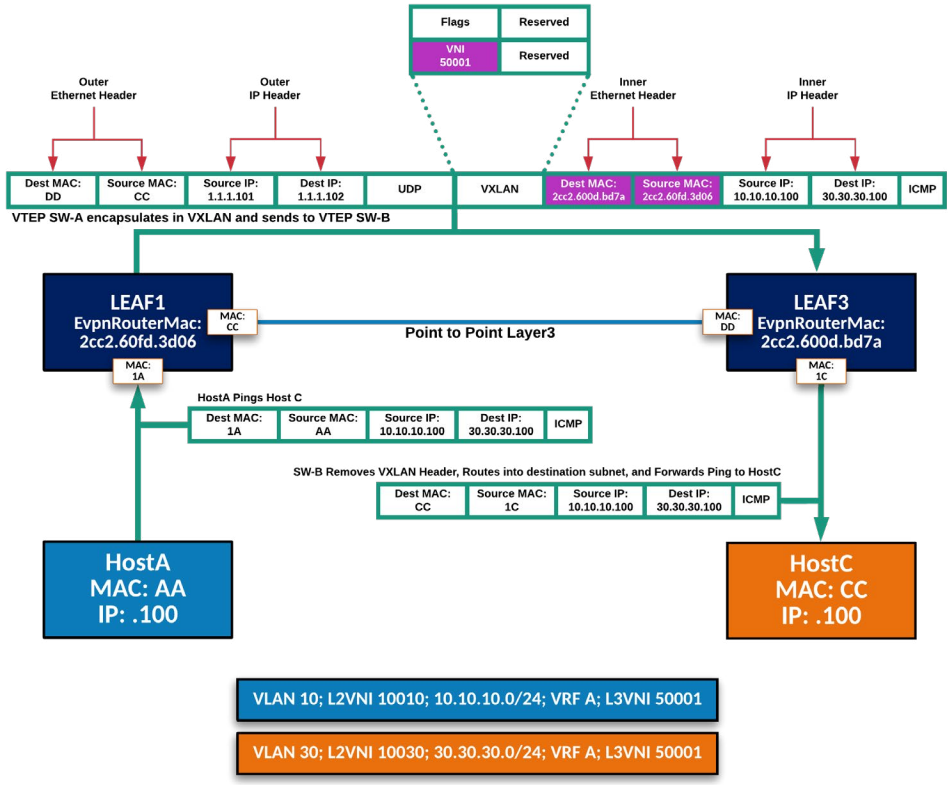
1. HostA, in VLAN 10, sends a ping to HostC, in VLAN 30
   a. This is bridged to it's default gateway (LEAF1)
2. LEAF1 sees destination IP address of 30.30.30.100, which can be reached via a directly connected interface (VLAN 30). LEAF1 routes into this subnet via interface VLAN 30
3. LEAF1, through a Type-2 MAC-IP route originated from LEAF3, knows the destination MAC address of HostC, as well as the L2VNI it should use in the VXLAN header
4. Using this information, LEAF1 performs VXLAN bridging towards egress VTEP LEAF3
   b. LEAF1 uses an L2VNI of 10030 in the VXLAN header
5. LEAF3 receives the VXLAN packet, sees the L2VNI of 10030, and knows that it will perform lookup/forwarding in VLAN 30 MAC-VRF (MAC Address-Table)
6. LEAF3 removes the VXLAN header, inspects the inner-Ethernet header, and forwards the packet towards HostC

Notes:
In this Asymmetric IRB example, LEAF1 and LEAF3 both have VLAN10 and VLAN30 locally configured and mapped to VNIs. They both have SVIs locally configured for each respective VLAN.

For bi-directional communication between HostA and HostC, L2VNI 10030 is used towards LEAF3, and L2VNI 10010 is used towards LEAF1

The inner-Ethernet header needed to be re-written because we crossed a routed boundary from VLAN10 to VLAN30. LEAF1 is using the MAC address associated with interface VLAN30 as the source (in this case "1A").

Symmetric IRB



1. HostA, in VLAN 10, sends a ping to HostC, in VLAN 30
   a. This is bridged to it's default gateway (LEAF1)
2. LEAF1 sees destination IP address of 30.30.30.100, and has a /32 host route in VRF RED to this destination, via a MAC-IP route originated from LEAF3
   a. This MAC-IP route contains the L3VNI to be used in the VXLAN header, and EvpnRouterMac to use as Dest MAC on inner-Ethernet header

3. Using this information, LEAF1 performs VXLAN routing towards egress VTEP LEAF3
    a. LEAF1 uses a L3VNI of 50001 in the VXLAN header
    b. LEAF1 re-writes the inner-Ethernet header to have:
        i. Destination MAC of EvpnRouterMac of LEAF3
        ii. Source MAC of EvpnRouterMac of LEAF1
    c. This is to ensure that once LEAF3 removes the VXLAN header, it will process/route the native IP packet accordingly
4. LEAF3 receives the VXLAN packet, sees the L3VNI of 50001, and knows that it will perform lookup/forwarding in VRF RED
5. LEAF3 removes the VXLAN header, and sees it's MAC as the destination in the Ethernet header. It processes and routes this packet into the proper destination subnet within VRF RED
6. The packet is then bridged towards HostC in VLAN 30

```
Note: In this Symmetric IRB example, LEAF1 does not have VLAN30 configured, and LEAF3
does not have VLAN 10 configured. However, using the L3VNI, end-to-end reachability
exists. L3VNI is used in both directions.

However, even if LEAF1 and LEAF3 both had VLAN 10 and VLAN 30 locally configured, the
L3VNI would still be used to route traffic between HostA and HostC.
```

Underlay and Overlay Concepts

VXLAN is an Overlay technology, and requires that an Underlay IP network exists prior to deployment. As previously discussed, EVPN provides the control-plane for the VXLAN overlay network. The primary functions and differences between the Underlay and Overlay are detailed below:

Underlay

- Primary purpose is to establish IP reachability between VTEP loopback addresses
- Leverages the Global Routing Table (VRF Default)
- No tenant (VRF) prefixes should ever exist within the underlay
- Arista recommends BGP peering in the IPv4 unicast address-family to establish reachability between VTEP loopbacks
- Note: An IGP such as IS-IS or OSPF can suffice as well, but is subject to scale limitations in large, densely connected topologies (such as spine/leaf) due to redundant flooding of LSPs/LSAs

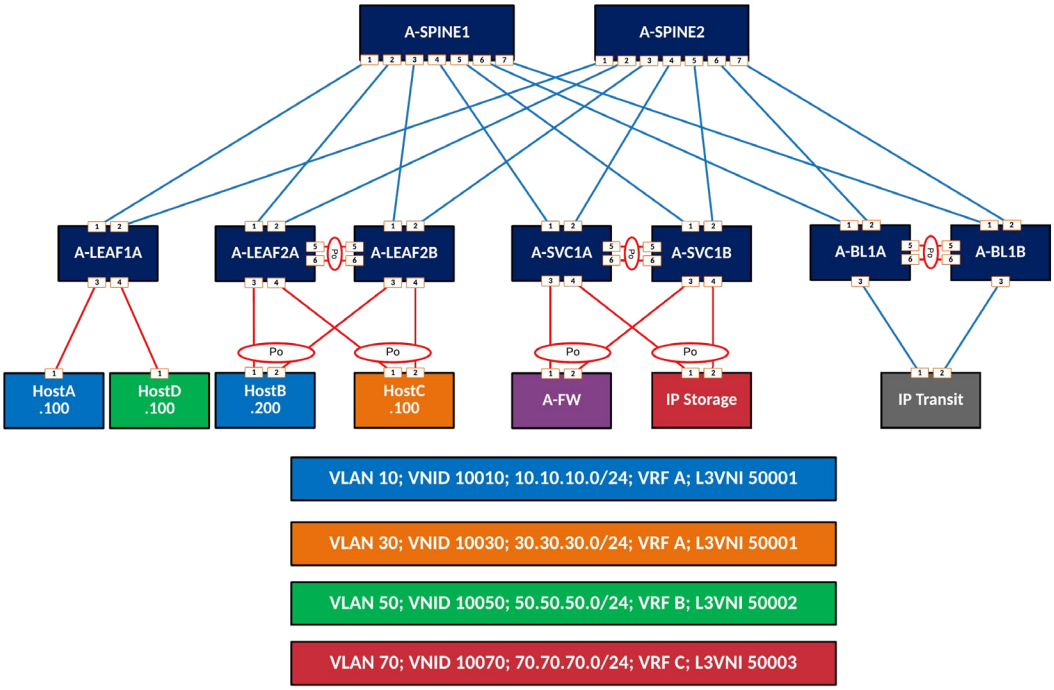
Overlay

- Primary purpose is to provide services to tenants (L2VPN, L3VPN) via VXLAN
- Leverages BGP Peering in the EVPN Address-Family
- All tenant reachability information exists in the overlay EVPN control-plane
- Underlay prefixes (such as Loopbacks) should not exist within the overlay control-plane
- The Underlay is the foundation for the Overlay
- For example, the BGP peerings in the EVPN Address-Family are established via loopback addresses, which exist in the underlay

Topology Overview

Throughout the remainder of this guide, all configuration commands and validation output will use the below Layer 3 Leaf and Spine reference topology:

Data Center A



In total, there are three tenants (VRFs). Tenants A and B are isolated from each other, while Tenant C is a shared services VRF that contains an IP storage array that is accessed via multiple tenants. Additionally, a Firewall exists to enforce security policy on inter-VRF communications.

The insertion of service devices and shared services, such as firewalls and IP storage arrays, will be covered in detail in a future release of this guide.

Underlay IP Addressing

The build-out of Data Center A will begin with addressing the IPv4 Unicast underlay that will serve as the transport between VTEPs. Each point-to-point connection between Spine and Leaf will be based upon the IP address schema listed in Appendix A (link to Appendix A here). While the point-to-point subnets configured within this guide are all /24 subnets, this is primarily for ease of readability. It is common for /31 subnets to be used for these connections. The approach of assigning /31 subnets on point-to-point connections is recommended if IP address conservation is of concern to the organization.

```
Note: When planning the underlay IP addressing schema for an EVPN deployment, it is
important to note that the only addresses that must be unique between EVPN domains (or
PODs) are the Loopback addresses of the VTEPs. The underlay point-to-point addresses
can be identical between PODs, as reachability to these prefixes is not required for
L2VPN or L3VPN services. L2VPN and L3VPN services data-plane traffic will always go
through the links within the EVPN domain IP underlay, but will never traverse these
links.
```

A-SPINE1 Interface Config Example:

```
interface Ethernet1
   description A-LEAF1A
   mtu 9214
   no switchport
   ip address 10.101.201.201/24
```

A-LEAF1A Interface Config Example:

```
interface Ethernet1
   description A-SPINE1
   mtu 9214
   no switchport
   ip address 10.101.201.101/24
```

In addition to configuring the IPv4 Underlay point-to-point interfaces, each device within the topology will have a unique IP address assigned to Loopback0. The IP address assigned to Loopback0 will used for BGP Peering within the EVPN Address-Family.

All Leaf switches will also have a Loopback1 interface defined, which will serve as the source for VXLAN tunnels. On Leaf switches in an MLAG domain, the IP address of Loopback1 will be identical between the MLAG peers. This will enable the MLAG domain Leaf switches to present themselves as a single logical VTEP. This will be covered in further detail in a later section.

A-SPINE1 Loopback Config Example:

```
interface Loopback0
   description EVPN Peering
   ip address 1.1.1.201/32
```

A-LEAF1A Loopback Config Example:

```
interface Loopback0
   description EVPN Peering
   ip address 1.1.1.101/32

interface Loopback1
   description VXLAN Tunnel Source
   ip address 2.2.2.1/32
```

A-LEAF2A Loopback Config Example:

```
interface Loopback0
   description EVPN Peering
   ip address 1.1.1.102/32

interface Loopback1
   description VXLAN Tunnel Source
   ip address 2.2.2.2/32
```

A-LEAF2B Loopback Config Example:

```
interface Loopback0
   description EVPN Peering
   ip address 1.1.1.103/32

interface Loopback1
   description VXLAN Tunnel Source
   ip address 2.2.2.2/32
```

Spine switches will not be serving as VTEPs, and because of this, they do not require a Loopback1 interface.

The primary responsibilities of the Spines will be:

- Providing fast and predictable forwarding throughout the fabric
- Act as eBGP Route Servers for the IPv4 Unicast and EVPN Address Families
- Forwarding of IP traffic, providing underlay transport between VTEPs

All MLAG pairs will have a dedicated VLAN interface for iBGP peering between each other in the IPv4 Unicast address-family. The details and purpose of this peering will be covered in the next section, but it is important to note that the IP addresses configured on this VLAN interface can be repeated across all MLAG pairs. The configuration of this interface is shown below:

A-LEAF2A Vlan4093 Config:

```
Vlan 4093
name MLAG-iBGP-Peering
!
interface Vlan4093
   description MLAG iBGP Peering
   ip address 192.0.0.1/24
```

A-LEAF2B Vlan4093 Config:

```
Vlan 4093
name MLAG-iBGP-Peering
!
interface Vlan4093
   description MLAG iBGP Peering
   ip address 192.0.0.2/24
```

A final note on the underlay IP addressing is that it does not require the use of VRFs. All underlay IP addressing should exist within the default routing table (also referred to as the *global* routing table). No tenant prefixes should exist within the underlay and no underlay prefixes should exist within a tenant's respective VRF. This will be further highlighted in a later section.

IP Underlay Configuration

Once all underlay IP addressing has been completed, reachability must be established between the Loopback0 address of all Spine and Leaf devices. This will be necessary to ensure BGP adjacencies can be formed within the EVPN address-family.

There are multiple options for establishing reachability within the IP underlay. The most common decision point is whether to implement BGP or an IGP to provide underlay reachability. Each method has its own unique benefits and flaws. A brief summary of each can be found below:

BGP

- Very high and proven scale (The Internet)
- No dynamic peering by default (requires additional configuration)
- Slow to converge by default (fast convergence requires additional configuration)
- Robust traffic engineering capabilities
- Single protocol for overlay and underlay along with IPv4/IPv6

IGP (OSPF or IS-IS)

- Scale limitations in large densely connected topologies
- Dynamic peering by default
- Default convergence faster than BGP
- Limited traffic engineering capabilities
- Requires BGP for overlay

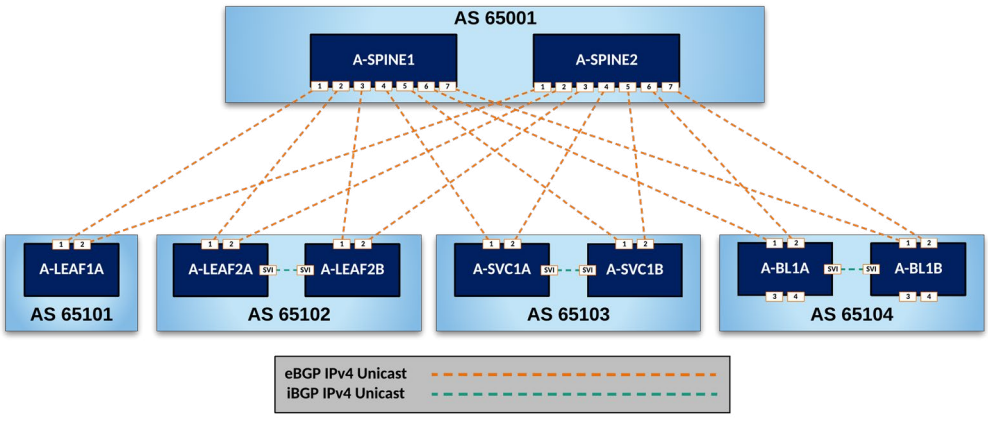
While the above is not an exhaustive list, it highlights some of the operational criteria that will impact the size and complexity of the configuration, as well as how standardized and repeatable the resulting configuration will be. The goal will be to make the resulting configuration of our IP underlay routing protocol as repeatable and dynamic as possible, therefore minimizing configuration variables without sacrificing scalability.

This guide will leverage eBGP as the protocol to establish reachability within the IP underlay. We will be implementing features within BGP to enable us to achieve very high scale without sacrificing the dynamic peering, fast convergence, and repeatable configuration of an IGP such as OSPF or IS-IS.

The first step in this guide will be to begin BGP configuration on the Spines. The Spines will serve as eBGP route-servers for the IPv4 Unicast address-family, peering with all Leaf switches, but not with each other. All of the Spine switches will reside in the same BGP Autonomous System Number (ASN). This will enable the repeatability of the BGP configuration on the Leaf switches and will reduce the amount of control-plane state in the BGP table. This reduction is done through the use of eBGP's default loop avoidance mechanism, which stipulates that a BGP update received that contains a device's own BGP ASN within the AS_PATH attribute will be rejected.

The end state of the eBGP IPv4 Unicast peering within the underlay of Data Center A is shown below:

eBGP IPv4 Unicast Underlay Peering



As seen above, while eBGP is used for peering between Spine and Leaf switches, each Leaf 'pod' is deployed as it's own respective ASN. All Leaf switches deployed in an MLAG pair will configure iBGP peering with each other. The iBGP peering ensures that if an MLAG peer loses it's northbound connectivity to the Spines, it can maintain reachability to remote VTEPs and the Spines through it's peer. It's important to note that this iBGP peering only exists within the IPv4 Unicast Address-Family, and not EVPN. iBGP peering in the EVPN address-family is not necessary between MLAG peers, as this would only introduce unnecessary control-plane state and configuration.

Enable Multi-Agent Routing Protocol Model

In order for EOS to support the use of the EVPN address-family, a multi-agent routing model must be used for BGP. The command below can be entered in global configuration mode on all Spine and Leaf switches to enable this:

```
service routing protocols model multi-agent
```

```
Note: A reboot of the switch is required after enabling the multi-agent routing model.
```

Once the multi-agent routing protocol model has been enabled, we can proceed with configuring BGP on the Spine and Leaf switches.

Spine BGP IPv4 Unicast Configuration

```
router bgp 65001
   router-id 1.1.1.201
   update wait-install
   no bgp default ipv4-unicast
   bgp listen range 10.0.0.0/8 peer-group IPv4-UNDERLAY-PEERS peer-filter LEAF-AS-RANGE
   neighbor IPv4-UNDERLAY-PEERS peer-group
   neighbor IPv4-UNDERLAY-PEERS password @rista123
   neighbor IPv4-UNDERLAY-PEERS send-community
   neighbor IPv4-UNDERLAY-PEERS maximum-routes 12000
   redistribute connected route-map RM-CONN-2-BGP
   !
   address-family ipv4
      neighbor IPv4-UNDERLAY-PEERS activate
```

Throughout this example, A-SPINE1 is used as the reference point for the configuration. Once complete, the configuration will be easily repeatable on all subsequent Spine switches in Data Center A, with only the Router-ID needing to be modified for each respective Spine.

In the next section, the above configuration will be broken down to explain each line and address any additional configuration that is necessary, such as prefix-lists, peer-filters and route-maps.

Enable BGP

```
router bgp 65001
```

This enables the BGP process and places the device into ASN 65001. A BGP instance can only be part of a single ASN.

Set the BGP Router-ID

```
router bgp 65001
  router-id 1.1.1.101
```

The BGP Router-ID should be unique within a BGP speaker's respective Autonomous System, and should not conflict with any of its BGP peers. The Router-ID is used in BGP Open messages exchanged during the establishment of a BGP peering session, and must not conflict between peers. While some BGP implementations allow this value to be automatically defined, best practice is to manually define the Router-ID. In this environment, the IP address of Loopback0 is used as our BGP Router-ID.

Optimize Convergence Event Behavior

```
router bgp 65001
   update wait-install
```

When a BGP convergence event occurs, the above command will enable the following behavior:

- Do not advertise reachability to a prefix until that prefix has been installed in hardware. This will eliminate any temporary black holes due to a BGP speaker advertising reachability to a prefix that may not yet be installed into the forwarding plane

Disable default peering in IPv4 Unicast address-family

```
router bgp 65001
  no bgp default ipv4-unicast
```

By default, the BGP process in EOS will automatically attempt to peer with all configured neighbors within the IPv4 Unicast address-family.

This behavior is disabled so that peerings within the IPv4 Unicast address-family must be explicitly configured. This ensures that when a neighbor is defined, peering does not immediately occur within the IPv4 address-family. Such immediate peering would result in unnecessary peerings and control-plane state.

Enable Dynamic BGP Peering

```
router bgp 65001
  bgp listen range 10.0.0.0/8 peer-group IPv4-UNDERLAY-PEERS peer-filter LEAF-AS-RANGE
```

Without the above command, BGP neighbors must be manually defined in order for peering to occur. Dynamic BGP peering overcomes this by indicating that peering will be established with any BGP speaker sourcing it's BGP session from an IP address within the defined range. In this case, any peer sourcing it's BGP session from 10.0.0.0/8 will be accepted.

Peers formed via this dynamic statement are placed into the "IPv4-UNDERLAY-PEERS" peer-group. This will allow a common policy to be applied to all dynamic peers.

Finally, eBGP will be used between the Spine and Leaf switches, and the remote ASNs that will be accepted for peering must also be specified. This, in combination with the "listen" range of 10.0.0.0/8 and authentication via the peer-group, will safeguard that the Spines are only peering with trusted and known BGP speakers.

The peer-filter below defines the range of acceptable remote ASNs:

```
peer-filter LEAF-AS-RANGE
   10 match as-range 65001-65199 result accept
```

Define the peer-group for IPv4 Unicast peering

```
router bgp 65001
  neighbor IPv4-UNDERLAY-PEERS peer-group
```

Creating a peer-group applies a common policy set across all peers within the peer-group.

Policies that will be set include BFD (fast convergence), authentication (security), and maximum-routes (RIB/FIB protection mechanism).

Protect against rogue BGP speakers (Authentication)

```
router bgp 65001
  neighbor IPv4-UNDERLAY-PEERS password @rista123
```

Since BGP dynamic peering is being used, it is important that peering only occurs with trusted BGP speakers sourcing their BGP session from within the 10.0.0.0/8 address-space.

Requiring a password for the peering to be established protects the BGP speakers from unexpected/unintended peerings.

Protect the RIB/FIB from unexpected large updates

```
router bgp 65001
  neighbor IPv4-UNDERLAY-PEERS maximum-routes 12000
```

This is the default configuration within BGP on EOS. If more than 12,000 prefixes are learned from a peer with this peer-group applied, the BGP adjacency will be disabled. If desired, this can be disabled by setting the maximum routes to a value of '0'.

Additionally, this can be set to only generate a warning by specifying an action of 'warning-only' after the defined prefix quantity.

Advertise Point-to-Point and Loopback interface prefixes

```
router bgp 65001
  redistribute connected route-map RM-CONN-2-BGP
```

In order to provide reachability to all locally connected prefixes (including the prefixes associated with Loopback0 and Loopback1), this command will bring all directly connected prefixes into BGP via a redistribution statement.

This will only redistribute the prefixes that match the route-map "RM-CONN-2-BGP". This route-map references prefix-lists to prevent any unexpected or unintended prefixes from being redistributed into BGP.

Listed below for reference is the configuration of the route-map, as well as the prefix-lists referenced by the route-map:

```
route-map RM-CONN-2-BGP permit 10
   match ip address prefix-list PL-LOOPBACKS
!
route-map RM-CONN-2-BGP permit 20
   match ip address prefix-list PL-P2P-UNDERLAY
```

The route-map configuration above dictates that only prefixes matched by the referenced prefix-lists are permitted to be brought into the BGP RIB.

```
ip prefix-list PL-LOOPBACKS seq 10 permit 1.1.1.0/24 eq 32
ip prefix-list PL-LOOPBACKS seq 20 permit 2.2.2.0/24 eq 32
ip prefix-list PL-P2P-UNDERLAY seq 10 permit 10.0.0.0/8 le 31
```

The prefix-list 'PL-LOOPBACKS' will only match on /32 prefixes that exist within the 1.1.1.0/24 and 2.2.2.0/24 ranges.

Likewise, the prefix-list PL-P2P-UNDERLAY will match all prefixes within the 10.0.0.0/8 range, but can only have a prefix length of up to /31.

As seen on the Leaf switches in a later section, these prefix-lists can be consistent throughout Data Center A.

```
Note: When reachability to the IP underlay point-to-point links within the EVPN
domain is not required, and/or when leveraging identical underlay IP addressing
schemas between different EVPN domains, the 'ip prefix-list PL-P2P-UNDERLAY seq 10
permit 10.0.0.0/8 le 31' configuration is not required. In this case you might need to
appropriately source the traffic off the loopback IP when issuing pings or traceroute
from the switch for troubleshooting purposes.

A scenario in which the underlay P2P links would not be advertised is when the same
IP Schema is used across multiple Data Centers. In such a  scenario, the underlay P2P
link addressing does not need to be known beyond the directly connected peer. This
approach is commonly used to remove the variable of unique IP address schemas between
Data Centers when scripting deployments. The only value that must be unique across
data centers is the Loopback IP address of each respective VTEP (or MLAG pair of
VTEPs).
```

Activate IPv4 Unicast BGP Peering

```
router bgp 65001
  address-family ipv4
      neighbor IPv4-UNDERLAY-PEERS activate
```

Finally, IPv4 Unicast address-family peering will be enabled with all dynamic peers formed as a part of the IPv4-UNDERLAY-PEERS peer-group.

Before the adjacencies are formed, the Leaf switches will need to be configured with the appropriate BGP configuration. That configuration will be the focus of the next section.

Leaf BGP IPv4 Unicast Configuration

```
router bgp 65102
   router-id 1.1.1.102
   update wait-install
   no bgp default ipv4-unicast
```

```
   maximum-paths 2
   neighbor IPv4-UNDERLAY-PEERS peer-group
   neighbor IPv4-UNDERLAY-PEERS remote-as 65001
   neighbor IPv4-UNDERLAY-PEERS password @rista123
   neighbor IPv4-UNDERLAY-PEERS send-community
   neighbor IPv4-UNDERLAY-PEERS maximum-routes 12000
   neighbor MLAG-IPv4-UNDERLAY-PEER peer-group
   neighbor MLAG-IPv4-UNDERLAY-PEER remote-as 65102
   neighbor MLAG-IPv4-UNDERLAY-PEER next-hop-self
   neighbor MLAG-IPv4-UNDERLAY-PEER password @rista123
   neighbor MLAG-IPv4-UNDERLAY-PEER send-community
   neighbor MLAG-IPv4-UNDERLAY-PEER maximum-routes 12000
   neighbor 192.0.0.2 peer-group MLAG-IPv4-UNDERLAY-PEER
   neighbor 10.102.201.201 peer-group IPv4-UNDERLAY-PEERS
   neighbor 10.102.202.202 peer-group IPv4-UNDERLAY-PEERS
   redistribute connected route-map RM-CONN-2-BGP
   !
   address-family ipv4
      neighbor IPv4-UNDERLAY-PEERS activate
      neighbor MLAG-IPv4-UNDERLAY-PEER activate
```

Throughout this section, we will use A-LEAF2A as our reference point for the configuration. We do so because it contains all BGP configuration necessary for a single-homed leaf, as well as a Leaf in an MLAG pair.

As seen above, much of the configuration is identical to that which was applied to A-SPINE1 in the previous section. Because of this, a detailed overview for any configuration that was covered in the previous section will not be duplicated here. The focus in this section will be on what is unique to the Leaf switch IPv4 BGP configuration.

Enable Equal Cost Multipathing

```
router bgp 65102
   maximum-paths 2
```

By default, BGP will choose a single best path to reach a particular prefix. Multi-pathing can be enabled for a prefix in BGP under the following conditions:

- The same route is received from multiple BGP peers, each with a unique next-hop
- All BGP attributes associated with the received routes, up to the smallest IGP metric to the next-hop, are equal
- Multipathing has been enabled (maximum-paths)

In this example, a maximum paths of 2 has been defined. This number should match the number of Spines deployed in the topology.

Applying Common IPv4 Unicast Peering Policy

```
router bgp 65102
  neighbor IPv4-UNDERLAY-PEERS remote-as 65001
```

Since dynamic BGP peering has been enabled on the Spines, dynamic peering configuration cannot be utilized on the Leaf switches. As such, the BGP ASN that the Spine switches will be peering from must be defined. Additionally, peering with the Spine switches will need to be manually defined within the BGP configuration, as shown below:

```
router bgp 65102
  neighbor 10.102.201.201 peer-group IPv4-UNDERLAY-PEERS
  neighbor 10.102.202.202 peer-group IPv4-UNDERLAY-PEERS
```

This configuration allows Spine switches to be added to/removed from the environment, without substantial configuration modifications to the Leaf switches. In this example, the only unique variables within the Leaf switch BGP configuration are:

- The second octet of the IP addresses used to peer with the Spines
- The BGP Router-ID for the Leaf switch

iBGP Peering with MLAG Peer

```
router bgp 65102
  neighbor 192.0.0.2 peer-group MLAG-IPv4-UNDERLAY-PEER
```

Since A-LEAF2A is a part of an MLAG domain with its peer A-LEAF2B, it is necessary to ensure that A-LEAF2A is able to maintain reachability to remote VTEPs and the Spines, even if it's connectivity to the Spines is lost. In order to do so, iBGP peering will be established with A-LEAF2B.

The MLAG BGP peer is placed in its own unique peer-group. This peer-group can be re-used on all MLAG domain BGP peerings, as the policy will be consistent across all MLAG domains. This peer-group contains a few unique policies that will be applied to the iBGP session between the MLAG peers, and differs from the IPv4-UNDERLAY-PEERS peer-group in the following ways:

- The peer's remote ASN will match the local ASN, making it an iBGP peering
- The 'next-hop-self' policy will be applied.
    - This will dictate that any prefix advertised to the MLAG peer via this iBGP peering session will have the next-hop attribute modified to whatever is defined as the local 'update-source', which in this case will be the directly connected interface used to reach the peer. This is done to avoid scenarios of 'next-hop inaccessible' occurring for prefixes sent to the MLAG peer, which would prevent the prefix being marked as a 'best' path and entered into the IPv4 RIB.

Note that a dedicated SVI (Vlan4093) is being used to establish the iBGP underlay peering between the MLAG peers. While it is technically possible to leverage the existing MLAG Peer-Sync SVI of 4094, using a dedicated SVI is recommended for operational and troubleshooting purposes.

The peering between MLAG peers is repeatable, as the prefix associated with the MLAG peering is not advertised through any routing process, making it link-local in nature. Accordingly, all MLAG domains within the environment will use VLAN 4093 for IPv4 BGP peering, and 192.0.0.0/24 as the prefix associated with this VLAN.

MLAG peer "A" will always have an IP address of 192.0.0.1

MLAG peer "B" will always have an IP address of 192.0.0.2

This minimizes configuration variables within the environment, particularly when dealing with MLAG pairs. For a full example of the MLAG configuration, please refer to Appendix X: Full Device Configurations (link to Appendix X here)

Underlay Validation

Once configuration is complete, it is time to validate that the configuration has had the intended effect. At this point, all Spine and Leaf switches in Data Center A should have full IP reachability between each other. This includes pings destined to, or sourced from, either point-to-point or Loopback0 interfaces. The command output will be from the perspective of A-SPINE1, but the commands are relevant to all of our Spine and Leaf switches.

Below are the commands that can be used to validate our configuration:

iBGP Peering with MLAG Peer

```
A-SPINE1(config-router-bgp)#show ip bgp summary
BGP summary information for VRF default
Router identifier 1.1.1.201, local AS number 65001
Neighbor Status Codes: m - Under maintenance
Neighbor         V  AS      MsgRcvd   MsgSent  InQ OutQ  Up/Down State  PfxRcd PfxAcc
10.101.201.101   4  65101       401       400    0    0 05:40:55 Estab       4       4
10.102.201.102   4  65102        77        64    0    0 00:53:15 Estab       6       6
10.103.201.103   4  65102        77        66    0    0 00:53:18 Estab       6       6
10.104.201.104   4  65103       420       407    0    0 05:40:33 Estab       7       7
10.105.201.105   4  65103       419       403    0    0 05:40:48 Estab       7       7
10.106.201.106   4  65104       416       403    0    0 05:40:30 Estab       7       7
10.107.201.107   4  65104       417       403    0    0 05:40:16 Estab       7       7
```

The 'show ip bgp summary' command output shows all BGP peers that are currently in an 'Established' state. We should see all Leaf switches within Data Center A listed in this output. Peerings should be formed using directly connected interfaces from the point-to-point connections between Leaf and Spine switches.

iBGP Peering with MLAG Peer

```
A-SPINE1#show ip bgp neighbors 10.102.201.102
BGP neighbor is 10.102.201.102, remote AS 65102, external link
  BGP version 4, remote router ID 1.1.1.102, VRF default
  Inherits configuration from and member of peer-group IPv4-UNDERLAY-PEERS
<...Output Omitted...>
  Neighbor Capabilities:
    Multiprotocol IPv4 Unicast: advertised and received and negotiated
<...Output Omitted...>
  Configured maximum total number of routes is 12000
<...Output Omitted...>
Local AS is 65001, local router ID 1.1.1.201
<...Output Omitted...>
MD5 authentication is enabled
```

This show command output validates, on a per neighbor basis, that peering was established within the proper address-family, and that policy has been applied via the IPv4-UNDERLAY-PEERS peer-group.

Items to note in this output include session state (Established), BFD, MD5 Authentication, Peer-Group, and Maximum-Routes value.

Prefixes Exist in BGP RIB:

```
A-SPINE1#show ip bgp
<...Output Omitted...>
 * >      1.1.1.101/32          10.101.201.101        -       100     0       65101 i
 * >      1.1.1.102/32          10.102.201.102        -       100     0       65102 i
 *        1.1.1.102/32          10.103.201.103        -       100     0       65102 i
 * >      1.1.1.103/32          10.102.201.102        -       100     0       65102 i
 *        1.1.1.103/32          10.103.201.103        -       100     0       65102 i
 * >      1.1.1.104/32          10.104.201.104        -       100     0       65103 i
 *        1.1.1.104/32          10.105.201.105        -       100     0       65103 i
 * >      1.1.1.105/32          10.104.201.104        -       100     0       65103 i
 *        1.1.1.105/32          10.105.201.105        -       100     0       65103 i
 * >      1.1.1.106/32          10.106.201.106        -       100     0       65104 i
 *        1.1.1.106/32          10.107.201.107        -       100     0       65104 i
 * >      1.1.1.107/32          10.106.201.106        -       100     0       65104 i
 *        1.1.1.107/32          10.107.201.107        -       100     0       65104 i
 * >      1.1.1.201/32          -                     -       -       0       i
 * >      2.2.2.1/32            10.101.201.101        -       100     0       65101 i
 * >      2.2.2.2/32            10.102.201.102        -       100     0       65102 i
 *        2.2.2.2/32            10.103.201.103        -       100     0       65102 i
 * >      2.2.2.3/32            10.104.201.104        -       100     0       65103 i
 *        2.2.2.3/32            10.105.201.105        -       100     0       65103 i
 * >      2.2.2.4/32            10.106.201.106        -       100     0       65104 i
 *        2.2.2.4/32            10.107.201.107        -       100     0       65104 i
 <...Output Omitted...>
```

Once peerings have been established within the IPv4 Unicast address-family, A-SPINE1 should be receiving prefixes from all Leaf switches. These prefixes should represent the point-to-point Leaf and Spine links within Data Center A, as well as all Loopback0 and Loopback1 addresses. Each prefix should have a 'Best' path selected, which is annotated by a ">" symbol.

Likewise, the output of 'show ip bgp' on the Leaf switches should show prefixes learned via the Spines.

iBGP Peering with MLAG Peer

```
A-SPINE1#show ip route bgp
<...Output Omitted...>
 B E      1.1.1.101/32 [200/0] via 10.101.201.101, Ethernet1
 B E      1.1.1.102/32 [200/0] via 10.102.201.102, Ethernet2
 B E      1.1.1.103/32 [200/0] via 10.102.201.102, Ethernet2
 B E      1.1.1.104/32 [200/0] via 10.104.201.104, Ethernet4
 B E      1.1.1.105/32 [200/0] via 10.104.201.104, Ethernet4
 B E      1.1.1.106/32 [200/0] via 10.106.201.106, Ethernet6
 B E      1.1.1.107/32 [200/0] via 10.106.201.106, Ethernet6
 B E      2.2.2.1/32 [200/0] via 10.101.201.101, Ethernet1
 B E      2.2.2.2/32 [200/0] via 10.102.201.102, Ethernet2
 B E      2.2.2.3/32 [200/0] via 10.104.201.104, Ethernet4
 B E      2.2.2.99/32 [200/0] via 10.106.201.106, Ethernet6
 <...Output Omitted...>
```

All prefixes that were annotated with a ">" in the output of 'show ip bgp' are candidates to be entered into the IPv4 RIB. We should see these prefixes in the output of 'show ip route' on our switches.

IP Reachability Exists between all Loopbacks:

```
A-SPINE1#ping 1.1.1.101 source Lo0
PING 1.1.1.101 (1.1.1.101) from 1.1.1.201 : 72(100) bytes of data.
80 bytes from 1.1.1.101: icmp_seq=1 ttl=64 time=0.201 ms
80 bytes from 1.1.1.101: icmp_seq=2 ttl=64 time=0.100 ms
80 bytes from 1.1.1.101: icmp_seq=3 ttl=64 time=0.098 ms
80 bytes from 1.1.1.101: icmp_seq=4 ttl=64 time=0.098 ms
80 bytes from 1.1.1.101: icmp_seq=5 ttl=64 time=0.100 ms
```

```
A-SPINE1#ping 2.2.2.1 source Lo0
PING 2.2.2.1 (2.2.2.1) from 1.1.1.201 : 72(100) bytes of data.
80 bytes from 2.2.2.1: icmp_seq=1 ttl=64 time=0.073 ms
80 bytes from 2.2.2.1: icmp_seq=2 ttl=64 time=0.031 ms
80 bytes from 2.2.2.1: icmp_seq=3 ttl=64 time=0.030 ms
80 bytes from 2.2.2.1: icmp_seq=4 ttl=64 time=0.030 ms
80 bytes from 2.2.2.1: icmp_seq=5 ttl=64 time=0.031 ms
```

Finally, if all Leaf and Spine switches can ping each other, then underlay IP reachability exists and we can begin the buildout of the overlay control-plane. Leaf and Spine switches should be able to ping each other, regardless of from where the ping is sourced. In the above example, A-SPINE1 is pinging A-LEAF1A's Loopback0 and Loopback1 addresses, sourced from it's own Loopback0 address.

EVPN Overlay Control-Plane

Now that IP reachability has been established between all Spine and Leaf switches within the Data Center, it is time to focus on building the eBGP adjacencies within the EVPN address-family. This address-family will be used to provide the control-plane for our VXLAN data-plane based Layer2 and Layer3 VPN services.

The EVPN Route-Types (to be explained in further detail later in this guide) that will be utilized for L2VPN and L3VPN service provisioning are:

Route-Type	Name	Purpose
2	MAC-IP	Layer 2 VPN: End-Host Information (MAC, IP, etc.)
3	IMET (Inclusive Multicast Ethernet Tag)	Signal desire to receive BUM traffic for a VNI
5	IP Prefix	Layer 3 VPN: IP Prefix and VRF Membership

The overlay will be using eBGP instead of iBGP, considering that the BGP ASN structure is already in place for these peerings.

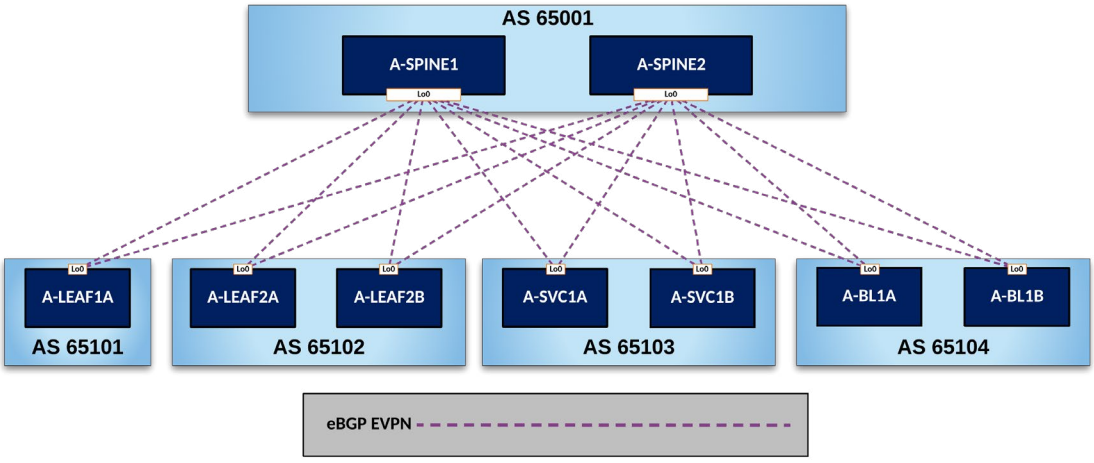
iBGP is a common choice when leveraging an IGP such as IS-IS or OSPF to provide IP reachability within the underlay. In an iBGP deployment, the Spines act as Route-Reflectors for the EVPN address-family.

It is worth noting that at this point, VXLAN interfaces haven't been created, nor have any Layer2 or Layer3 VPN services been defined. The EVPN control-plane is laying the foundation that these services will be provided upon.

Peerings within the EVPN address-family will leverage an update-source of Loopback0. Consequently, all sessions will be established to/from Loopback0 IP addresses. However, because Loopback1 will be specified as the source interface for VXLAN, any locally originated routes within the EVPN address-family will have a next-hop value of the originating device's Loopback1 IP address.

While this will utilize underlay IP addressing for the EVPN address-family peerings, underlay prefixes will not exist within the EVPN overlay control-plane.

Upon completion of the EVPN peerings, the following peering structure will be in place:



Spine BGP EVPN Peering Configuration

```
router bgp 65001
   bgp listen range 1.1.1.0/24 peer-group EVPN-OVERLAY-PEERS peer-filter LEAF-AS-RANGE
   neighbor EVPN-OVERLAY-PEERS peer-group
   neighbor EVPN-OVERLAY-PEERS next-hop-unchanged
   neighbor EVPN-OVERLAY-PEERS update-source Loopback0
   neighbor EVPN-OVERLAY-PEERS fall-over bfd
   neighbor EVPN-OVERLAY-PEERS ebgp-multihop 3
   neighbor EVPN-OVERLAY-PEERS password @rista123
   neighbor EVPN-OVERLAY-PEERS send-community
   neighbor EVPN-OVERLAY-PEERS maximum-routes 0
   !
   address-family evpn
      neighbor EVPN-OVERLAY-PEERS activate
```

The above configuration, using A-SPINE1 as the reference device, omits all configuration related to the IPv4 Unicast address-family. The IPv4 Unicast configuration was covered in detail in the previous section. Additionally, any configuration covered in detail in the previous section will not have a duplicate explanation here.

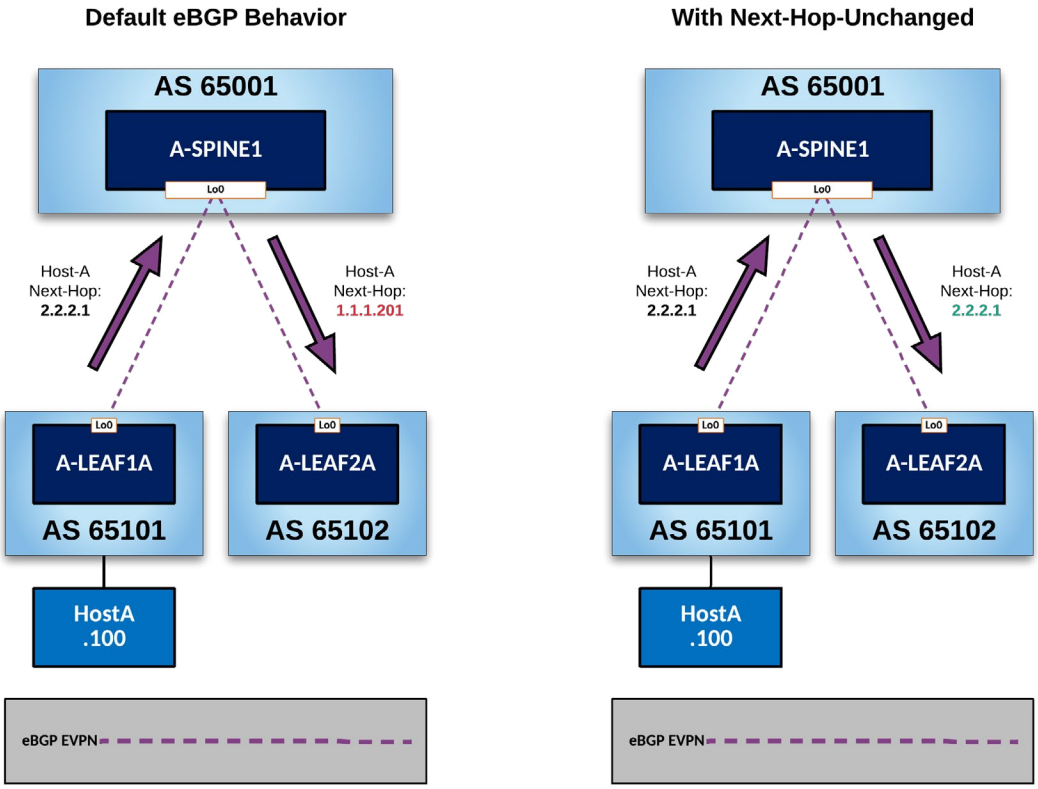
```
router bgp 65001
  bgp listen range 1.1.1.0/24 peer-group EVPN-OVERLAY-PEERS peer-filter LEAF-AS-RANGE
```

Just as with the IPv4 Unicast address-family, dynamic peering within the EVPN address-family will be configured. Within the EVPN address-family, adjacencies will be established with BGP speakers sourcing their BGP session from within the 1.0.0.0/24 address range. This maps to the Loopback0 IP addresses within Data Center A.

Since the policy applied to peerings within the EVPN address-family will differ from those applied to IPv4 Unicast, it is necessary to define a separate peer-group of "EVPN-OVERLAY-PEERS". The policy configuration applied via this peer-group will be explored further in this section.

```
router bgp 65001
  neighbor EVPN-OVERLAY-PEERS next-hop-unchanged
```

The default behavior of eBGP is that any prefix learned from an eBGP peer, and then advertised to another eBGP peer, will have the value of the NEXT_HOP attribute modified to whatever interface is used as the source for the BGP peering session towards the receiving peer.



As can be seen above, we do not want the result of the default eBGP behavior (the Spine setting the next-hop attribute of the update to its own Loopback0 address) as the update is sent to A-LEAF2A. The primary reason for this is that our Spines are not VTEPs. The value of the next-hop attribute will be the IP address that a remote VTEP uses as the destination IP address in the outer-IP header of a VXLAN packet. If this is set to the Spine's Loopback0 address, the VXLAN tunnel will not be created, and reachability to HostA will not exist from remote VTEPs.

```
router bgp 65001
  neighbor EVPN-OVERLAY-PEERS update-source Loopback0
```

All peerings in the EVPN address-family will be sourced from the IP address associated with Loopback0.

```
router bgp 65001
  neighbor EVPN-OVERLAY-PEERS ebgp-multihop 3
```

In order to allow for peering between Loopback0 IP addresses, the TTL field in the IP header of eBGP control-plane packets will be set to 3. By default this is set to a value of 1 for eBGP sessions. If left at the default value, the TTL would expire before reaching the Loopback address of the remote peer.

This value is consistently set to 3 on all Leaf and Spine switches. Spine and Leaf switches in steady state would only require a TTL of 2. However, if an MLAG peer lost it's connectivity to the Spines, it would need to maintain it's BGP EVPN adjacencies with the Spines by sending it's BGP Keepalives through it's MLAG peer (due to the iBGP IPv4 peering between the MLAG peers). This would result in an additional decrement of the TTL, hence the need for 3 to cover this failure scenario.

```
router bgp 65001
  neighbor EVPN-OVERLAY-PEERS send-community
```

As we will see in later sections where we instantiate Layer2 and Layer3 VPN services, the EVPN control-plane makes extensive use of Extended Communities in BGP. This is done for signaling VNIs, Route-Targets, EvpnRouterMac, etc. between VTEPs.

To enable this, all BGP speakers peering within the EVPN address-family must be configured to support extended communities.

```
Note: The 'send-community' neighbor configuration enables the sending of both standard
and extended communities by default
```

```
router bgp 65001
  address-family evpn
      neighbor EVPN-OVERLAY-PEERS activate
```

Peering must be enabled explicitly within the EVPN address-family for all peers that are a part of the EVPN-OVERLAY-PEERS peer-group.

```
Note: Recall that default peering within the IPv4 Unicast address-family was globally
disabled in the previous section, via the 'no bgp default ipv4-unicast' command under
BGP. Because of this, there is no need to explicitly disable peering within the IPv4
Unicast address-family for the EVPN-OVERLAY-PEERS peer-group. However, if peering
within the IPv4 Unicast address-family was not globally disabled, then it would need
to be explicitly disabled for the EVPN-OVERLAY-PEERS peer-group by using the 'no
neighbor EVPN-OVERLAY-PEERS activate' command under the address-family ipv4 section of
BGP.
```

Leaf BGP EVPN Peering Configuration

```
router bgp 65102
   neighbor EVPN-OVERLAY-PEERS peer-group
   neighbor EVPN-OVERLAY-PEERS remote-as 65001
   neighbor EVPN-OVERLAY-PEERS update-source Loopback0
   neighbor EVPN-OVERLAY-PEERS fall-over bfd
   neighbor EVPN-OVERLAY-PEERS ebgp-multihop 3
   neighbor EVPN-OVERLAY-PEERS password @rista123
   neighbor EVPN-OVERLAY-PEERS send-community
   neighbor EVPN-OVERLAY-PEERS maximum-routes 0
   neighbor 1.1.1.201 peer-group EVPN-OVERLAY-PEERS
   neighbor 1.1.1.202 peer-group EVPN-OVERLAY-PEERS
   !
   address-family evpn
      neighbor EVPN-OVERLAY-PEERS activate
```

Throughout this section, just as in the previous IPv4 Unicast configuration overview, A-LEAF2A will be used as the reference point for the configuration. Notice there is no need for peering in the EVPN address-family between MLAG peers, since this would just result in unnecessary control-plane state.

As seen above, much of the configuration is identical to that which was applied to A-SPINE1 in the previous section. For a detailed overview of that configuration, please refer to the previous section. This section will focus on what is unique to the Leaf switch EVPN BGP configuration.

```
router bgp 65102
  neighbor EVPN-OVERLAY-PEERS remote-as 65001
```

Since dynamic peers will not be used on the Leaf switches, the remote BGP ASN must be specified within the EVPN-OVERLAY-PEERS peer-group. Dynamic peering cannot be used on the Leaf switches as this is in place on the Spines. At least one end of the BGP peering session must be initiated by a BGP speaker. Considering that the Spines are only listening for incoming BGP sessions, the Leaf switches will initiate these sessions.

```
router bgp 65102
   neighbor EVPN-OVERLAY-PEERS fall-over bfd
```

The use of Bidirectional Forwarding Detection (BFD) will enable fast detection of a transport failure between EVPN peers. Instead of relying on the BGP keepalive timer values, BFD will notify the BGP process when a transport failure occurs, triggering subsequent convergence actions.

The default intervals for multi-hop BFD sessions in EOS are as follows:

Transmission Rate: 300ms

Minimum Receive Rate: 300ms

Multiplier: 3

While the above defaults will enable subsecond transport failure detection, these intervals can be tuned by the operator as needed by using the 'bfd multihop interval' command in global configuration mode.

```
router bgp 65102
   neighbor 1.1.1.201 peer-group EVPN-OVERLAY-PEERS
   neighbor 1.1.1.202 peer-group EVPN-OVERLAY-PEERS
```

EVPN peering will be established with the Loopback0 address of each respective Spine switch in Data Center A. Each Spine switch will be defined as a peer, and placed into the EVPN-OVERLAY-PEERS peer-group. Any BGP peer placed into this peer-group will implicitly be activated within the EVPN address-family.

Notice that all BGP EVPN peering configuration on the Leaf switches can be repeated across all Leaf switches, as there are no unique variables that are specific to any one Leaf.

Overlay EVPN Peering Validation

Once all the necessary configuration to facilitate peering within the EVPN address-family is complete, the control-plane is active and ready for L2VPN and L3VPN service provisioning. This section will validate the EVPN control-plane.

```
A-SPINE1#show bgp evpn summary
BGP summary information for VRF default
Router identifier 1.1.1.201, local AS number 65001
Neighbor Status Codes: m - Under maintenance
Neighbor    V  AS           MsgRcvd   MsgSent  InQ OutQ  Up/Down State   PfxRcd PfxAcc
1.1.1.101   4  65101            120       107    0    0 01:23:16 Estab   0      0
1.1.1.102   4  65102            115       118    0    0 01:23:16 Estab   0      0
1.1.1.103   4  65102            112       117    0    0 01:23:16 Estab   0      0
1.1.1.104   4  65103            117       120    0    0 01:23:16 Estab   0      0
1.1.1.105   4  65103            116       122    0    0 01:23:16 Estab   0      0
1.1.1.106   4  65104            122       120    0    0 01:23:16 Estab   0      0
1.1.1.107   4  65104            124       122    0    0 01:23:16 Estab   0      0
```

Notice that all peerings are in the "Established" state, and all are employing the Loopback0 IP address of each respective peer.

Additionally, do not be alarmed at this point when the "Prefixes Received" and "Prefixes Accepted" columns report a zero value. This is expected, as the L2VPN and L3VPN services have not yet been provisioned.

Validate BGP EVPN Neighbor Status and Policies:

```
A-SPINE1#show bgp neighbor 1.1.1.102
BGP neighbor is 1.1.1.102, remote AS 65102, external link
<...Output Omitted...>
  Inherits configuration from and member of peer-group EVPN-OVERLAY-PEERS
<...Output Omitted...>
  BGP state is Established, up for 01:33:55
<...Output Omitted...>
  Neighbor Capabilities:
    Multiprotocol L2VPN EVPN: advertised and received and negotiated
<...Output Omitted...>
  Configured maximum total number of routes is 0
<...Output Omitted...>
Local AS is 65001, local router ID 1.1.1.201
TTL is 3, external peer can be 3 hops away
<...Output Omitted...>
L2VPN EVPN missing local-nexthop, cannot send nexthop-self updates
Bfd is enabled and state is Up
MD5 authentication is enabled
```

This validates, on a per neighbor basis, that peering was established within the proper address-family, and that policy has been applied via our EVPN-OVERLAY-PEERS peer-group.

Items to pay close attention to in this output include:

- Session state is 'Established'
- TTL of 3 is used on eBGP control-plane traffic
- Next-hop-unchanged is in place
- MD5 Authentication in place
- Peer-Group membership
- Maximum-Routes value

With all BGP IPv4 Unicast and EVPN peerings in place, Layer2 and Layer3 VPN services can be provisioned to the tenants. This will be the focus of the next section.

Tenant Layer2 VPN Configuration

From this point forward, all configuration will be performed on the Leaf switches. Leaf switches can also be referred to as VTEPs, and for the remainder of this guide the terms will be used interchangeably.

The key reasons the Spines will not require any further configuration are:

- Spines are not VTEPs
- Spines do not have any locally configured VLANs
- Spines do not have any locally configured VRFs
- No end hosts or tenant workloads are connected to the Spines

Keeping the Spine configurations lean and their forwarding responsibilities to a minimum, ensures that maintenance performed on the Spines is as seamless as possible. It also conserves hardware resources as the TCAM is not consumed by Tenant MAC addresses or IP Prefixes.

Set TCAM profile to Vxlan-Routing (R-Series Platforms Only)

```
hardware tcam profile vxlan-routing
```

If the device being configured is a part of the "R" series of Arista platforms, the above command will be required, in global configuration, to enable VXLAN routing in hardware. This command will result in a brief data-plane interruption. Ensure it is entered when the device has been placed into maintenance mode, or during a window where a brief data-plane interruption is acceptable.

For details on placing a device into maintenance mode, please refer to the following resources:

- https://eos.arista.com/eos-4-15-2f/maintenance-mode/
- https://eos.arista.com/maintenance-mode-lab-example-of-bgp-on-spine/

An additional platform specific configuration, required for Trident2 and some Tomahawk ASIC based platforms, is the recirculation of packets in order to perform VXLAN routing in hardware. Please refer to the following resources for details on recirculation configuration:

- https://eos.arista.com/eos-4-15-2f/recirculation-channel/
- https://eos.arista.com/eos-4-15-2f/unconnected-ethernet/

L2VPN Configuration

As discussed earlier in this guide, VXLAN data-plane encapsulation will be used to provide Layer2 and Layer3 VPN services for the tenants. The focus of this section will be on providing Layer2 VPN services.

```
vlan 10
   name Ten
!
vlan 50
   name Fifty
!
interface Vxlan1
   vxlan source-interface Loopback1
   vxlan udp-port 4789
   vxlan vlan 10-1000 vni 10010-11000
!
```

```
router bgp 65101
  vlan-aware-bundle TENANT-A
      rd 1.1.1.101:1
      route-target both 1:1
      redistribute learned
      vlan 10-49
   !
   vlan-aware-bundle TENANT-B
      rd 1.1.1.101:2
      route-target both 2:2
      redistribute learned
      vlan 50-69
```

In the above example, A-LEAF1A is used as the reference point for the configuration. This configuration will be explained further in the next section, with each configuration component covered in detail. Additionally, any MLAG specific configuration will be focused upon, with A-LEAF2A and A-LEAF2B used as reference points for MLAG specific configuration.

```
Note: The above configuration shows the VLAN Aware Bundle MAC-VRF configuration
methodology. The alternative to this approach, VLAN-Based MAC-VRF, is detailed later
in this section, as will the differences and interoperability considerations between
the two different approaches.
```

Create VLANs

```
vlan 10
   name Ten
!
vlan 50
   name Fifty
```

Before configuring the components necessary to provide L2VPN services via VXLAN and EVPN, the local L2 forwarding constructs must be in place for tenant connectivity.

A-LEAF1A has endpoints from both Tenant-A (VLAN 10) and Tenant-B (VLAN 50) connected to it. The above configuration provisions the broadcast domains that will later be mapped to VXLAN VNIs to provide L2VPN services.

Enable VTEP Functionality

Standalone VTEP

```
interface Vxlan1
   vxlan source-interface Loopback1
   vxlan udp-port 4789
```

In this example, the definition of Loopback1 as the source-interface for VXLAN serves two purposes:

- The IP address that will be used in the Source IP field of the Outer-IP header of the VXLAN packet
- The IP address to be listed as the "next-hop" in the Network Layer Reachability Information (NLRI) for EVPN address-family BGP pdate messages sourced from this VTEP

The definition of the UDP port 4789 is a default configuration value and does not need to be manually configured. UDP port 4789 is the port defined within the RFC for the VXLAN protocol (https://tools.ietf.org/html/rfc7348). This default configuration shows up in the running config primarily for reference, but can be modified if needed.

MLAG VTEP

```
interface Vxlan1
   vxlan source-interface Loopback1
   vxlan udp-port 4789
```

The Leaf switches deployed as an MLAG pair must present themselves as a single logical VTEP. This is accomplished by defining an additional Loopback interface on each MLAG peer (in this case Loopback1), where both peers assign an identical /32 IP address to their respective Loopback1 interface.

As seen below, each MLAG pair will have a unique 'shared' IP address for Loopback1. However, the VXLAN interface configuration is consistent and can be replicated across all MLAG pairs.

A-LEAF2A (MLAG Peer)

```
interface Loopback1
   ip address 2.2.2.2/32
```

A-LEAF2B (MLAG Peer)

```
interface Loopback1
   ip address 2.2.2.2/32
```

Once this configuration is in place, all EVPN BGP Updates sourced from either node in the MLAG domain will have a next-hop attribute of 2.2.2.2, as opposed to that node's locally unique Loopback0 IP address. This helps prevent traffic destined to workloads connected to an MLAG domain from being polarized to a particular VTEP within that MLAG domain.

Notice that there are no per-VTEP unique variables in either Standalone or MLAG VTEP configuration. Hence, this can be replicated on all Leaf switches within the environment, with the only decision being if the VTEP is an MLAG peer, or stand-alone device.

```
Note: The source of a VXLAN tunnel must be a Loopback.
```

Map VLANs to VXLAN VNIs

```
interface Vxlan1
   vxlan vlan 10-1000 vni 10010-11000
```

Once VTEP functionality has been enabled, VLANs can now be mapped to the appropriate VNI that will be used within the VXLAN header to provide the L2VPN service.

The above configuration uses a configuration optimization technique to map VLANs to the appropriate VNIs. Instead of a line-by-line mapping of VLAN to VNI, this methodology instead enables the operator to map a range of VLANs to a range of VNIs. With the above configuration, VLAN 10 will be mapped to VNI 10010, VLAN 50 mapped to VNI 10050, etc.

It is important to note that just because a VLAN has been mapped to a VNI, it does not require that the VLAN itself be created on the VTEP. Nor does the mapping automatically create a VLAN on the switch. Therefore, pre-defining VLAN to VNI mappings does not result in any unnecessary control-plane state or hardware resource consumption on the VTEP where the mapping is defined.

This approach enables the operator to pre-provision VLAN to VNI mappings, reducing the steps necessary when defining a new L2VPN service for a broadcast domain within the environment.

Once a VLAN is mapped to a VNI, the VLAN becomes a switch-local construct. The VNI is now leveraged between VTEPs to signal L2VPN data-plane operations. While it is technically possible to map different VLAN IDs into the same global VNI, it is not recommended.

It is recommended to keep VLAN to VNI mappings consistent across all VTEPs that provide L2VPN services. This helps ensure the repeatability of configuration, as well as the ease of ongoing operations.

Create MAC-VRFs in BGP

Option A: VLAN Aware Bundle

```
Router bgp 65101
   vlan-aware-bundle TENANT-A
      rd 1.1.1.101:1
      route-target both 1:1
      redistribute learned
      vlan 10-49
   !
   vlan-aware-bundle TENANT-B
      rd 1.1.1.101:2
      route-target both 2:2
      redistribute learned
      vlan 50-69
```

Option B: VLAN-Based

```
router bgp 65101
   vlan 10
      rd 1.1.1.2:10010
      route-target both 10010:10010
      redistribute learned
   !
   vlan 11
      rd 1.1.1.2:10011
      route-target both 10011:10011
      redistribute learned
   !
   <...omitted...>
   !
   vlan 49
      rd 1.1.1.2:10049
      route-target both 10049:10049
      redistribute learned
```

```
!
vlan 50
   rd 1.1.1.2:10050
   route-target both 10050:10050
   redistribute learned
!
vlan 51
   rd 1.1.1.2:10051
   route-target both 10051:10051
   redistribute learned
!
<...omitted...>
!
vlan 69
   rd 1.1.1.2:10069
   route-target both 10069:10069
   redistribute learned
```

Up to this point, VLANs have been created, VTEP functionality has been enabled, and VLAN to VNI mappings have been completed. However, L2VPN services are not yet functional as two key components are missing:

Definition of a flood list to ensure BUM traffic is distributed to all VTEPs servicing the L2VPN from where the BUM traffic originated

Control-Plane state mapping workload MAC addresses to VTEPs

The EVPN control-plane provides the following solutions to address these two missing components:

- Dynamic population of VXLAN flood list for BUM traffic distribution (via EVPN Type-3 Route, commonly referred to as 'IMET' route)
- The advertising of real-time information for hosts that exist within these broadcast domains (via EVPN Type-2 Route, commonly referred to as 'MAC-IP' route)

Note that there are two methods of creating MAC-VRFs within BGP: VLAN Aware Bundle and VLAN-Based. A VLAN can only be mapped to a single MAC-VRF configuration method, making these options mutually exclusive on a per VLAN basis. Additionally, in multi-vendor deployments, interoperability considerations must be accounted for when choosing which MAC-VRF method to implement. Most vendor implementations of EVPN today support VLAN-Based, but not all support VLAN Aware Bundles.

```
Note: Always test interoperability between vendor EVPN control-plane implementations
prior to deployment.
```

Each MAC-VRF configuration option will be explored in greater detail below, with an initial high level comparison listed for reference:

VLAN Aware Bundle

- Uses Route-Targets to determine whether or not to accept a received EVPN route
- Uses the Ethernet Tag ID, found within the EVPN NLRI, to import state received via EVPN routes into the proper bridge table (MAC address-table)
- Multiple VLANs can be associated with a single MAC-VRF, defined within BGP
- Adheres to EVPN control-plane standards defined in RFC7432 and RFC8365
- Use caution in multi-vendor deployments, as this method is not widely supported across vendors

VLAN-Based

- Uses Route-Targets to determine whether or not to accept a received EVPN route
- Uses the L2VNI, found within the EVPN NLRI, to import state received via EVPN routes into the proper bridge table (MAC address-table)
- Unique MAC-VRF defined in BGP for each VLAN configured with this method
- Adheres to EVPN control-plane standards defined in RFC7432 and RFC8365
- Ideal for multi-vendor deployments, as this method is widely supported across vendors

When choosing to implement VLAN Aware-Bundles, each tenant can have its respective VLANs mapped to a single MAC-VRF. Like the VLAN to VNI mapping detailed in the previous section, this is a configuration optimization technique. It enables multiple VLANs and their associated VNIs to be mapped to a single MAC-VRF and it's associated Route-Distinguisher and Route-Target(s). However, in a multi-vendor environment, the decision to use VLAN Aware Bundles should only be made after interoperability testing has been completed between vendor platforms. As of this writing, VLAN Aware Bundles and their use of the Ethernet Tag within the EVPN NLRI is not widely supported across vendors.

```
Note: When using VLAN-Aware Bundles, a consistent mapping of VLANs to VLAN-Aware
Bundle(s) should be maintained on all VTEPs that are providing L2VPN services via this
MAC-VRF methodology.
```

With a VLAN-Based approach, each VLAN is mapped to its own unique MAC-VRF within BGP. This methodology is widely supported across vendors, as it uses the L2VNI within the EVPN NLRI to import state into the proper bridge table, and does not make use of the Ethernet Tag ID.

Just as with VLAN to VNI mappings, the mapping of a VLAN to a MAC-VRF within BGP (regardless of the MAC-VRF method chosen) does not require that the VLAN exists on the VTEP, nor does it automatically create the VLAN on the VTEP when the mapping is put into place. This enables the operator to pre-provision mappings in BGP to further ease the administrative overhead associated with the creation of future L2VPN services for each tenant.

The Route-Distinguishers in the examples above are populated using the following methodologies:

- VLAN Aware Bundle RD = "vxlan-source-address:Tenant ID"
- VLAN-Based RD = "vxlan-source-address:L2VNI"

It is important to note that this is entirely operator defined, and ultimately the RD mapping should be set as something that is meaningful to the operations team supporting the EVPN control-plane.

Note that the Route-Distinguisher value is always unique, even on peers within an MLAG domain. This is intentional, as configuring unique Route-Distinguishers will provide the following benefits:

- Reduced convergence time
  - Multiple copies of an EVPN route for a common host-route or prefix are maintained within the BGP table. These copies are each unique, even if they have an identical NEXT_HOP, because of the unique RD value on the originating VTEP
  - If one of these EVPN routes is withdrawn, the other copy originated from a different VTEP is already in the BGP table and ready to be promoted to the 'best' path and inserted into the RIB
- Increased troubleshooting efficiency
  - Validation that a remote VTEP is originating an EVPN route can be performed by specifying the remote VTEP's RD in the associated show command(s)
- Enable ECMP
  - If two VTEPs, each with a unique vxlan source address, are advertising reachability to HostX, then having a unique RD on each VTEP will enable installation of both EVPN routes towards HostX into the RIB
  - If identical RDs were used, then there would be no way to uniquely identify the BGP EVPN update, and only one instance of the route would be imported into the BGP table

Each MAC-VRF will receive its own unique Route-Target value, which will be placed into the Extended Communities field of the EVPN NLRI originated by a VTEP. This value will be used by a VTEP receiving an EVPN Update to determine whether or not to accept the update. Just like the Route-Distinguisher, this is an operator-defined value, and should be set to a value that is meaningful to the team managing the EVPN control-plane.

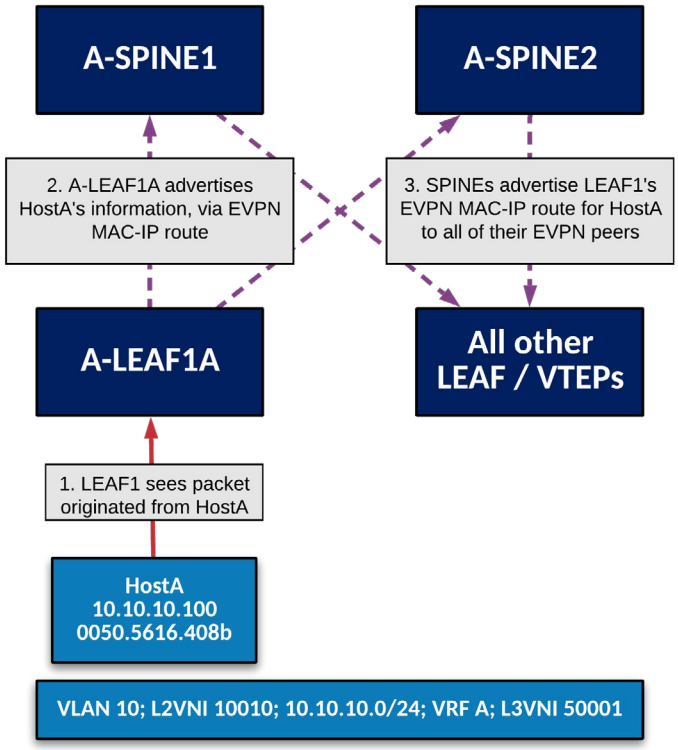
When using VLAN aware bundles, we have multiple VLANs/VNIs mapped to the same MAC-VRF. Therefore, a VTEP must have a method to determine which local bridge table (MAC address-table) that the information found in the update gets imported into. The Ethernet Tag ID, included in the NLRI of Type-2 (MAC-IP) and Type-3 (IMET) EVPN routes, is used to make this determination. The Ethernet Tag is set to the value of the VNI associated with the VLAN where the EVPN update originated. VTEPs use this Ethernet Tag ID to properly define per-VNI flood lists, and to import information into the proper bridge table.

With a VLAN-Based MAC-VRF configuration, the Route-Target is used to determine whether or not the advertisement is accepted. Once accepted, the L2VNI encoded into the NLRI of the EVPN update is used to import the information contained within the EVPN route into the proper bridge table.

In order to ensure EVPN control-plane state is originated for endpoints within a MAC-VRF, the 'redistribute-learned' command is used.

Once a MAC-VRF configuration method has been chosen, and the above configuration is in place, Type-2 (MAC-IP) and Type-3 (IMET) EVPN routes will be originated and accepted for locally configured VLANs. Examples of these Route-Types can be seen below:

EVPN Type-2 (MAC-IP) Route Example:



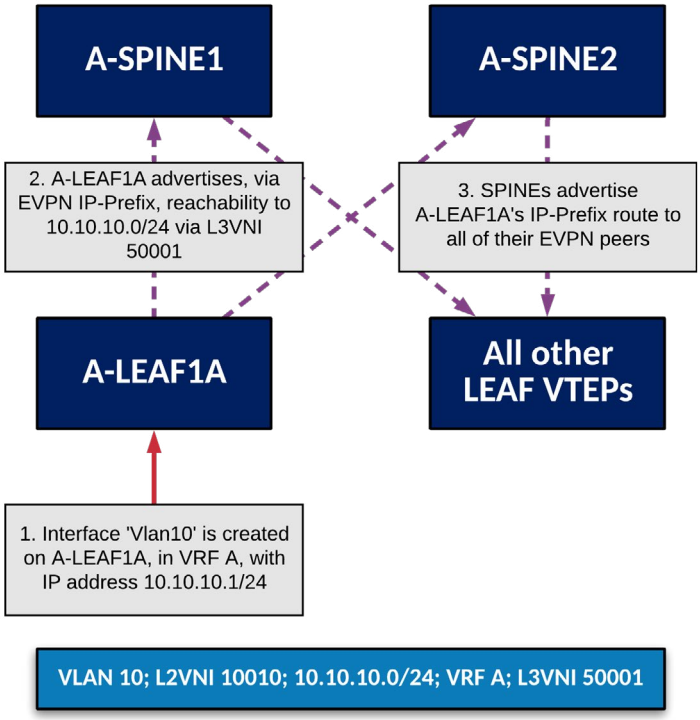
In the above example, A-LEAF1A originates and advertises the following information about HostA, via the EVPN address-family:

- MAC Address
- IP Address (/32 Host Route)
- L2VNI
- L3VNI
- VTEP Location
- Mobility Tracking Number
- Route-Targets (VRF, MAC-VRF)
- Route-Distinguisher

```
Note: L3VNI, Host IP Address, and VRF Route-Target(s) are only included in 'Dual-
VNI' (L2VNI and L3VNI) Type-2 Routes. These routes are automatically generated if the
VTEP has Layer3 presence (an SVI for example) on the Host's broadcast domain. The
origination of Dual-VNI Type-2 Routes can be disabled if desired.
```

EVPN Type-3 (IMET) Route Example

The Inclusive Multicast Ethernet Tag (IMET) EVPN route enables a VTEP to signal it's interest in receiving all BUM traffic for a given VNI. This is shown in the example below:



In the above example, all VTEPs that receive A-LEAF1A's IMET route will add A-LEAF1A's VXLAN source-interface address (in this case Loopback1, the EVPN NEXT_HOP address) to their flood-list for the VNI that A-LEAF1A defined within the IMET route. This is how the flood-list is dynamically populated in real-time via the EVPN control-plane.

Since the flood-list is dynamically populated using the NEXT_HOP address listed within the Type-3 (IMET) route, only a single flood-list entry will exist towards VTEPs that are MLAG peers. This is because the MLAG peer VTEPs use an identical NEXT_HOP address when originating their EVPN routes.

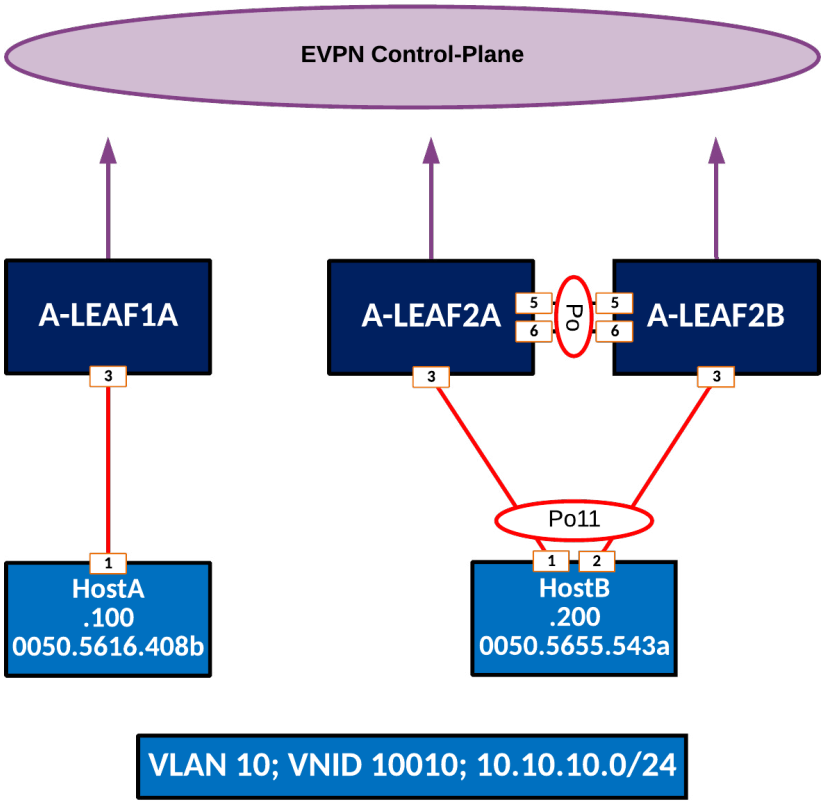
```
Note: A VTEP will originate a Type-3 (IMET) route for every locally configured VLAN
that has been mapped to a VNI and MACVRF.
```

Tenant L2VPN Service Validation

The focus of L2VPN service functionality will be within the scope of Tenant-A, VLAN 10. Specifically, reachability between HostA and HostB.

A-LEAF1A will be used as the reference point for validation commands.

Details of HostA and HostB can be found below:



Since the focus of this section is L2VPN services, the focus will be placed upon the MAC addresses of these respective hosts.

```
A-LEAF1A#show vlan brief
VLAN  Name                             Status    Ports
----- -------------------------------- --------- -------------------------
1     default                          active
10    Ten                              active    Et3, Vx1
<...Output Omitted...>
```

VLAN 10 has been created, and is active on the expected interfaces. Ethernet3 is a connection point for a workload within TenantA, VLAN 10. For reference, the configuration of Ethernet3 is shown below:

```
interface Ethernet3
   description HostA
   switchport access vlan 10
   spanning-tree portfast
```

Validate VXLAN interface status and VLAN to VNI Mappings:

```
A-LEAF1A#show interface vxlan1
Vxlan1 is up, line protocol is up (connected)
  Hardware is Vxlan
  Source interface is Loopback1 and is active with 2.2.2.1
  Replication/Flood Mode is headend with Flood List Source: EVPN
 Remote MAC learning via EVPN
  VNI mapping to VLANs
  Static VLAN to VNI mapping is
    [10, 10010]       [11, 10011]       [12, 10012]       [13, 10013]
<...Output Omitted...>
```

The above output confirms the following:

- The Vxlan1 interface is up
- Flood list is dynamically populated via the EVPN control-plane
- Remote MAC addresses are learned via the EVPN control-plane
- All expected VLAN to VNI mappings exist

As covered in the previous section, even though mappings exist for VLANs that have not yet been created on the local VTEP, there is no penalty from a control-plane or hardware consumption perspective. These VLANs that do not yet exist do not have any originated/accepted control-plane state, and are not consuming entries in the VTEPs local MAC address-table.

Verify Flood-List:

```
A-LEAF1A#show vxlan flood vtep vlan 10
          VXLAN Flood VTEP Table
------------------------------------------------------------------------

VLANS                            Ip Address
-------------------------------  ----------------------------------------
10                               2.2.2.2
```

This output lists the VTEPs that have signaled interest in receiving BUM traffic originating in VLAN 10. Given that the only other VTEPs providing L2VPN service for VLAN 10 are A-LEAF2A and A-LEAF2B, this output matches what was expected.

Notice that A-LEAF2A and A-LEAF2B, while separate VTEPs, are not listed separately in the above output. This is because they are MLAG peers and have been configured to present themselves as a single logical VTEP from an EVPN control-plane standpoint.

Verify VXLAN Address-Table:

```
A-LEAF1A#show vxlan address-table vlan 10
          Vxlan Mac Address Table
----------------------------------------------------------------------

VLAN  Mac Address     Type      Prt  VTEP             Moves   Last Move
----  -----------     ----      ---  ----             -----   ---------
  10  0050.5655.543a  EVPN      Vx1  2.2.2.2          1       3:18:07 ago
Total Remote Mac Addresses for this criterion: 1
```

As expected, the MAC address associated with HostB is shown as residing behind the VTEP 2.2.2.2(MLAG Logical VTEP).

Also of note is that this MAC address was learned and imported into the VLAN 10 instance of the local MAC address-table, via EVPN.

Check MAC Address-Table Mappings

```
A-LEAF1A#show mac address-table vlan 10
          Mac Address Table
------------------------------------------------------------------

Vlan    Mac Address       Type        Ports      Moves   Last Move
----    -----------       ----        -----      -----   ---------
  10    0050.5616.408b    DYNAMIC     Et3        1       4:06:09 ago
  10    0050.5655.543a    DYNAMIC     Vx1        1       4:02:42 ago
Total Mac Addresses for this criterion: 2
```

It's important that the MAC address-table entries for endpoints that exist behind remote VTEPs are consistent with the VXLAN address-table. In the output above, HostB's MAC address is shown as reachable via port Vx1, which is expected.

HostA's MAC address is locally connected, which implies that A-LEAF1A should be originating reachability information for this host via EVPN. This will be verified next.

```
Note: Throughout the remainder of this section, the output of validation commands will
be listed for both VLAN Aware Bundle and VLAN-Based MAC-VRF configuration models.
```

Validate EVPN Type-2 (MAC-IP) Routes

VLAN Aware Bundle:

```
A-LEAF1A#show bgp evpn route-type mac-ip vni 10010
<...Output Omitted...>

         Network                Next Hop            Metric  LocPref Weight  Path
 * >     RD: 1.1.1.101:1 mac-ip 10010 0050.5616.408b
                                -                   -       -     0     i
 * >Ec   RD: 1.1.1.102:1 mac-ip 10010 0050.5655.543a
                                2.2.2.2             -       100     0     65001 65102 i
<...Output Omitted...>
 * >Ec   RD: 1.1.1.103:1 mac-ip 10010 0050.5655.543a
                                2.2.2.2             -       100     0     65001 65102 i
<...Output Omitted...>
```

VLAN-Based MAC-VRF:

```
A-LEAF1A#show bgp evpn route-type mac-ip vni 10010
<...Output Omitted...>
         Network                Next Hop            Metric  LocPref Weight  Path
 * >     RD: 1.1.1.101:10010 mac-ip 0050.5616.408b
                                -                     -       -       0       i
 * >Ec   RD: 1.1.1.102:10010 mac-ip 0050.5655.543a
                                2.2.2.2               -       100     0       65001 65102 i
<...Output Omitted...>
 * >Ec   RD: 1.1.1.103:10010 mac-ip 0050.5655.543a
                                2.2.2.2               -       100     0       65001 65102 i
<...Output Omitted...>
```

As expected, A-LEAF1A is originating reachability information for HostA. Additionally, reachability information for HostB is being originated via the single logical VTEP MLAG domain of A-LEAF2A and A-LEAF2B.

Note that in the VLAN-Based MAC-VRF output, the Ethernet Tag of "10010" is not present between "mac-ip" and the MAC address of the endpoint.

Also note that even though both MLAG peers are advertising the same next-hop IP address within the NLRI, their routes are treated as two independent unique routes and imported into the BGP table. In this scenario, if A-LEAF2A were to have its route withdrawn, convergence time would be minimal as A-LEAF2B's route already exists within A-LEAF1A's BGP table.

Below is a closer look at the details of the reachability information being advertised for HostB:

VLAN Aware Bundle MAC-VRF

```
A-LEAF1A#show bgp evpn route-type mac-ip 0050.5655.543a detail
<...Output Omitted...>
BGP routing table entry for mac-ip 10010 0050.5655.543a, Route Distinguisher:
1.1.1.102:1
 Paths: 2 available
  65001 65102
    2.2.2.2 from 1.1.1.201 (1.1.1.201)
      Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP head, best,
ECMP contributor
      Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
      VNI: 10010 ESI: 0000:0000:0000:0000:0000
<...Output Omitted...>
BGP routing table entry for mac-ip 10010 0050.5655.543a, Route Distinguisher:
1.1.1.103:1
 Paths: 2 available
  65001 65102
    2.2.2.2 from 1.1.1.201 (1.1.1.201)
      Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP head, best,
ECMP contributor
      Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
      VNI: 10010 ESI: 0000:0000:0000:0000:0000
<...Output Omitted...>
```

Note the presence of the Ethernet Tag ID in the above output. As covered earlier, this only exists if a VLAN aware bundle is in use. In the above output, the Ethernet Tag ID of "10010" can be found in the following line:

```
BGP routing table entry for mac-ip 10010 0050.5655.543a, Route Distinguisher:
1.1.1.2:1
```

If the presence of the Ethernet Tag ID value in EVPN NLRI is not supported by another vendor's implementation of the EVPN control-plane, then VLAN-Based MAC-VRFs must be used for interoperability.

For comparison, the output of the same command when using the VLAN-Based MAC-VRF approach can be seen below.

VLAN-Based MAC-VRF

```
A-LEAF1A#show bgp evpn route-type mac-ip 0050.5655.543a detail
<...Output Omitted...>
BGP routing table entry for mac-ip 0050.5655.543a, Route Distinguisher:
1.1.1.102:10010
 Paths: 2 available
  65001 65102
    2.2.2.2 from 1.1.1.201 (1.1.1.201)
      Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP head, best,
ECMP contributor
      Extended Community: Route-Target-AS:1:10010 TunnelEncap:tunnelTypeVxlan
      VNI: 10010 ESI: 0000:0000:0000:0000:0000
<...Output Omitted...>
BGP routing table entry for mac-ip 0050.5655.543a, Route Distinguisher:
1.1.1.103:10010
 Paths: 2 available
  65001 65102
    2.2.2.2 from 1.1.1.202 (1.1.1.202)
      Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP head, best,
ECMP contributor
      Extended Community: Route-Target-AS:1:10010 TunnelEncap:tunnelTypeVxlan
      VNI: 10010 ESI: 0000:0000:0000:0000:0000
<...Output Omitted...>
```

Note: For the remainder of this document, all validation output will be based on the VLAN Aware Bundle MAC-VRF configuration method. The only difference in this output vs. the VLAN-Based approach, is the presence of the Ethernet Tag ID. All other fields within the NLRI are identical.

Validate EVPN Type-3 (IMET) Routes

```
A-LEAF1A#show bgp evpn route-type imet vni 10010
<...Output Omitted...>

         Network                Next Hop            Metric  LocPref Weight  Path
 * >Ec    RD: 1.1.1.102:1 imet 10010 1.1.1.2
                                2.2.2.2               -       100     0     65001 65102 i
<...Output Omitted...>
 * >Ec    RD: 1.1.1.103:1 imet 10010 1.1.1.2
                                2.2.2.2               -       100     0     65001 65102 i
<...Output Omitted...>
 * >      RD: 1.1.1.101:1 imet 10010 2.2.2.1
                                -                     -       -       0     i
```

As expected, A-LEAF1A is signaling it's interest in receiving BUM traffic for VNI 10010.

As with the Type-2 (MAC-IP) routes, the MLAG pair of A-LEAF2A and A-LEAF2B are signaling their intent to receive BUM traffic for VNI 10010, but still present themselves as a single logical VTEP via an identical Next Hop value. This ensures that BUM traffic within this broadcast domain is load-shared between the two MLAG peers, and that only one copy of the BUM traffic is sent from the originating VTEP.

Also note the presence of the Ethernet Tag ID, which can be found right after the 'imet' acronym for each respective Type-3 route.

```
A-LEAF1A#show bgp evpn route-type imet vni 10010 detail
<...Output Omitted...>
BGP routing table entry for imet 10010 2.2.2.2, Route Distinguisher: 1.1.1.102:1
 Paths: 2 available
  65001 65102
    2.2.2.2 from 1.1.1.201 (1.1.1.201)
      Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP head, best,
ECMP contributor
      Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
      VNI: 10010
<...Output Omitted...>
BGP routing table entry for imet 10010 2.2.2.2, Route Distinguisher: 1.1.1.103:1
 Paths: 2 available
  65001 65102
    2.2.2.2 from 1.1.1.201 (1.1.1.201)
      Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP head, best,
ECMP contributor
      Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
      VNI: 10010
<...Output Omitted...>
BGP routing table entry for imet 10010 2.2.2.1, Route Distinguisher: 1.1.1.101:1
 Paths: 1 available
  Local
    - from - (0.0.0.0)
      Origin IGP, metric -, localpref -, weight 0, valid, local, best
      Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
      VNI: 10010
```

A more detailed look at IMET routes that exist for VNI 10010 shows the expected information. The appropriate route-targets, route-distinguishers and VNIs are all listed.

At this point, the EVPN control-plane has been validated, and reachability between HostA and HostB exists via L2VPN service.

Tenant Layer3 VPN Configuration

```
hardware tcam profile vxlan-routing
!Above command only required on R Series Platforms
!
vrf definition A
!
vrf definition B
!
interface Vxlan1
   vxlan vrf A vni 50001
   vxlan vrf B vni 50002
!
router bgp 65101
   vrf A
      rd 1.1.1.101:1
      route-target import evpn 1:1
      route-target export evpn 1:1
      redistribute connected
   !
   vrf B
      rd 1.1.1.101:2
      route-target import evpn 2:2
      route-target export evpn 2:2
      redistribute connected
!
ip virtual-router mac-address 00:1c:73:aa:bb:cc
!
interface Vlan10
   vrf forwarding A
   ip address virtual 10.10.10.1/24
!
interface Vlan50
   vrf forwarding B
   ip address virtual 50.50.50.1/24
!
interface Loopback201
   vrf forwarding A
   ip address 201.0.0.101/32
!
interface Loopback202
   vrf forwarding B
   ip address 202.0.0.101/32
!
ip address virtual source-nat vrf A address 201.0.0.101
ip address virtual source-nat vrf B address 202.0.0.101
```

For the duration of this section, A-LEAF1A will continue to be used as the primary reference point for configuration, with MLAG specific configuration being called out when necessary using A-LEAF2A and A-LEAF2B for reference.

The EVPN control-plane has the unique ability to provide both Layer2 and Layer3 VPN services via a single address-family.

Up to this point, the term VNI has been used to define the value that a VLAN is mapped to in order to provide L2VPN services via VXLAN data-plane encapsulation. However, with the introduction of Layer3 VPN services, there is a need to differentiate between a VNI that is used to signal L2VPN service membership, and a VNI that is used to signal L3VPN service membership. These VNIs are referred to as the L2VNI and L3VNI, respectively.

A brief overview of the differences between the L2VNI and L3VNI can be found below for reference:

L2VNI

- Defines an L2VPN Service
- Mapped to a VLAN
- Used in VXLAN Bridging Operations between VTEPs
- Data-Plane mechanism, signaled via the Control-Plane
- Encoded into all EVPN Type-2 (MAC/MAC-IP) and Type-3 (IMET) updates

L3VNI

- Defines an L3VPN Service
- Mapped to an IP VRF
- Used for VXLAN Routing Operations between VTEPs
- Data-Plane mechanism, signaled via the Control-Plane
- Encoded into all EVPN Type-2 (MAC-IP) and Type-5 (IP Prefix) updates

The focus of this section will be on the use of the L3VNI to provide L3VPN services to Tenants.

Create VRFs

```
vrf definition A
!
vrf definition B
```

The creation of VRFs is the first step to providing L3VPN services, with each tenant receiving their own unique VRF.

Independent control-plane state, such as IPv4 Unicast forwarding information in the RIB, is maintained for each respective tenant's VRF.

By default, communication between VRFs is prohibited as routes are not known, or exchanged, between VRFs. Each tenant's VRF maintains its own isolated control-plane, which is extended end-to-end across the environment via the EVPN address-family.

Map VRF(s) to L3VNI(s)

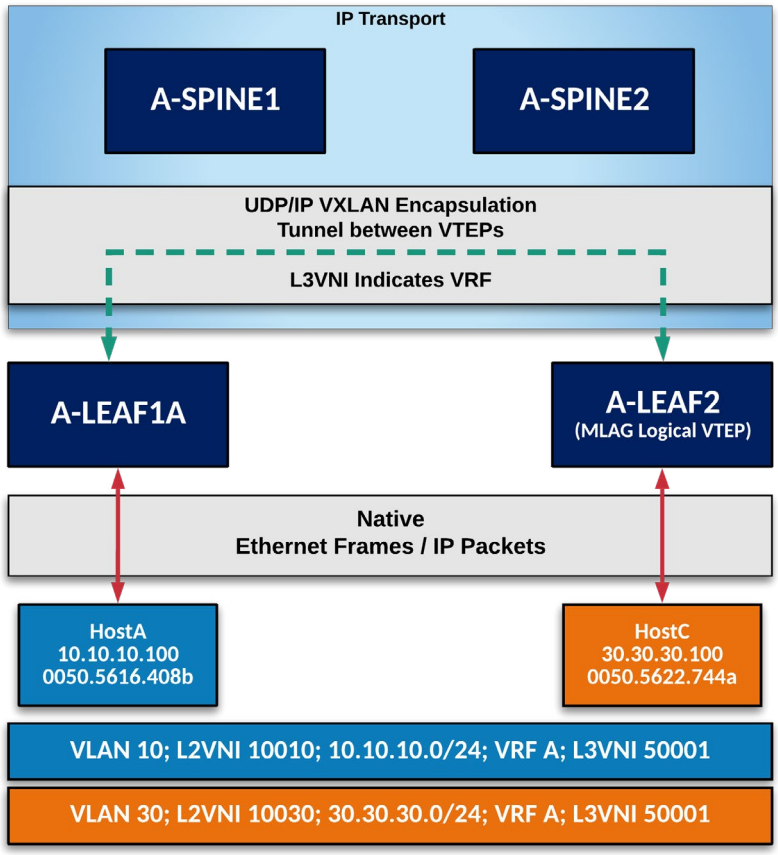
```
interface Vxlan1
   vxlan vrf A vni 50001
   vxlan vrf B vni 50002
```

When VRFs are created, they become a local construct on the switch upon which they were initiated. When a tenant's VRF exists on multiple VTEPs, these VTEPs will need a way to signal to each other, via the data-plane, which VRF an incoming VXLAN encapsulated packet should be forwarded in. This is done by mapping a VRF to an L3VNI.

Once this mapping is in place, the ingress VTEP places the L3VNI associated with the VRF into the VXLAN header when the packet is sent towards the egress VTEP.

The egress VTEP, upon receipt of the VXLAN encapsulated packet, inspects the VXLAN header and sees the L3VNI. This VTEP will then use whichever VRF is locally mapped to the value of the L3VNI for lookup and forwarding operations associated with the inner-IP header.

This process, at a high level, is illustrated below:



It is important to note that once a VRF has been mapped to an L3VNI, Symmetric IRB will then be used for all VXLAN routing operations between VTEPs.

It is technically possible to leverage VRFs without mapping them to L3VNIs. This approach would require Asymmetric IRB between VTEPs, with all VLANs and VRFs instantiated on all VTEPs in order for end-to-end reachability to be maintained.

When using Symmetric IRB, the ingress VTEP does not require L3 presence on the destination subnet, and the egress VTEP does not require L3 presence on the source subnet. For a detailed overview of Symmetric and Asymmetric IRB operations and their associated advantages and disadvantages, refer to the EVPN Overview and Nomenclature section of this document.

```
Note:  Symmetric IRB is recommended, as it does not require that all VLANs and VRFs
exist everywhere to guarantee host reachability. Additionally, with Symmetric IRB,
Remote ARP entries are maintained in software instead of consuming hardware resources.
Thus making it a more scalable solution.
```

Enable EVPN L3VPN Control-Plane in BGP

Standalone VTEP

```
router bgp 65101
   vrf A
      rd 1.1.1.101:1
      route-target import evpn 1:1
      route-target export evpn 1:1
      redistribute connected
   !
   vrf B
      rd 1.1.1.101:2
      route-target import evpn 2:2
      route-target export evpn 2:2
      redistribute connected
```

MLAG VTEP A-LEAF2A

```
router bgp 65101
   vrf A
      rd 1.1.1.102:1
      route-target import evpn 1:1
      route-target export evpn 1:1
      redistribute connected
   !
   vrf B
      rd 1.1.1.102:2
      route-target import evpn 2:2
      route-target export evpn 2:2
      redistribute connected
```

MLAG VTEP A-LEAF2B

```
router bgp 65101
   vrf A
      rd 1.1.1.103:1
      route-target import evpn 1:1
      route-target export evpn 1:1
      redistribute connected
   !
   vrf B
      rd 1.1.1.103:2
      route-target import evpn 2:2
      route-target export evpn 2:2
      redistribute connected
```

Similar to the MAC-VRFs created in the previous L2VPN section, IP VRFs will be added into BGP in order to provide L3VPN services via the EVPN control-plane.

As stated in the L2VPN section of this document, the Route-Distinguishers in the examples above are populated using a methodology of "vxlan-source-address:Tenant ID". It is important to note that this is entirely operator defined, and ultimately the mapping should be set as something that is meaningful to the operations team supporting the EVPN control-plane.

Note that the Route-Distinguisher value is unique on the two MLAG peers. This is intentional, as covered in the previous section. Having unique Route Distinguisher values on all VTEPs improves convergence time, assists in troubleshooting, and enables ECMP.

Each tenant's VRF will receive its own unique Route-Target value, which will be placed into the Extended Communities field of the EVPN NLRI originated by a VTEP. This value will also be used by a VTEP receiving an EVPN Update to determine whether or not to accept the update. Just like the Route-Distinguisher, this is an operator-defined value, and should be set to a value that is meaningful to the team managing the EVPN control-plane.

The redistribute connected statement will ensure that any locally configured IP address within a VRF automatically have its associated IP Prefix advertised into BGP with the appropriate Route-Distinguisher, Route-Target and L3VNI.

Create SVIs for Tenant Subnets

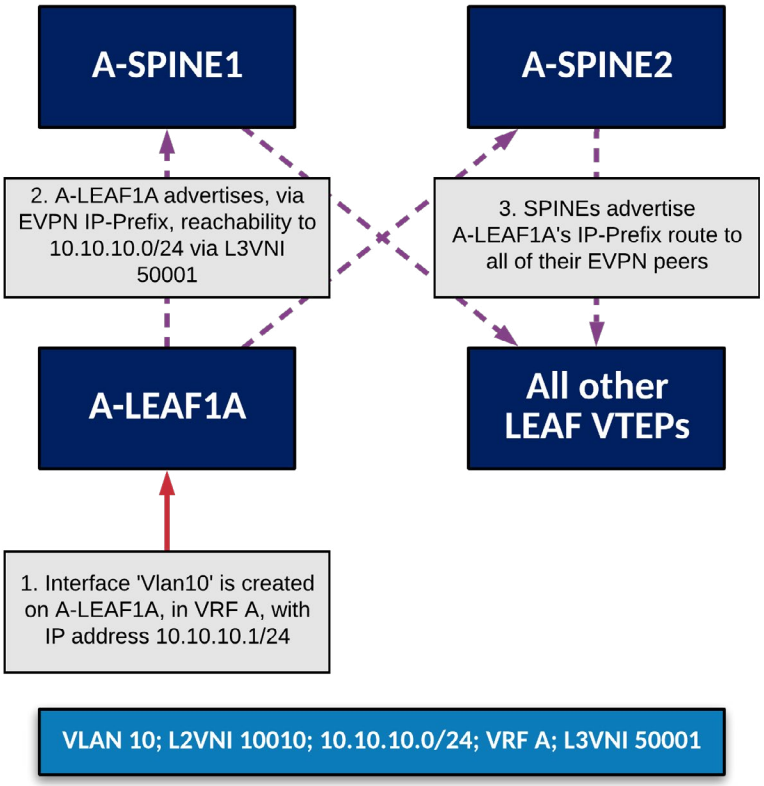
```
ip virtual-router mac-address 00:1c:73:aa:bb:cc
!
interface Vlan10
   vrf forwarding A
   ip address virtual 10.10.10.1/24
!
interface Vlan50
   vrf forwarding B
   ip address virtual 50.50.50.1/24
```

The above configuration can be duplicated on any other VTEPs servicing VLAN 10 and VLAN 50 in Tenants A and B respectively. The fact that the IP address for an SVI will match on all VTEPs is not a cause for concern regarding duplicate IP address assignment. This is commonly referred to as an Anycast Gateway, and is configured through the use of the 'ip address virtual' command under the SVI. The purpose of the Anycast Gateway is to ensure a workload is always directly attached to its First Hop Gateway, regardless of which VTEP it resides behind.

Additionally, the virtual MAC address associated with Anycast Gateway IP addresses is defined within global configuration mode, via the 'ip virtual-router mac-address' command. This is important, as it enables the same virtual MAC address to be used across all subnets where an Anycast Gateway exists. Since Ethernet is link local in nature, the same MAC address existing across multiple VLANs is not an issue.

Note that with an Anycast Gateway, an FHRP (First Hop Redundancy Protocol) such as VRRP or vARP is not necessary.

Each SVI is placed into the VRF associated with the Tenant that it is servicing. When this association occurs, Type-5 (IP Prefix) EVPN routes will be originated by the VTEP for the network prefix defined on the SVI. The process of this route advertisement is illustrated below for reference:



MLAG VTEP Optimal Forwarding and Resiliency

The use of MLAG in an EVPN deployment improves the resiliency of the network and the services it provides. Giving dual-connected hosts the ability to maintain uninterrupted connectivity in the event of an MLAG device outage, planned or unplanned, is important for critical services and applications.

However, when deploying MLAG in combination with EVPN, additional convergence and forwarding optimizations must be considered to ensure maintenance events, and unplanned outages, have minimal impact on data-plane forwarding operations.

A summary of recommendations when deploying MLAG with EVPN can be found below:

1. Unique Route Distinguisher (RD) for each MLAG Peer
   - Covered in section "Tenant Layer2 VPN Configuration"
2. Enable MLAG Shared EVPN Router MAC
3. Adjust MLAG reload delay timers
   - Delay for MLAG interfaces should be shorter than Non-MLAG interfaces
4. Establish iBGP peering in the underlay between MLAG peers
   - Covered in section "IP Underlay Configuration"
5. Establish iBGP peering in overlay VRF(s) between MLAG peers
   - Only required if single-homed subnets exist within a VRF on either MLAG peer

Each of these recommendations, if not covered in a previous section, are covered in detail in the upcoming sections.

Enable MLAG Shared EVPN Router MAC (MLAG Only)

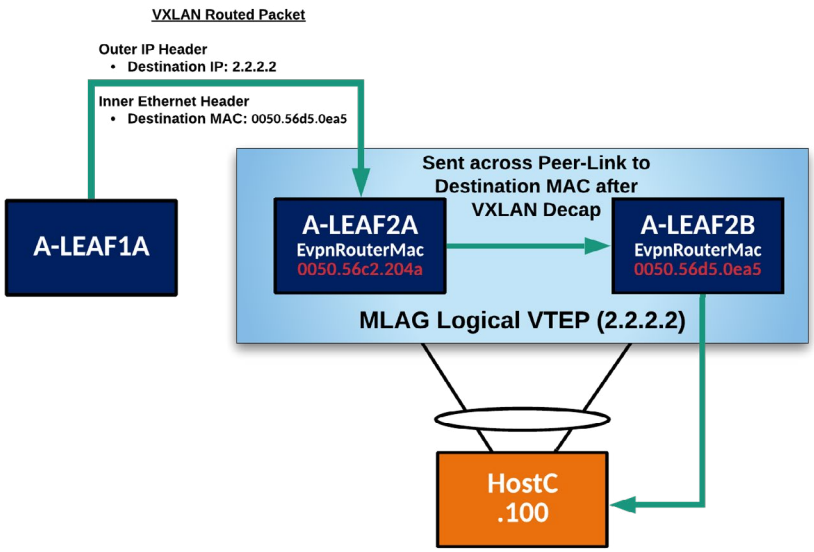
MLAG VTEP

```
interface Vxlan1
   vxlan virtual-router encapsulation mac-address mlag-system-id
```

When a VTEP is providing L3VPN services for a Tenant, it will add an "EvpnRouterMac" Extended Community into the NLRI of the EVPN Type-2 (MAC-IP) and Type-5 (IP Prefix) updates that it originates into BGP.

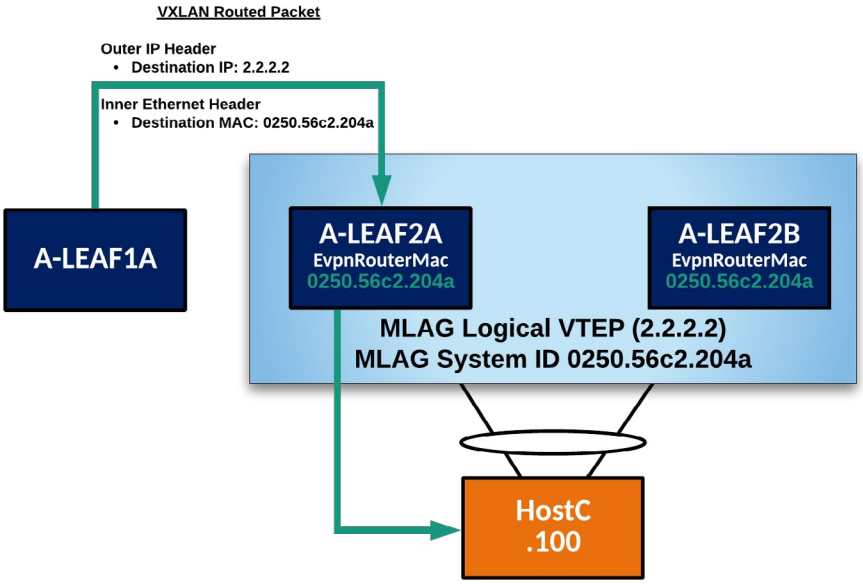
This EvpnRouterMac field is the MAC address that an Ingress VTEP will place into the Destination MAC Address field of the Inner-Ethernet header. This MAC address is only used when performing VXLAN Routing operations via the L3VNI.

Every VTEP maintains its own unique EVPN Router MAC, based on the chassis MAC address. In an MLAG environment, this can cause traffic to unintentionally traverse the peer-link of the MLAG domain. This scenario is illustrated to the right for reference:



This issue is addressed by using the MLAG System ID as the value for the EvpnRouterMac Extended Community in Type-2 (MAC-IP) and Type-5 (IP Prefix) EVPN routes.

Once this has been configured, through the use of the 'vxlan virtual-router encapsulation mac-address mlag-system-id' command, both MLAG peers will locally forward any VXLAN routed packets destined to dual-connected workloads. The scenario below illustrates the solution:



Adjust MLAG Reload-Delay Timers (MLAG Only)

R Series Devices (Jericho ASIC)

```
mlag configuration
   reload-delay mlag 780
   reload-delay non-mlag 1020
```

All other Platforms (Trident/Tomahawk/XP ASIC)

```
mlag configuration
   reload-delay mlag 300
   reload-delay non-mlag 330
```

When performing maintenance on an MLAG peer, it is important that the concept of MLAG and Non-MLAG interfaces is understood and considered.

Upon boot-up, an MLAG switch will place all of its physical interfaces into an 'err-disabled' state. The only exception to this are the Peer-Link interfaces, which are used to establish the Peer-Link port-channel and synchronize Layer-2 control-plane state between the MLAG peers.

```
Note:  Remote MAC addresses learned via EVPN are NOT synchronized between MLAG peers.
```

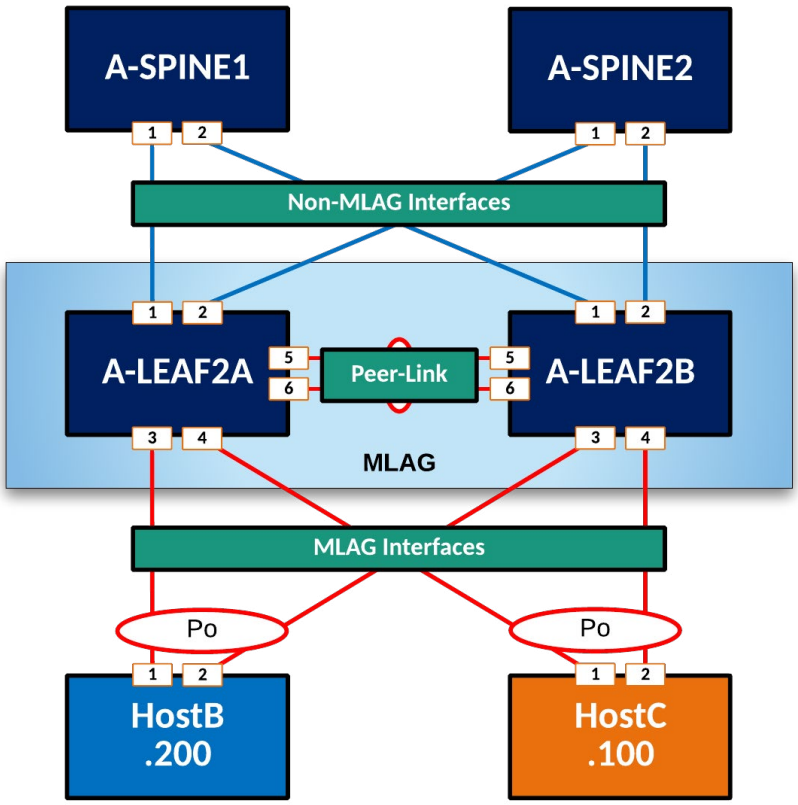
Since the Peer-Link is immediately transitioned into a forwarding state, all VLAN Interfaces are also immediately brought online on the reloaded MLAG peer. This is important, as the SVI for VLAN 4093 is used for iBGP peering between the MLAG peers in the IPv4 Underlay. This means that the reloaded MLAG peer will be able to quickly establish IP reachability to the Loopback interfaces of all Spines and VTEPs.

With this Loopback reachability in place, the reloaded MLAG peer can also establish multi-hop eBGP EVPN peerings, through it's MLAG peer, to the spines. This will allow the reloaded MLAG peer to begin to populate its local MAC address and ARP table with reachability information for remote hosts. This is important, as remote host information learned via EVPN, such as MAC addresses, is not synchronized between MLAG peers.

Each MLAG peer will classify an interface as either MLAG or Non-MLAG. The difference between the interface classifications can be found below:

- **MLAG Interface:** Any physical interface that is a member of an MLAG port-channel
- **Non-MLAG Interface:** Any Layer-3 interface, or any Layer-2 interface that is not a member of an MLAG port-channel

Interface classification is illustrated below:



After an MLAG peer has been reloaded, the reload-delay timer automatically begins counting down once EOS has initialized and the MLAG agent has been started.

As seen in the configuration snippets above, it is recommended that the MLAG interface reload-delay timer be shorter than the Non-MLAG interface reload-delay timer.

iBGP peering in overlay VRF(s) between MLAG peers (MLAG Only)

A-BL1A

```
interface Vlan4000
   description MLAG iBGP Peering: VRF A
   vrf forwarding A
   ip address 192.0.0.1/24
!
router bgp 65104
   vrf A
      neighbor 192.0.0.2 peer-group MLAG-IPv4-UNDERLAY-PEER
```

A-BL1B

```
interface Vlan4000
   description MLAG iBGP Peering: VRF A
   vrf forwarding A
   ip address 192.0.0.2/24
!
router bgp 65104
   vrf A
      neighbor 192.0.0.1 peer-group MLAG-IPv4-UNDERLAY-PEER
```

Deploying VTEPs in MLAG pairs (with the VTEPs presenting themselves as a single logical VTEP via a shared IP address and EvpnRouterMac) is recommended for resiliency and ECMP. However, without additional configuration, this model can result in the isolation of prefixes that are single-homed on an MLAG peer. This scenario is covered in this section.

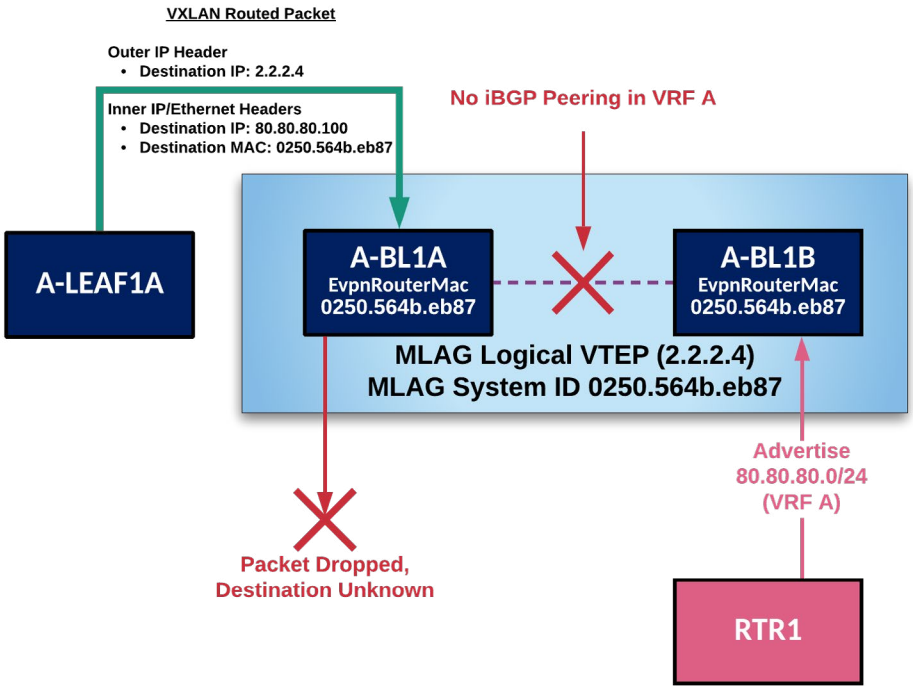
Note that the 192.0.0.0/24 subnet is re-used in the above configuration for iBGP peering in VRF A between the MLAG peers. Additionally, the "MLAG-IPv4-UNDERLAY-PEER" peer-group is re-used. By using the same IP subnet and BGP peer-group for all VRF specific iBGP peerings, the configuration is made repeatable and consistent, regardless of the number of VRFs where this peering is necessary.

```
Note:  The per-VRF peering between MLAG VTEPs is only required to maintain
reachability to prefixes that are only known to one of the two MLAG peers. Common
causes of this are:

1.) Single homed router, advertising reachability information to only one of the MLAG
peers

2.) Loopback interface(s) on the MLAG peers
```

MLAG: Isolated Subnet (Broken)



In the above example, RTR1 is single-homed to A-BL1B. RTR1, via a routing protocol peering in VRF A, is advertising reachability to the 80.80.80.0/24 prefix. A packet destined to an endpoint in the 80.80.80.0/24 network is VXLAN encapsulated via A-LEAF1A, and then sent to the shared VTEP IP address of 2.2.2.4. Since this is a VXLAN routed packet, the L3VNI of 50,001 is used in the VXLAN header.

Because the IP address of 2.2.2.4 is shared across both A-BL1A and A-BL1B, ECMP occurs via the underlay towards this address. In the example above, the traffic hashed to A-BL1A.

Once A-BL1A receives this packet, it examines the VXLAN header and sees the L3VNI of 50,001, indicating that VRF A should be used for the lookup and forwarding operation associated with the inner-IP header.

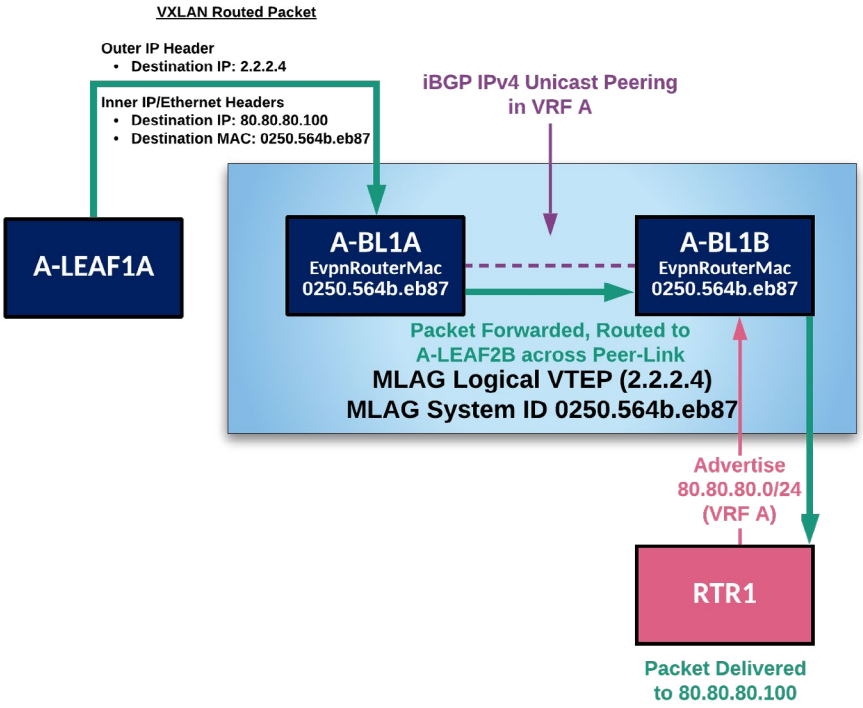
Upon de-encapsulation, A-BL1A sees that the inner-Ethernet header has a destination MAC address of 0250.564b.eb87, which is a MAC address shared between the two MLAG peers.

Considering that A-BL1A owns 0250.564b.eb87, it will perform the routing operation for this packet. Looking at the destination IP address of 80.80.80.100, A-BL1A finds it does not have a route to this network in VRF A, and drops the packet.

```
Note: The details, configuration and reasoning behind the shared MLAG EVPN Router MAC
(Including illustrated examples), are covered in the previous section.
```

In order to resolve this issue, an iBGP IPv4 Unicast peering in VRF A between the MLAG peers can be established.

Once this peering is in place, A-BL1B can advertise reachability to the prefix 80.80.80.0/24, via BGP, to A-BL1A. This will result in the scenario below:



Just as in the first scenario, A-BL1A will de-encapsulate the VXLAN packet and perform the routing operation in VRF A based on the information in the inner-IP header. This time A-BL1A has the 80.80.80.0/24 prefix in its local routing table for VRF A. This prefix was learned via the iBGP adjacency between the MLAG peers.

With this information, A-BL1A routes the native IP packet towards A-BL1B, which will then route the packet towards its final destination behind RTR1.

Ensure VTEPs can ping workloads behind remote VTEPs

A-LEAF1A

```
interface Loopback201
   vrf forwarding A
   ip address 201.0.0.101/32
!
interface Loopback202
   vrf forwarding B
   ip address 202.0.0.101/32
!
ip address virtual source-nat vrf A address 201.0.0.101
ip address virtual source-nat vrf B address 202.0.0.101
```

A-LEAF2A

```
interface Loopback201
   vrf forwarding A
   ip address 201.0.0.102/32
!
ip address virtual source-nat vrf A address 201.0.0.102
```

While the use of the Anycast Gateway is an effective model for FHRP functionality, there is a caveat when a VTEP attempts to ping a workload that resides behind a remote VTEP. If the VTEP is a member of an MLAG domain, this caveat also applies to locally connected workloads.

Because all VTEPs share the same IP address and MAC address for each respective SVI, pings destined to workloads behind remote VTEPs, or local workloads in the case of MLAG VTEPs, may not be successful. This is because when the destination host replies to either the ARP request, or the ICMP echo request, the reply is processed by the first VTEP it arrives at, as it has the same IP and MAC address locally configured as the originating VTEP.

In order to ensure VTEPs are able to successfully ping workloads, a feature was introduced to allow operators to optionally configure a Loopback that can be automatically used as the source address for pings to hosts within a respective VRF.

As shown in the configuration above, there is one Loopback per VRF that exists on the VTEP, and this Loopback is a unique address on each respective VTEP where it is configured. Once this is in place, VTEPs will then be able to ping all workloads within a VRF.

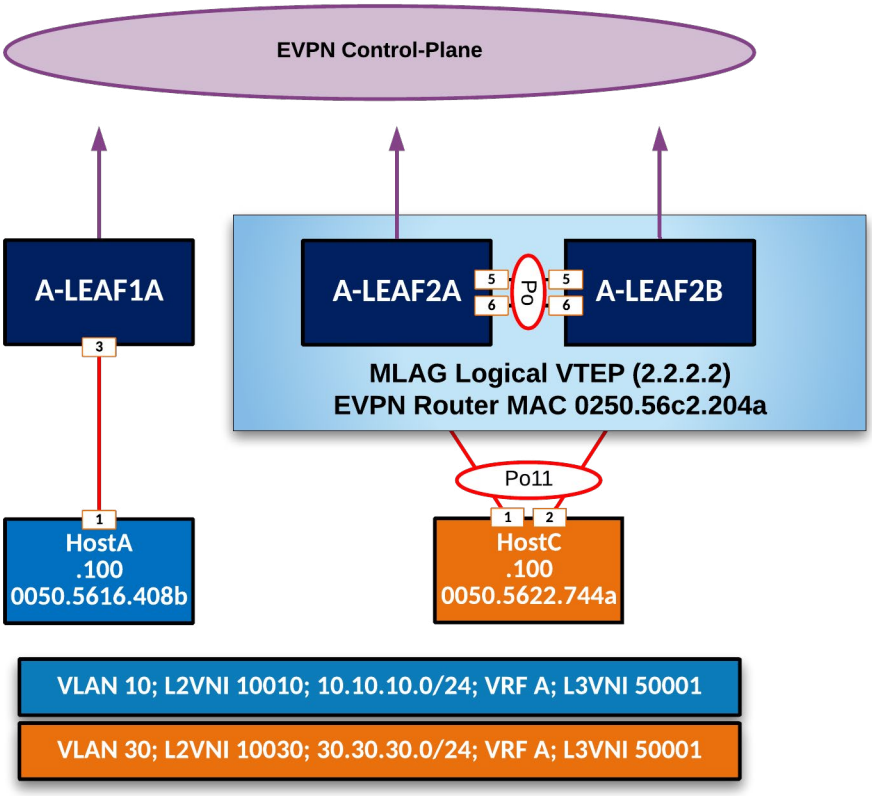
There is no need to manually specify this loopback as the source when issuing a ping command, it is done automatically when this feature is configured.

Tenant L3VPN Service Validation

The focus of L3VPN service validation will be within the scope of the VRF associated with Tenant-A. Specifically, reachability between HostA and HostC.

A-LEAF1A will be used as the reference point for validation commands.

Details of the area of the topology that will be the focus of this validation section can be found below:



Verify VRF Exists, and Routing is Enabled

```
A-LEAF1A#show vrf A
   Vrf     RD              Protocols     State              Interfaces
---------- -------------- ------------ ---------------- ----------------------
   A       1.1.1.101:1     ipv4,ipv6     v4:routing,      Loopback201, Vlan10, Vlan4093
                                         v6:no routing
```

The output above confirms that VRF A has been created, with routing enabled for the IPv4 Unicast address-family. Additionally, all interfaces that have been configured within the VRF are listed.

Note that VLAN 4093 is listed as an interface in VRF A, even though it was not explicitly configured. Also note that the SVI associated with the dynamic VLAN is automatically created behind the scenes. It is an unnumbered interface with no IP address, but can be seen through the use of the command 'show int vlan [vlan id]'.

Validate Dynamic VLAN, L3VNI and VRF Mappings

Standalone VTEP

```
A-LEAF1A#show interface vxlan1
Vxlan1 is up, line protocol is up (connected)
  Hardware is Vxlan
  Source interface is Loopback1 and is active with 2.2.2.1
  Replication/Flood Mode is headend with Flood List Source: EVPN
 Remote MAC learning via EVPN
  VNI mapping to VLANs
  Static VLAN to VNI mapping is
    [10, 10010]       [50, 10050]
  Dynamic VLAN to VNI mapping for 'evpn' is
    [4093, 50001]
  Note: All Dynamic VLANs used by VCS are internal VLANs.
        Use 'show vxlan vni' for details.
  Static VRF to VNI mapping is
   [A, 50001]
  Headend replication flood vtep list is:
    10 2.2.2.2
  MLAG Shared Router MAC is 0000.0000.0000
  VTEP address mask is None
```

MLAG VTEP

```
A-LEAF2A#show interface vxlan1
Vxlan1 is up, line protocol is up (connected)
  <...omittedd...>
  Source interface is Loopback1 and is active with 1.1.1.2
  <...omittedd...>
  MLAG Shared Router MAC is 0250.56c2.204a
  <...omittedd...>
```

The output above shows that VLAN 4093 is a Dynamic VLAN, which was created upon the association of VRF A to L3VNI 50001. When this association occurred, VLAN 4093 was automatically created and mapped to VNI 50001.

Notice in the MLAG output that the shared logical VTEP IP address, and EVPN Router MAC, are both listed for reference. This output should match on both MLAG peers.

Validate that Type-2 (MAC-IP) Routes now include IP addresses

```
A-LEAF1A#show bgp evpn route-type mac-ip 0050.5616.408b detail
BGP routing table information for VRF default
Router identifier 1.1.1.101, local AS number 65101
BGP routing table entry for mac-ip 10010 0050.5616.408b, Route Distinguisher:
1.1.1.101:1
 Paths: 1 available
  Local
    - from - (0.0.0.0)
      Origin IGP, metric -, localpref -, weight 0, valid, local, best
      Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
      VNI: 10010 ESI: 0000:0000:0000:0000:0000

BGP routing table entry for mac-ip 10010 0050.5616.408b 10.10.10.100, Route
Distinguisher: 1.1.1.101:1
 Paths: 1 available
  Local
    - from - (0.0.0.0)
      Origin IGP, metric -, localpref -, weight 0, valid, local, best
      Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
EvpnRouterMac:00:50:56:3a:b4:a1
      VNI: 10010 L3 VNI: 50001 ESI: 0000:0000:0000:0000:0000
```

The output above shows that A-LEAF1A is originating a Type-2 (MAC-IP) route for HostA, as expected and seen in the L2VPN section of this document. However, notice that there are now two distinct MAC-IP routes originated by A-LEAF1A for HostA.

The first MAC-IP route contains only Layer 2 information related to HostA, including:

- L2VNI
- Route-Target associated with the VLAN aware bundle that VLAN 10 is a member of
- Ethernet Tag ID associated with VNI
- MAC address of HostA

The second MAC-IP route contains both Layer 2 and Layer 3 information related to HostA. This update contains all of the information from the Layer 2 MAC-IP route, plus:

- L3VNI
- Route-Target associated with VRF A
- EVPN Router MAC
- IP Address of HostA

A MAC-IP route containing Layer 3 information is only originated for a host if the VTEP is providing L3VPN services for that tenant in which the host resides. In the scenario above, A-LEAF1A is providing L3VPN services for Tenant A (VRF A), which is the VRF that HostA resides in. Because of this, A-LEAF1A is originating a MAC-IP route for both L2VPN and L3VPN services related to HostA.

Validate that Type-5 (IP-Prefix) Routes are being Originated

```
A-LEAF1A#show bgp evpn route-type ip-prefix 10.10.10.0/24 detail
BGP routing table information for VRF default
Router identifier 1.1.1.101, local AS number 65101
BGP routing table entry for ip-prefix 10.10.10.0/24, Route Distinguisher: 1.1.1.101:1
 Paths: 1 available
  Local
    - from - (0.0.0.0)
      Origin IGP, metric -, localpref -, weight 0, valid, local, best
      Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
EvpnRouterMac:00:50:56:3a:b4:a1
      VNI: 50001
```

Once a VTEP is configured to provide L3VPN services, it will then be capable of originating Type-5 (IP-Prefix) routes within the tenant VRF.

Shown above, A-LEAF1A is originating reachability to the 10.10.10.0/24 prefix via an EVPN Type-5 route. This origination occurred because A-LEAF1A has been configured to redistribute locally connected prefixes in VRF A into BGP, and the 10.10.10.0/24 prefix is associated with local interface "VLAN 10".

Key information included within an EVPN Type-5 (IP-Prefix) route includes:

- IP Prefix
- Route-Target associated with the VRF from which the prefix is originated
- EVPN Router MAC
- L3VNI

Ensure that Routing Information is Properly Imported

As seen throughout this section, the EVPN control-plane can advertise and receive information related to L3VPN services. This information can be in the form of a traditional IP Prefix, via the Type-5 IP Prefix Route, or a /32 Host Route via the Type-2 MAC-IP Route. Regardless of whether the EVPN NLRI is advertising reachability to an IP-Prefix or a Host Route, these are both IPv4 Unicast address-family constructs that must exist in the RIB/FIB for this address-family to establish proper forwarding of packets destined to the IP Prefix, or Host Route.

In order for forwarding to occur, the information found within the EVPN NLRI of a Type-2 (MAC-IP) or Type-5 (IP-Prefix) route must be imported into the IPv4 Unicast address-family via the VRF associated with the Tenant these routes were originated in. When the commands 'route-target import evpn 1:1' and 'route-target export evpn 1:1' were configured under VRF A in BGP in the previous section, this enabled the automatic importing of the EVPN NLRI reachability information into the IPv4 Unicast address-family.

To illustrate this, we will look at reachability to HostC from the perspective of A-LEAF1A.

First, validate that A-LEAF1A has received the expected EVPN Type-2 (MAC-IP) route containing Layer 3 information for HostC.

```
A-LEAF1A#show bgp evpn route-type mac-ip 0050.5622.744a detail
<...Output Omitted...>
BGP routing table entry for mac-ip 10030 0050.5622.744a 30.30.30.100, Route
Distinguisher: 1.1.1.102:1
 Paths: 2 available
  65001 65102
    2.2.2.2 from 1.1.1.201 (1.1.1.201)
      Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP head, best,
ECMP contributor
      Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
EvpnRouterMac:02:50:56:c2:20:4a
      VNI: 10030 L3 VNI: 50001 ESI: 0000:0000:0000:0000:0000
<...Output Omitted...>
BGP routing table entry for mac-ip 10030 0050.5622.744a 30.30.30.100, Route
Distinguisher: 1.1.1.103:1
 Paths: 2 available
  65001 65102
    2.2.2.2 from 1.1.1.201 (1.1.1.201)
      Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP head, best,
ECMP contributor
      Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
EvpnRouterMac:02:50:56:c2:20:4a
      VNI: 10030 L3 VNI: 50001 ESI: 0000:0000:0000:0000:0000
<...Output Omitted...>
```

Next, ensure that this information has been properly imported into VRF A via the IPv4 Unicast address-family within BGP.

```
A-LEAF1A#show ip bgp 30.30.30.100/32 detail vrf A
BGP routing table information for VRF A
<...Output Omitted...>
BGP routing table entry for 30.30.30.100/32
 Paths: 4 available
  65001 65102
    2.2.2.2 from 1.1.1.201 (1.1.1.201), imported EVPN route, RD 1.1.1.103:1
      Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP head, best,
ECMP contributor
      Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
EvpnRouterMac:02:50:56:c2:20:4a
      Remote VNI: 50001
  65001 65102
    2.2.2.2 from 1.1.1.201 (1.1.1.201), imported EVPN route, RD 1.1.1.102:1
      Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP, ECMP
contributor
      Not best: ECMP-Fast configured
      Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
EvpnRouterMac:02:50:56:c2:20:4a
      Remote VNI: 50001
<...Output Omitted...>
```

Note that the above command is looking at the BGP table for the IPv4 Unicast address-family, and not EVPN. Additionally, the output of the NLRI associated with the host route specifically calls out that this information was imported into the IPv4 Unicast BGP table via EVPN. The extended communities associated with the EVPN NLRI are imported as well, but these can be safely removed when advertising this host route to IPv4 Unicast BGP peers that are not VTEPs.

Lastly, ensure that the prefix is properly imported from the VRF A IPv4 Unicast BGP table into the VRF A IPv4 Unicast Routing Table (RIB).

```
VRF: A
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route, L - VRF Leaked

 B E     30.30.30.100/32 [200/0] via VTEP 2.2.2.2 VNI 50001 router-mac
02:50:56:c2:20:4a
```

Following a similar process to that above, validate that A-LEAF1A has received the expected EVPN Type-5 (IP-Prefix) route containing 30.30.30.0/24 prefix.

First, validate that the expected EVPN Route-Type has been received for this prefix.

```
A-LEAF1A#show bgp evpn route-type ip-prefix 30.30.30.0/24 detail
<...Output Omitted...>
BGP routing table entry for ip-prefix 30.30.30.0/24, Route Distinguisher: 1.1.1.102:1
 Paths: 2 available
  65001 65102
    2.2.2.2 from 1.1.1.201 (1.1.1.201)
      Origin IGP, metric -, localpref 100, weight 0, valid, external, best
      Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
EvpnRouterMac:02:50:56:c2:20:4a
      VNI: 50001
<...Output Omitted...>
BGP routing table entry for ip-prefix 30.30.30.0/24, Route Distinguisher: 1.1.1.103:1
 Paths: 2 available
  65001 65102
    2.2.2.2 from 1.1.1.201 (1.1.1.201)
      Origin IGP, metric -, localpref 100, weight 0, valid, external, best
      Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
EvpnRouterMac:02:50:56:c2:20:4a
      VNI: 50001
<...Output Omitted...>
```

Next, validate that the prefix exists within IPv4 Unicast BGP table for VRF A.

```
A-LEAF1A#show ip bgp 30.30.30.0/24 detail vrf A
BGP routing table information for VRF A
<...Output Omitted...>
BGP routing table entry for 30.30.30.0/24
 Paths: 4 available
  65001 65102
    2.2.2.2 from 1.1.1.201 (1.1.1.201), imported EVPN route, RD 1.1.1.103:1
      Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP head, best,
ECMP contributor
      Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
EvpnRouterMac:02:50:56:c2:20:4a
      Remote VNI: 50001
<...Output Omitted...>
  65001 65102
    2.2.2.2 from 1.1.1.201 (1.1.1.201), imported EVPN route, RD 1.1.1.102:1
      Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP, ECMP
contributor
      Not best: ECMP-Fast configured
      Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
EvpnRouterMac:02:50:56:c2:20:4a
      Remote VNI: 50001
<...Output Omitted...>
```

Next, validate that the prefix exists within IPv4 Unicast BGP table for VRF A.

```
A-LEAF1A#show ip route vrf A 30.30.30.100

VRF: A
Codes: C - connected, S - static, K - kernel,
       O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
       E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
       N2 - OSPF NSSA external type2, B I - iBGP, B E - eBGP,
       R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
       O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
       NG - Nexthop Group Static Route, V - VXLAN Control Service,
       DH - DHCP client installed default route, M - Martian,
       DP - Dynamic Policy Route, L - VRF Leaked

 B E     30.30.30.100/32 [200/0] via VTEP 2.2.2.2 VNI 50001 router-mac
02:50:56:c2:20:4a
```

With all of this information populated, reachability to 30.30.30.100/32, as well as other addresses within the 30.30.30.0/24 subnet, should be in place.

Appendix A: Topology IP Addressing

Loopback0 (BGP EVPN Peering):

Data Center A	
A-SPINE1	1.1.1.201
A-SPINE2	1.1.1.202
A-LEAF1A	1.1.1.101
A-LEAF2A	1.1.1.102
A-LEAF2B	1.1.1.103
A-SVC1A	1.1.1.104
A-SVC1B	1.1.1.105
A-BL1A	1.1.1.106
A-BL1B	1.1.1.107

Loopback1 (VXLAN Tunnel Source):

Data Center A	
A-LEAF1A	2.2.2.1
A-LEAF2A	2.2.2.2
A-LEAF2B	2.2.2.2
A-SVC1A	2.2.2.3
A-SVC1B	2.2.2.3
A-BL1A	2.2.2.4
A-BL1B	2.2.2.4

Point-to-Point Connections:

Data Center A					
Node 1	**Interface**	**IP Address**	**Node 2**	**Interface**	**IP Address**
A-SPINE1	Eth1	10.101.201.201/24	A-LEAF1A	Eth1	10.101.201.101/24
A-SPINE1	Eth2	10.102.201.201/24	A-LEAF2A	Eth1	10.102.201.102/24
A-SPINE1	Eth3	10.103.201.201/24	A-LEAF2B	Eth1	10.103.201.103/24
A-SPINE1	Eth4	10.104.201.201/24	A-SVC1A	Eth1	10.104.201.104/24
A-SPINE1	Eth5	10.105.201.201/24	A-SVC1B	Eth1	10.105.201.105/24
A-SPINE1	Eth6	10.106.201.201/24	A-BL1A	Eth1	10.106.201.106/24
A-SPINE1	Eth7	10.107.201.201/24	A-BL1B	Eth1	10.107.201.107/24
A-SPINE2	Eth1	10.101.202.202/24	A-LEAF1A	Eth1	10.101.202.101/24
A-SPINE2	Eth2	10.102.202.202/24	A-LEAF2A	Eth1	10.102.202.102/24
A-SPINE2	Eth3	10.103.202.202/24	A-LEAF2B	Eth1	10.103.202.103/24
A-SPINE2	Eth4	10.104.202.202/24	A-SVC1A	Eth1	10.104.202.104/24
A-SPINE2	Eth5	10.105.202.202/24	A-SVC1B	Eth1	10.105.202.105/24
A-SPINE2	Eth6	10.106.202.202/24	A-BL1A	Eth1	10.106.202.106/24
A-SPINE2	Eth7	10.107.202.202/24	A-BL1B	Eth1	10.107.202.107/24

MLAG Pair Peering (Consistent on all Pairs):

Node 1	Interface	IP Address	Node 2	Interface	IP Address
MLAG Peer 1	Vlan4094	10.0.0.1/30	MLAG Peer 2	Vlan4094	10.0.0.2/30

MLAG iBGP Peering (Consistent on all Pairs):

Node 1	Interface	IP Address	Node 2	Interface	IP Address
MLAG Peer 1	Vlan4093	192.0.0.1/24	MLAG Peer 2	Vlan4094	192.0.0.2/24

End Hosts:

Data Center A	
HostA	10.10.10.100/24
HostB	10.10.10.200/24
HostC	30.30.30.100/24
HostD	50.50.50.100/24

Appendix B: vEOS-LAB known Caveats and Recommendations

vEOS-LAB is a great tool for network engineers to learn Arista EOS, and to validate and test their designs in a virtual environment. However, due to the nature of it's virtual data plane, there are minor caveats to be aware of in order to conduct successful validation and testing of this EVPN design guide.

- **VM resources:**

  For best performance, it is recommended that each vEOS-LAB instance be assigned a minimum of 2 vCPUs and 4096 MB of memory. Depending on the size of the topology and routing table, you may need to increase these values accordingly. CPU and Memory utilization can be monitored using the following command: show processes top

  Note: In vEOS-Lab, the Etba process relates to virtual driver/hardware simulating underlying network processor silicon, so its high CPU utilization indicates forwarding activity or learning/writing communication within the control-plane.

- **MTU Size:**

  The MTU size should not exceed 1500, as this is a limitation of the virtual data plane. Point-to-Point links should be left to their default of 1500 MTU. Unless one is transmitting large packets on the virtual fabric, EVPN and VXLAN functionality can be successfully tested with this default value.

- **Bidirectional Forwarding Detection (BFD):**

  BFD is enabled for fast detection of a transport failure between EVPN peers. It has been observed that if the default values are not modified, one may experience excess CPU utilization, which could result in an unstable fabric.

  The default BFD values (interval 300 min_rx 300 multiplier 3) should be changed to increase the intervals and min_rx values to a minimum of 1200 and 1200 respectively. To do so, the following command can be applied at the global config level:

  bfd multihop interval 1200 min_rx 1200 multiplier 3

  If high CPU utilization is still observed, one can try increasing these values, or simply disable fall-over bfd under the BGP EVPN peer group. This would only impact convergence times when simulating failure scenarios.

Appendix C: References

L2/ L3 EVPN using VXLAN

EOS-4.18.1F EVPN extension to BGP using VXLAN

EOS-4.20.1F EVPN Integrated Routing and Bridging with VXLAN

EOS-4.20.6F Centralized Anycast Gateway for EVPN

EOS-4.21.3F EVPN MLAG shared router MAC

EOS-4.22.0F EVPN VXLAN IPv6 overlay routing

EOS-4.22.0F EVPN VXLAN multihoming Type4 routes

EOS-4.18.1F L3 EVPN extension to BGP using MPLS

L2 EVPN using MPLS

EOS-4.20.5F L2 EVPN MPLS

EOS-4.20.5F L2 EVPN MPLS on 7500R, 7500R2, 7280R & 7280R2

EOS-4.21.3F Configurable L2 EVPN MPLS control word

EOS-4.22.0F VLAN aware bundle

EOS-4.21.3F EVPN IRB with MPLS underlay

Related features and applications

EOS-4.20.5F DHCP relay in VXLAN EVPN

EOS-4.20.5F Multicast in VXLAN VLAN using underlay

EOS-4.20.5F VXLAN nexthop for PBR

EOS-4.21.3F VXLAN Static and EVPN - Dual configuration

EOS-4.21.3F Federating CVX across DCI using EVPN

EOS-4.21.5F Propagating ECN header on VXLAN encap/ decap

EOS-4.22.0F VNI based rate limiting (QoS policer)

EOS-4.22.0F Inter VRF route leaking

RFCs:

RFC 7432 BGP MPLS Based Ethernet VPN

RFC 7348 Virtual eXtensible Local Area Network (VXLAN)

Appendix D: Final Configurations

All final configurations associated with this guide can be found in the "EVPN Deployment Guide" folder within the following GitHub Repository:

https://github.com/aristanetworks/eos-deployment-guide-configs

Santa Clara—Corporate Headquarters
5453 Great America Parkway,
Santa Clara, CA 95054

Phone: +1-408-547-5500
Fax: +1-408-538-8920
Email: info@arista.com

Ireland—International Headquarters
3130 Atlantic Avenue
Westpark Business Campus
Shannon, Co. Clare
Ireland

Vancouver—R&D Office
9200 Glenlyon Pkwy, Unit 300
Burnaby, British Columbia
Canada V5J 5J8

San Francisco—R&D and Sales Office 1390
Market Street, Suite 800
San Francisco, CA 94102

India—R&D Office
Global Tech Park, Tower A & B, 11th Floor
Marathahalli Outer Ring Road
Devarabeesanahalli Village, Varthur Hobli
Bangalore, India 560103

Singapore—APAC Administrative Office
9 Temasek Boulevard
#29-01, Suntec Tower Two
Singapore 038989

Nashua—R&D Office
10 Tara Boulevard
Nashua, NH 03062