L2 EVPN-MH-VXLAN with SR Linux

Prepared by: Hasan-Uz-Zaman Ashik
Reference: https://learn.srlinux.dev/tutorials/l2evpn/summary/

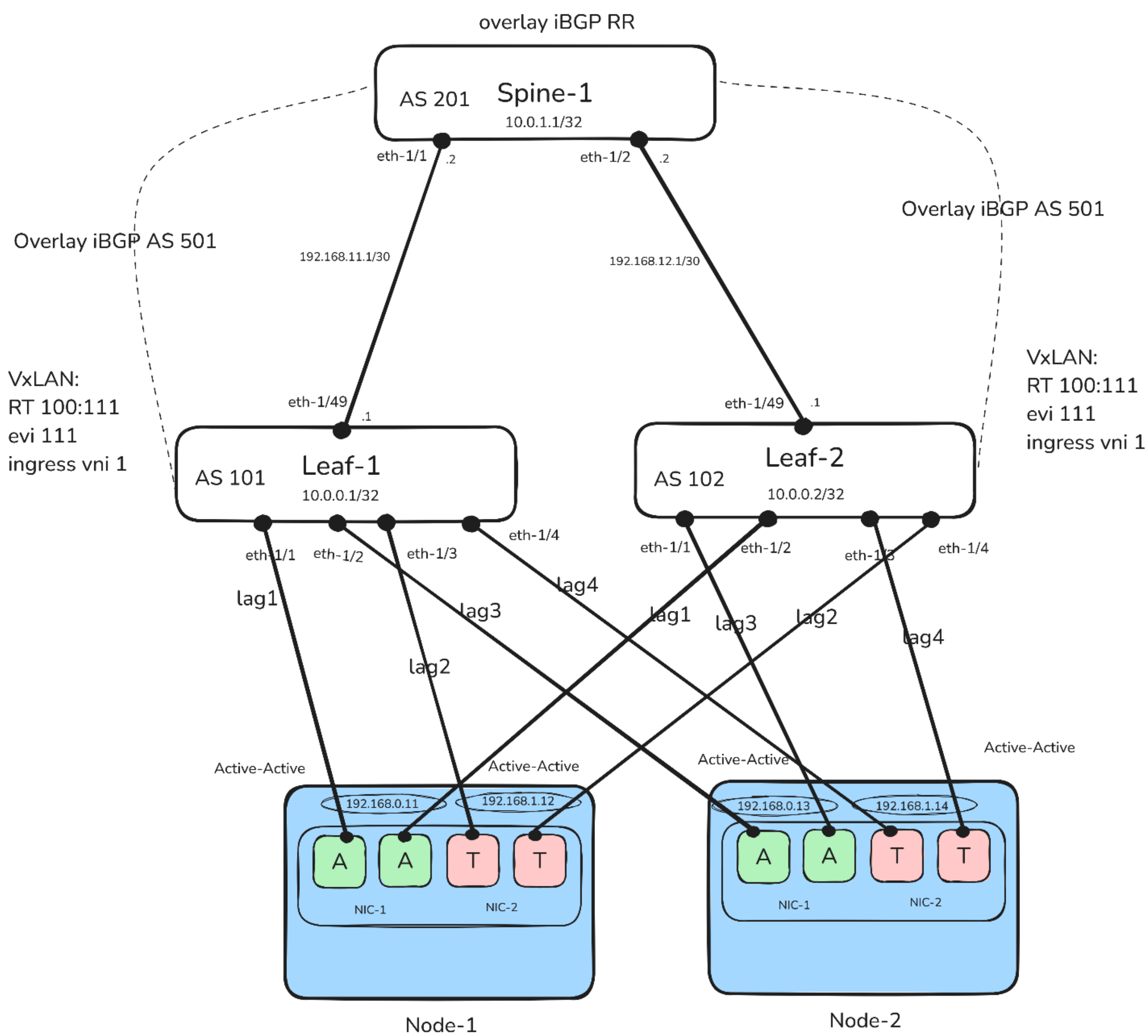
Contents

Ethernet Virtual Private Network (EVPN) is a standard technology in multi-tenant Data Centers (DCs) and provides a control plane framework for many functions.

Objective:

1. Configure a VxLAN based Layer 2 EVPN-MH service

DC fabric: Two leaf switches (acting as Top-Of-Rack) and a single spine



<u>Major parts</u>:

Fabric configuration: Here we will configure the eBGP routing protocol in the underlay of a fabric to advertise the Virtual Tunnel Endpoints (VTEP) of the leaf switches.

- Leaf-Spine interfaces configure
- Assign network-instance default to interfaces for basic IP connectivity
- Configure eBGP peering between the leaf-spine pairs (eBGP-underlay ipv4-unicast)

EVPN-MH configuration:

- iBGP session is established between leafs & spine - acting as RR (iBGP-Overlay evpn)
- Server facing interfaces configure (Access, Trunk, Lag, LACP)
- Ethernet segment configure for MH.
- Load-balancing for all-active multihoming segments - ecmp
- VXLAN tunnel interface configuration under tunnel interface (ingress VNI)
- Access, trunk and vxlan interfaces with the mac-vrf
- EVPN in MAC-VRF (bgp-vpn, bgp-evpn, EVPN Virtual Identifier (EVI))

Node configuration:

- LACP aware 2 bond interfaces (Access, Trunk) configure with 4 physical links with leaf.

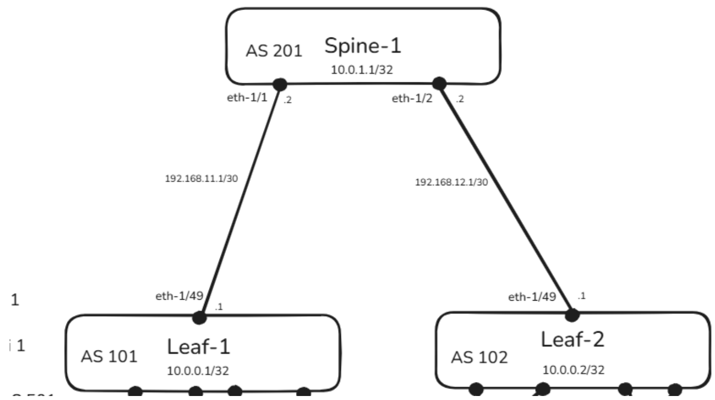
Let's see in detail now…

Fabric configuration

Prior to configuring EVPN based overlay, a routing protocol needs to be deployed in the fabric to advertise the reachability of all the leaf VxLAN Termination End Point (VTEP) addresses throughout the IP fabric. We will use a eBGP based fabric design as described in RFC7938 due to its simplicity, scalability, and ease of multi-vendor interoperability.

Leaf-Spine interfaces configure

On each leaf and spine we will bring up the relevant interface and address its routed subinterface to achieve L3 connectivity.



Assign network-instance default to interfaces for basic IP connectivity

Leaf1:

```
A:leaf1#  info interface ethernet-1/49
    interface ethernet-1/49 {
        admin-state enable
        subinterface 0 {
            ipv4 {
                admin-state enable
                address 192.168.11.1/30 {
                }
            }
        }
    }
--{ running }--[  ]--
A:leaf1#  info interface system0
    interface system0 {
        admin-state enable
        subinterface 0 {
            ipv4 {
                admin-state enable
                address 10.0.0.1/32 {
                }
            }
        }
    }
--{ running }--[  ]--
```

```
A:leaf1# info network-instance default
    network-instance default {
        interface ethernet-1/49.0 {
        }
        interface system0.0 {
        }
```

Spine1:

```
A:spine1# info interface ethernet-1/1
    interface ethernet-1/1 {
        admin-state enable
        subinterface 0 {
            ipv4 {
                admin-state enable
                address 192.168.11.2/30 {
                }
            }
        }
    }
--{ running }--[  ]--
A:spine1# info interface ethernet-1/2
    interface ethernet-1/2 {
        admin-state enable
        subinterface 0 {
            ipv4 {
                admin-state enable
                address 192.168.12.2/30 {
                }
            }
        }
    }
--{ running }--[  ]--
A:spine1# info interface system0
    interface system0 {
        admin-state enable
        subinterface 0 {
            ipv4 {
                admin-state enable
                address 10.0.1.1/32 {
                }
            }
        }
    }
--{ running }--[  ]--
```

```
A:spine1# info network-instance default
    network-instance default {
        interface ethernet-1/1.0 {
        }
        interface ethernet-1/2.0 {
        }
        interface system0.0 {
        }
        protocols {
            bgp {
                autonomous-system 201
                router-id 10.0.1.1
                afi-safi ipv4-unicast {
                    admin-state enable
                }
                group eBGP-underlay {
                    export-policy all
                    import-policy all
                }
                group evpn-rr {
                    admin-state enable
                    export-policy all
                    import-policy all
                    peer-as 501
                    afi-safi evpn {
                        admin-state enable
                    }
                    afi-safi ipv4-unicast {
                        admin-state disable
                    }
                    local-as {
                        as-number 501
                    }
                    route-reflector {
                        client true
                    }
                    timers {
                        minimum-advertisement-interval 1
                    }
                }
```

```
                neighbor 10.0.0.1 {
                    peer-group evpn-rr
                    transport {
                        local-address 10.0.1.1
                    }
                }
                neighbor 10.0.0.2 {
                    peer-group evpn-rr
                    transport {
                        local-address 10.0.1.1
                    }
                }
                neighbor 192.168.11.1 {
                    peer-as 101
                    peer-group eBGP-underlay
                }
                neighbor 192.168.12.1 {
                    peer-as 102
                    peer-group eBGP-underlay
                }
            }
        }
    }
--{ candidate shared default }--[  ]--
A:spine1#
```

Leaf2:

```
A:leaf2# info interface ethernet-1/49
    interface ethernet-1/49 {
        admin-state enable
        subinterface 0 {
            ipv4 {
                admin-state enable
                address 192.168.12.1/30 {
                }
            }
        }
    }
--{ running }--[  ]--
A:leaf2# info interface system0
    interface system0 {
        admin-state enable
        subinterface 0 {
            ipv4 {
                admin-state enable
                address 10.0.0.2/32 {
                }
            }
        }
    }
--{ running }--[  ]--
```

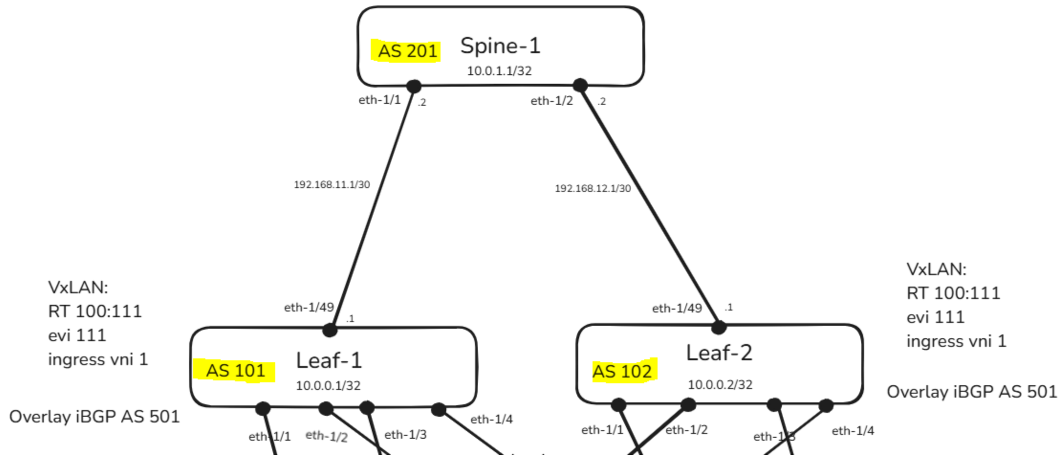
```
A:leaf2# info network-instance default
    network-instance default {
        interface ethernet-1/49.0 {
        }
        interface system0.0 {
        }
```

When interfaces are owned by the network-instance default, we can ensure that the basic IP connectivity is working by issuing a ping between the pair of interfaces.

```
A:leaf1#  ping 192.168.12.1 network-instance default
Using network instance default
PING 192.168.12.1 (192.168.12.1) 56(84) bytes of data.
64 bytes from 192.168.12.1: icmp_seq=1 ttl=63 time=13.3 ms
64 bytes from 192.168.12.1: icmp_seq=2 ttl=63 time=8.69 ms
^C
--- 192.168.12.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 8.689/11.001/13.314/2.314 ms
--{ running }--[  ]--
```

Configure eBGP peering between the leaf-spine pairs (eBGP-underlay ipv4-unicast)

The EBGP will make sure of advertising the VTEP IP addresses (loopbacks) across the fabric.



We will use AS 101 leaf1, AS 102 leaf2 and AS 201 Spine1 for the eBGP peering.

```
A:leaf1# info network-instance default protocols bgp
    network-instance default {
        protocols {
            bgp {
                autonomous-system 101
                router-id 10.0.0.1
                afi-safi ipv4-unicast {
                    admin-state enable
                }
```

```
                group eBGP-underlay {
                    export-policy all
                    import-policy all
                    peer-as 201
                }
```

```
                neighbor 192.168.11.2 {
                    peer-group eBGP-underlay
                }
            }
        }
    }
```

Now create export/import policies. The export/import policy is required for an eBGP peer to advertise and install routes.

The policy named “all” that we create below will be used both as an import and export policy, effectively allowing all routes to be advertised and received.

```
A:leaf1# info / routing-policy
    routing-policy {
        policy all {
            default-action {
                policy-result accept
            }
        }
    }
--{ running }--[  ]--
```

Create peer-group config: A peer group should include sessions that have a similar or almost identical configuration. Here, the peer group is named eBGP-underlay since it will be used to enable underlay routing between the leafs and spines.

```
A:leaf1# info network-instance default protocols bgp group eBGP-underlay
    network-instance default {
        protocols {
            bgp {
                group eBGP-underlay {
                    export-policy all
                    import-policy all
                    peer-as 201
                }
            }
        }
    }
--{ running }--[  ]--
```

Configure neighbor:

```
A:leaf1# info network-instance default protocols bgp neighbor 192.168.11.2
    network-instance default {
        protocols {
            bgp {
                neighbor 192.168.11.2 {
                    peer-group eBGP-underlay
                }
            }
        }
    }
--{ running }--[  ]--
```

Loopbacks:

As we will create a iBGP based EVPN control plane at a later stage, we need to configure loopback addresses for our leaf devices so that they can build an iBGP peering over those interfaces. In the context of the VXLAN data plane, a special kind of a loopback needs to be created - system0 interface.

The system0.0 interface hosts the loopback address used to originate and typically terminate VXLAN packets. This address is also used by default as the next-hop of all EVPN routes.

```
A:leaf1# info interface system0
    interface system0 {
        admin-state enable
        subinterface 0 {
            ipv4 {
                admin-state enable
                address 10.0.0.1/32 {
                }
            }
        }
    }
--{ running }--[  ]--
A:leaf1# info network-instance default interface system0.0
    network-instance default {
        interface system0.0 {
        }
    }
--{ running }--[  ]--
```

Verification:

VXLAN VTEPs need to be advertised throughout the DC fabric. The system0 interfaces we just configured are the VTEPs and they should be advertised via EBGP peering established before.

a) eBGP status

```
A:leaf1# show network-instance default protocols bgp summary
-------------------------------------------------------------
BGP is enabled and up in network-instance "default"
Global AS number  : 101
BGP identifier    : 10.0.0.1
-------------------------------------------------------------
  Total paths               : 13
  Received routes           : 4294967269
  Received and active routes: 16
  Total UP peers            : 2
  Configured peers          : 2, 0 are disabled
  Dynamic peers             : None
-------------------------------------------------------------
Default preferences
  BGP Local Preference attribute: 100
  EBGP route-table preference   : 170
  IBGP route-table preference   : 170
-------------------------------------------------------------
Wait for FIB install to advertise: True
Send rapid withdrawals           : disabled
-------------------------------------------------------------
Ipv4-unicast AFI/SAFI
    Received routes               : 4
    Received and active routes    : 3
    Max number of multipaths      : 1, 1
    Multipath can transit multi AS: True
-------------------------------------------------------------
Ipv6-unicast AFI/SAFI
    Received routes               : 0
    Received and active routes    : 0
    Max number of multipaths      : None,None
    Multipath can transit multi AS: None
-------------------------------------------------------------
EVPN-unicast AFI/SAFI
    Received routes               : 0
    Received and active routes    : 0
    Max number of multipaths      : N/A
    Multipath can transit multi AS: N/A
-------------------------------------------------------------
--{ running }--[  ]--
```

b) BGP neighbor status

```
A:leaf1# show network-instance default protocols bgp neighbor
-----------------------------------------------------------------------------------------------------------------------------------------------
BGP neighbor summary for network-instance "default"
Flags: S static, D dynamic, L discovered by LLDP, B BFD enabled, - disabled, * slow
-----------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------
+---------------------+-------------------------------+---------------------+--------+-----------+-----------------+-----------------+---------------+-----------------------------+
|      Net-Inst       |             Peer              |        Group        | Flags  |  Peer-AS  |      State      |     Uptime      |   AFI/SAFI    |       [Rx/Active/Tx]        |
+=====================+===============================+=====================+========+===========+=================+=================+===============+=============================+
| default             | 10.0.0.2                      | eBGP-overlay        | S      | 501       | established     | 0d:14h:45m:16s  | evpn          | [13/13/15]                  |
| default             | 192.168.11.2                  | eBGP-underlay       | S      | 201       | established     | 1d:3h:39m:53s   | ipv4-unicast  | [4/3/2]                     |
+---------------------+-------------------------------+---------------------+--------+-----------+-----------------+-----------------+---------------+-----------------------------+
-----------------------------------------------------------------------------------------------------------------------------------------------
Summary:
2 configured neighbors, 2 configured sessions are established,0 disabled peers
0 dynamic peers
--{ running }--[  ]--
```

c) Received/Advertised routes

```
A:leaf1# show network-instance default protocols bgp neighbor 192.168.11.2 advertised-routes ipv4
-----------------------------------------------------------------------------------------------------
Peer        : 192.168.11.2, remote AS: 201, local AS: 101
Type        : static
Description : None
Group       : eBGP-underlay
-----------------------------------------------------------------------------------------------------
Origin codes: i=IGP, e=EGP, ?=incomplete
+---------------------------------------------------------------------------------------------------+
|      Network          Path-id       Next Hop       MED        LocPref        AsPath       Origin  |
+===================================================================================================+
| 10.0.0.1/32           0             192.168.11.1   -          100            [101]        i       |
| 192.168.11.0/30       0             192.168.11.1   -          100            [101]        i       |
+---------------------------------------------------------------------------------------------------+
-----------------------------------------------------------------------------------------------------
2 advertised BGP routes
-----------------------------------------------------------------------------------------------------
--{ running }--[  ]--
A:leaf1# show network-instance default protocols bgp neighbor 192.168.11.2 received-routes ipv4
-----------------------------------------------------------------------------------------------------
Peer        : 192.168.11.2, remote AS: 201, local AS: 101
Type        : static
Description : None
Group       : eBGP-underlay
-----------------------------------------------------------------------------------------------------
Status codes: u=used, *=valid, >=best, x=stale
Origin codes: i=IGP, e=EGP, ?=incomplete
+---------------------------------------------------------------------------------------------------+
|  Status     Network          Path-id     Next Hop       MED     LocPref     AsPath       Origin   |
+===================================================================================================+
|   u*>       10.0.0.2/32      0           192.168.11.2   -       100         [201, 102]   i        |
|   u*>       10.0.1.1/32      0           192.168.11.2   -       100         [201]        i        |
|    *        192.168.11.0/30  0           192.168.11.2   -       100         [201]        i        |
|   u*>       192.168.12.0/30  0           192.168.11.2   -       100         [201]        i        |
+---------------------------------------------------------------------------------------------------+
-----------------------------------------------------------------------------------------------------
4 received BGP routes : 3 used 4 valid
-----------------------------------------------------------------------------------------------------
--{ running }--[  ]--
```

d) Route table:

```
A:leaf1# show network-instance default route-table ipv4-unicast summary
-----------------------------------------------------------------------------------------------------------------------------------
IPv4 unicast route table of network instance default
-----------------------------------------------------------------------------------------------------------------------------------
+-----------------+----+------------+--------------+--------+----------+--------+------+--------------------+--------------------+
|     Prefix      | ID | Route Type | Route Owner  | Active | Origin   | Metric | Pref | Next-hop (Type)    | Next-hop Interface |
|                 |    |            |              |        | Network  |        |      |                    |                    |
|                 |    |            |              |        | Instance |        |      |                    |                    |
+=================+====+============+==============+========+==========+========+======+====================+====================+
| 10.0.0.1/32     | 4  | host       | net_inst_mgr | True   | default  | 0      | 0    | None (extract)     | None               |
| 10.0.0.2/32     | 0  | bgp        | bgp_mgr      | True   | default  | 0      | 170  | 192.168.11.0/30    | ethernet-1/49.0    |
|                 |    |            |              |        |          |        |      | (indirect/local)   |                    |
| 10.0.1.1/32     | 0  | bgp        | bgp_mgr      | True   | default  | 0      | 170  | 192.168.11.0/30    | ethernet-1/49.0    |
|                 |    |            |              |        |          |        |      | (indirect/local)   |                    |
| 192.168.11.0/30 | 3  | local      | net_inst_mgr | True   | default  | 0      | 0    | 192.168.11.1       | ethernet-1/49.0    |
|                 |    |            |              |        |          |        |      | (direct)           |                    |
| 192.168.11.1/32 | 3  | host       | net_inst_mgr | True   | default  | 0      | 0    | None (extract)     | None               |
| 192.168.11.3/32 | 3  | host       | net_inst_mgr | True   | default  | 0      | 0    | None (broadcast)   |                    |
| 192.168.12.0/30 | 0  | bgp        | bgp_mgr      | True   | default  | 0      | 170  | 192.168.11.0/30    | ethernet-1/49.0    |
|                 |    |            |              |        |          |        |      | (indirect/local)   |                    |
+-----------------+----+------------+--------------+--------+----------+--------+------+--------------------+--------------------+
-----------------------------------------------------------------------------------------------------------------------------------
IPv4 routes total                    : 7
IPv4 prefixes with active routes     : 7
IPv4 prefixes with active ECMP routes: 0
-----------------------------------------------------------------------------------------------------------------------------------
--{ running }--[  ]--
```

Both leaf2 and spine1 prefixes are found in the route table of network-instance default and the bgp_mgr is the owner of those prefixes, which means that they have been added to the route-table by the BGP app.

e) Dataplane ping

```
A:leaf1# ping -I 10.0.0.1 network-instance default 10.0.0.2
Using network instance default
PING 10.0.0.2 (10.0.0.2) from 10.0.0.1 : 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=63 time=14.4 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=63 time=13.0 ms
^C
--- 10.0.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 13.030/13.704/14.378/0.674 ms
--{ running }--[  ]--
```

So VTEPs or system ips are reachable and the fabric underlay is properly configured.

EVPN

Background

Ethernet Virtual Private Network (EVPN), along with Virtual eXtensible LAN (VXLAN), is a technology that allows Layer 2 and Layer 3 traffic to be tunneled across an IP network.

The SR Linux EVPN-VXLAN solution enables Layer 2 Broadcast Domains (BDs) in multi-tenant data centers using EVPN for the control plane and VXLAN as the data plane. It includes the following features:

- EVPN for VXLAN tunnels (Layer 2), extending a BD in overlay multi-tenant DCs
- EVPN for VXLAN tunnels (Layer 3), allowing inter-subnet-forwarding for unicast traffic within the same tenant infrastructure

EVPN-VXLAN provides Layer-2 connectivity in multi-tenant DCs. EVPN-VXLAN Broadcast Domains (BD) can span several leaf routers connected to the same IP fabric, allowing hosts attached to the same BD to communicate as though they were connected to the same layer-2 switch.

VXLAN tunnels bridge the layer-2 frames between leaf routers with EVPN providing the control plane to automatically setup tunnels and use them efficiently.

What is Underlay and Overlay Network?

Underlay Network is physical infrastructure above which overlay network is built. It is the underlying network responsible for delivery of packets across networks. Underlay networks can be Layer 2 or Layer 3 networks. Layer 2 underlay networks today are typically based on Ethernet, with segmentation accomplished via VLANs. The Internet is an example of a Layer 3 underlay network.

An Overlay Network is a virtual network that is built on top of underlying network infrastructure (Underlay Network). Actually, “Underlay” provides a “service” to the overlay. Overlay networks implement network virtualization concepts. A virtualized network consists of overlay nodes (e.g., routers), where Layer 2 and Layer 3 tunneling encapsulation (VXLAN, GRE, and IPSec) serves as the transport overlay protocol.

What is VxLAN?

VxLAN — or Virtual Extensible LAN addresses the requirements of the Layer 2 and Layer 3 data center network infrastructure in the presence of VMs in a multi-tenant environment. It runs over the existing networking infrastructure and provides a means to "stretch" a Layer 2 network. In short, VXLAN is a Layer 2 overlay scheme on a Layer 3 network. Each overlay is termed a VXLAN segment. Only VMs within the same VXLAN segment can communicate with each other. Each VXLAN segment is identified through a 24-bit segment ID, termed the "VNI". This allows up to 16 M VXLAN segments to coexist within the same administrative domain.

What is EVI in EVPN?

EVI: Represents a logical EVPN instance, managing control-plane operations for a VPN.

The EVPN Virtual Identifier (EVI) is a numerical identifier that defines a specific EVPN instance. It is used to distinguish different EVPN instances within a single device or network. Each EVI corresponds to a distinct Layer 2 or Layer 3 VPN instance in the EVPN control plane.

What is VNI?

Unlike VLAN, VxLAN does not have ID limitation. It uses a 24-bit header, which gives us about 16 million VNI's to use. A VNI VXLAN Network Identifier (VNI) is the identifier for the LAN segment, similar to a VLAN ID. With an address space this large, an ID can be assigned to a customer, and it can remain unique across the entire network.

In a VXLAN network, the VXLAN Network Identifier (VNI) is embedded within the VXLAN header, which is encapsulated in a UDP packet as part of the overall VXLAN tunnel.

When a frame is sent over a VXLAN tunnel:

- Original Ethernet Frame: The frame to be transported is encapsulated.
- VXLAN Header: A VXLAN header, which includes the VNI, is added to the frame.
- UDP Header: The VXLAN packet is wrapped in a UDP header.
- Outer IP Header: The UDP packet is encapsulated in an outer IP header for transport across the Layer 3 network.

Position of VNI

- The VNI is a **24-bit field** in the **VXLAN header**.
- The VXLAN header is **8 bytes or 64 bits** long and resides between the UDP header and the encapsulated Ethernet frame.

Here's a breakdown of the VXLAN header format:

|R|R|R|R|I|R|R|R| Reserved | VNI | R | = 8+24+24+8 = 64 bit

Key Fields:

1. **Flags (8 bits)**:
   - The "I" flag (bit 3) indicates that the VNI field is valid.
2. **Reserved (24 bits)**:
   - These bits are reserved for future use and must be set to 0.
3. **VXLAN Network Identifier (VNI)**:
   - A 24-bit field uniquely identifying the VXLAN segment.
4. **Reserved (8 bits)**:
   - Reserved for future use and must be set to 0.

How VNI is Attached in the Tunnel IP Header

The VNI is **not directly part of the outer IP header**. Instead:

1. **Outer IP Header**: Contains the source and destination IP addresses of the VXLAN Tunnel Endpoints (VTEPs).

- **Source IP**: The IP address of the sending VTEP.
- **Destination IP**: The IP address of the receiving VTEP.

2. **UDP Header**:
    - **Source Port**: Typically derived using a hash of the inner packet's headers to allow for load balancing.
    - **Destination Port**: The well-known port for VXLAN (4789).
3. **VXLAN Header**:
    - Encodes the VNI, enabling the receiving VTEP to identify the VXLAN segment to which the encapsulated traffic belongs.

Encapsulation Packet Structure:

| Outer Ethernet Header | Outer IP Header | UDP Header | VXLAN Header | Inner Ethernet Frame |

Outer Ethernet Header: Contains MAC addresses of the transport network.

Outer IP Header: Contains source and destination IP addresses of the VTEPs.

UDP Header:

- Source Port: Random or hashed value for load balancing.
- Destination Port: 4789 (VXLAN standard port).

VXLAN Header:

- Includes the VNI.

Inner Ethernet Frame:

- Original Ethernet frame (with its own MAC addresses).

Role of the VNI in the Tunnel

- At the **ingress VTEP**, the VNI is assigned to traffic based on the VLAN or Layer 3 VRF configuration.
- At the **egress VTEP**, the receiving device uses the VNI from the VXLAN header to determine the appropriate Layer 2 or Layer 3 segment for decapsulation.

EVI vs. VNI

- EVI operates at the EVPN layer (control plane) and represents the service instance for a Layer 2 or Layer 3 VPN.
- VNI operates at the VXLAN data plane and maps to a specific Layer 2 broadcast domain (or Layer 3 instance in EVPN).

Relationship:

- Each EVPN instance (EVI) can have one or more associated VNIs.
- EVI organizes and controls the services and routes at the control plane, while VNI carries encapsulated traffic over the VXLAN data plane.

What is VTEP?

VxLAN traffic is encapsulated before it is sent over the network. This creates stateless tunnels across the network, from the source switch to the destination switch. The encapsulation and decapsulation are handled by a component called a VTEP (VxLAN Tunnel End Point). A VTEP has an IP address in the underlay network. It also has one or more VNI's associated with it. When frames from one of these VNI's arrives at the Ingress VTEP, the VTEP encapsulates it with UDP and IP headers. The encapsulated packet is sent over the IP network to the Egress VTEP. When it arrives, the VTEP removes the IP and UDP headers, and delivers the frame as normal.

Why Only Ingress VNI is Configured?

The ingress VNI is configured to encapsulate traffic originating on the local device into the correct VXLAN tunnel, based on the VLAN or IP subnet associated with the traffic.

On the egress side, the VNI is automatically determined based on the EVPN control plane (BGP advertisements). Each VTEP (VXLAN Tunnel Endpoint) learns which VNI corresponds to which destination MAC or IP address, eliminating the need for manual configuration of egress VNIs.

How Does Egress VNI Auto-Detection Work?

When EVPN is used with VXLAN, iBGP distributes information about MAC addresses, IP addresses, and VNIs for each EVI.

A remote VTEP learns this mapping and can decapsulate the incoming VXLAN traffic based on the VNI in the packet header.

Since the control plane already shares the necessary mappings, the egress VNI doesn't need to be manually configured.

EVPN-MH configuration

- iBGP session is established between leafs (iBGP-Overlay evpn)
- Server facing interfaces configure (Access, Trunk, Lag, LACP)
- Ethernet segment configure for MH.
- Load-balancing for all-active multihoming segments - ecmp
- VXLAN tunnel interface configuration under tunnel interface (ingress VNI)
- Access, trunk and vxlan interfaces with the mac-vrf
- EVPN in MAC-VRF (bgp-vpn, bgp-evpn, EVPN Virtual Identifier (EVI))

iBGP session is established between leaf-spine (iBGP-Overlay evpn)

Now that the DC fabric has a routed underlay, and the loopbacks of the leaf, spine switches are mutually reachable, we can proceed with the VXLAN based EVPN service configuration. We will use spine as Route Reflector so that new leaf configuration will be easier.

For that iBGP configuration we will create a group called iBGP-overlay which will have the peer-as and local-as set to 501 to form an iBGP neighborship. The group will also host the same permissive all routing policy, enabled evpn and disabled ipv4-unicast address families.

Then for each leaf we add a new BGP neighbor addressed by the remote system0 interface address and local system address as the source.

info network-instance default protocols bgp group iBGP-overlay

info network-instance default protocols bgp neighbor 10.0.1.1

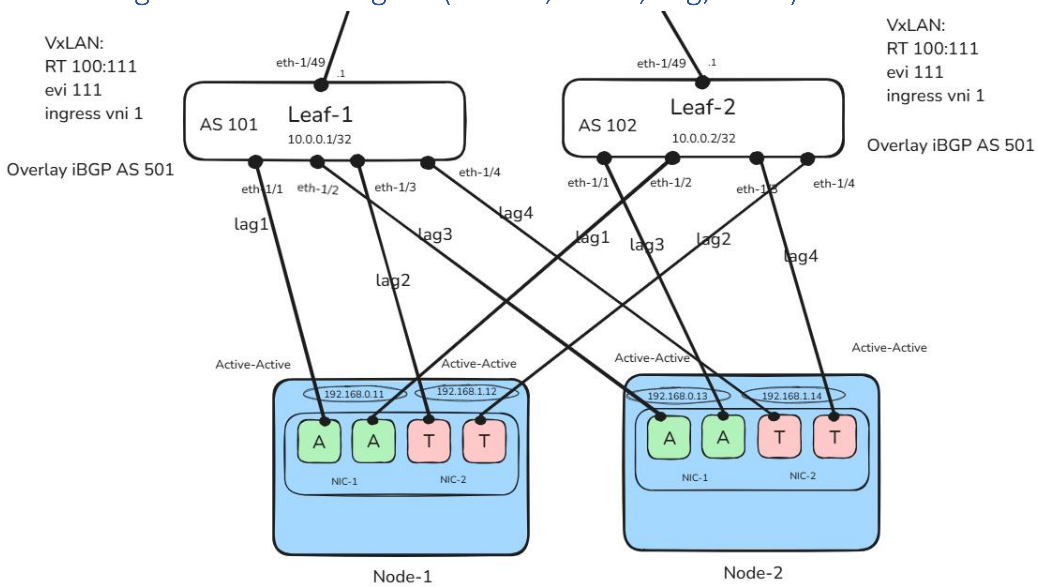
```
A:leaf1# info network-instance default protocols bgp group iBGP-overlay
    network-instance default {
        protocols {
            bgp {
                group iBGP-overlay {
                    export-policy all
                    import-policy all
                    peer-as 501
                    afi-safi evpn {
                        admin-state enable
                    }
                    afi-safi ipv4-unicast {
                        admin-state disable
                    }
                    local-as {
                        as-number 501
                    }
                    timers {
                        minimum-advertisement-interval 1
                    }
                }
            }
        }
    }
--{ candidate shared default }--[  ]--
A:leaf1# info network-instance default protocols bgp neighbor 10.0.1.1
    network-instance default {
        protocols {
            bgp {
                neighbor 10.0.1.1 {
                    peer-group iBGP-overlay
                    transport {
                        local-address 10.0.0.1
                    }
                }
            }
        }
    }
--{ candidate shared default }--[  ]--
```

Ensure that the iBGP session is established.

```
A:leaf1# show network-instance default protocols bgp neighbor 10.0.1.1
------------------------------------------------------------------------------------------------------------------------
BGP neighbor summary for network-instance "default"
Flags: S static, D dynamic, L discovered by LLDP, B BFD enabled, - disabled, * slow
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
+-------------+------------------+-------------+-----+-------+-----------+-----------+----------+------------------+
| Net-Inst    |       Peer       |    Group    | Fla | Peer- |   State   |  Uptime   | AFI/SAFI |  [Rx/Active/Tx]  |
|             |                  |             | gs  |  AS   |           |           |          |                  |
+=============+==================+=============+=====+=======+===========+===========+==========+==================+
| default     | 10.0.1.1         | iBGP-       | S   | 501   | establish | 0d:0h:50m | evpn     | [13/13/15]       |
|             |                  | overlay     |     |       | ed        | :9s       |          |                  |
+-------------+------------------+-------------+-----+-------+-----------+-----------+----------+------------------+
------------------------------------------------------------------------------------------------------------------------
Summary:
2 configured neighbors, 2 configured sessions are established,0 disabled peers
0 dynamic peers
--{ candidate shared default }--[  ]--
```

As we don't have any EVPN service created, there are no EVPN routes that are being sent/received. This snapshot is taked after creating EVPN service, so we are seeing in last column EVPN routes.

Server facing interfaces configure (Access, Trunk, Lag, LACP)



In this demo we have two Node/servers with 4 interfaces. We made Ethernet 1,2 as bond0 and Ethernet 3,4 as bond1. One bond interface is access and other is vlan 10 tagged.

On the leaf side, 4 lag interfaces are created for doing multi homing.

```
A:leaf1# info interface ethernet-1/1
    interface ethernet-1/1 {
        admin-state enable
        ethernet {
            aggregate-id lag1
        }
    }
--{ running }--[  ]--
A:leaf1# info interface ethernet-1/2
    interface ethernet-1/2 {
        admin-state enable
        ethernet {
            aggregate-id lag3
        }
    }
--{ running }--[  ]--
A:leaf1# info interface ethernet-1/3
    interface ethernet-1/3 {
        admin-state enable
        ethernet {
            aggregate-id lag2
        }
    }
--{ running }--[  ]--
A:leaf1# info interface ethernet-1/4
    interface ethernet-1/4 {
        admin-state enable
        ethernet {
            aggregate-id lag4
        }
    }
--{ running }--[  ]--
```

```
A:leaf1# info interface lag1
    interface lag1 {
        admin-state enable
        vlan-tagging true
        subinterface 0 {
            type bridged
            vlan {
                encap {
                    untagged {
                    }
                }
            }
        }
        lag {
            lag-type lacp
            member-speed 10G
            lacp {
                interval SLOW
                lacp-mode ACTIVE
                admin-key 11
                system-id-mac 00:00:00:00:00:11
                system-priority 11
            }
        }
    }
--{ running }--[  ]--
```

```
A:leaf1# info interface lag2
    interface lag2 {
        admin-state enable
        vlan-tagging true
        subinterface 0 {
            type bridged
            admin-state enable
            vlan {
                encap {
                    single-tagged-range {
                        low-vlan-id 10 {
                            high-vlan-id 15
                        }
                    }
                }
            }
        }
        lag {
            lag-type lacp
            member-speed 10G
            lacp {
                interval SLOW
                lacp-mode ACTIVE
                admin-key 12
                system-id-mac 00:00:00:00:00:12
                system-priority 12
            }
        }
    }
--{ running }--[  ]--
```

Tunnel/VXLAN interface:

After creating the access/trunk lag sub-interfaces we are proceeding with creation of the VXLAN/Tunnel interfaces. The VXLAN encapsulation in the dataplane allows MAC-VRFs of the same BD to be connected throughout the IP fabric.

The SR Linux models VXLAN as a tunnel-interface which has a vxlan-interface within. The tunnel-interface for VXLAN is configured with a name vxlan<N> where N = 0..255.

A **vxlan-interface** is configured under a **tunnel-interface**. At a minimum, a vxlan-interface must have an **index, type, and ingress VXLAN Network Identifier (VNI)**.

- The index can be a number in the range 0-4294967295.
- The type can be bridged or routed and indicates whether the vxlan-interface can be linked to a mac-vrf (bridged) or ip-vrf (routed).
- The ingress VNI is the **VXLAN Network Identifier** that the system looks for in incoming VXLAN packets to classify them to this vxlan-interface and its network-instance. VNI can be in the range of 1..16777215.
  The VNI is used to find the MAC-VRF where the inner MAC lookup is performed. **The egress VNI is not configured and is determined by the imported EVPN routes.**
  SR Linux requires that the egress VNI (discovered) matches the configured ingress VNI so that two leaf routers attached to the same BD can exchange packets.

The source IP used in the vxlan-interfaces is the IPv4 address of subinterface system0.0 in the default network-instance.

```
A:leaf1# info tunnel-interface vxlan1
    tunnel-interface vxlan1 {
        vxlan-interface 1 {
            type bridged
            ingress {
                vni 1
            }
        }
    }
--{ running }--[  ]--
A:leaf1#
```

To verify the tunnel interface configuration:

show tunnel-interface vxlan-interface brief

```
A:leaf1# show tunnel-interface vxlan-interface brief
-------------------------------------------------------------------------------------------
Show report for vxlan-tunnels
-------------------------------------------------------------------------------------------
+------------------+-----------------+---------+-------------+------------------+
| Tunnel Interface | VxLAN Interface |  Type   | Ingress VNI | Egress source-ip |
+==================+=================+=========+=============+==================+
| vxlan1           | vxlan1.1        | bridged | 1           | 10.0.0.1/32      |
+------------------+-----------------+---------+-------------+------------------+
-------------------------------------------------------------------------------------------
Summary
  1 tunnel-interfaces, 1 vxlan interfaces
  1 vxlan-destinations, 0 unicast, 0 es, 1 multicast, 0 ip
-------------------------------------------------------------------------------------------
--{ running }--[  ]--
```

The network-instance type mac-vrf functions as a broadcast domain. Each mac-vrf network-instance builds a bridge table composed of MAC addresses that can be learned via the data path on network-instance interfaces, via BGP EVPN or provided with static configuration.

We have added vxlan interface and lag interfaces to our mac vrf.

```
A:leaf1# info network-instance mac-vrf-1
    network-instance mac-vrf-1 {
        type mac-vrf
        admin-state enable
        interface lag1.0 {
        }
        interface lag2.0 {
        }
        interface lag3.0 {
        }
        interface lag4.0 {
        }
        vxlan-interface vxlan1.1 {
        }
        protocols {
            bgp-evpn {
                bgp-instance 1 {
                    admin-state enable
                    vxlan-interface vxlan1.1
                    evi 111
                    ecmp 2
                }
            }
            bgp-vpn {
                bgp-instance 1 {
                    route-target {
                        export-rt target:100:111
                        import-rt target:100:111
                    }
                }
            }
        }
    }
--{ running }--[  ]--
```

show tunnel-interface vxlan1 vxlan-interface 1 brief
show tunnel-interface vxlan-interface brief
show tunnel vxlan-tunnel all
show network-instance default tunnel-table all
show tunnel-interface vxlan1 vxlan-interface 1 bridge-table unicast-destinations destination *
show tunnel-interface vxlan1 vxlan-interface 1 bridge-table multicast-destinations destination *
show network-instance default protocols bgp routes evpn route-type summary

```
A:leaf1#  show tunnel-interface vxlan-interface brief
-----------------------------------------------------------------------------------------------------------------
Show report for vxlan-tunnels
-----------------------------------------------------------------------------------------------------------------
+------------------+-----------------+----------+-------------+------------------+
| Tunnel Interface | VxLAN Interface |  Type    | Ingress VNI | Egress source-ip |
+==================+=================+==========+=============+==================+
| vxlan1           | vxlan1.1        | bridged  | 1           | 10.0.0.1/32      |
+------------------+-----------------+----------+-------------+------------------+
-----------------------------------------------------------------------------------------------------------------
Summary
  1 tunnel-interfaces, 1 vxlan interfaces
  1 vxlan-destinations, 0 unicast, 0 es, 1 multicast, 0 ip
-----------------------------------------------------------------------------------------------------------------
--{ candidate shared default }--[  ]--
A:leaf1#  show tunnel vxlan-tunnel all
-----------------------------------------------------------------------------------------------------------------
Show report for vxlan-tunnels
-----------------------------------------------------------------------------------------------------------------
+--------------+---------------+--------------------------+
| VTEP Address |     Index     |       Last Change        |
+==============+===============+==========================+
| 10.0.0.2     | 8349072465251 | 2024-11-16T17:40:24.000Z |
+--------------+---------------+--------------------------+
1 VXLAN tunnels, 1 active, 0 inactive
-----------------------------------------------------------------------------------------------------------------
--{ candidate shared default }--[  ]--
A:leaf1# show network-instance default tunnel-table all
-----------------------------------------------------------------------------------------------------------------
IPv4 tunnel table of network-instance "default"
-----------------------------------------------------------------------------------------------------------------
+-----------------------------------+--------+---------------+----------+-----+---------+---------+--------------------------+------------------+------------------+
|            IPv4 Prefix            | Encaps |  Tunnel Type  |  Tunnel  | FIB | Metric  | Prefere |       Last Update        |  Next-hop (Type) |     Next-hop     |
|                                   |  Type  |               |    ID    |     |         |   nce   |                          |                  |                  |
+===================================+========+===============+==========+=====+=========+=========+==========================+==================+==================+
| 10.0.0.2/32                       | vxlan  | vxlan         | 3        | Y   | 0       | 0       | 2024-11-16T17:40:24.860Z | 192.168.11.2     | ethernet-1/49.0  |
|                                   |        |               |          |     |         |         |                          | (direct)         |                  |
+-----------------------------------+--------+---------------+----------+-----+---------+---------+--------------------------+------------------+------------------+
-----------------------------------------------------------------------------------------------------------------
1 VXLAN tunnels, 1 active, 0 inactive
```

```
A:leaf1# show tunnel-interface vxlan1 vxlan-interface 1 bridge-table unicast-destinations destination *
-----------------------------------------------------------------------------------------------------------------
Show report for vxlan-interface vxlan1.1 unicast destinations
-----------------------------------------------------------------------------------------------------------------
Destinations
-----------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------
Ethernet Segment Destinations
-----------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------
Summary
  0 unicast-destinations, 0 non-es, 0 es
  0 MAC addresses, 0 active, 0 non-active
-----------------------------------------------------------------------------------------------------------------
--{ candidate shared default }--[  ]--
A:leaf1# show tunnel-interface vxlan1 vxlan-interface 1 bridge-table multicast-destinations destination  *
-----------------------------------------------------------------------------------------------------------------
Show report for vxlan-interface vxlan1.1 multicast destinations (flooding-list)
-----------------------------------------------------------------------------------------------------------------
+--------------+------------+-------------------+----------------------+
| VTEP Address | Egress VNI | Destination-index | Multicast-forwarding |
+==============+============+===================+======================+
| 10.0.0.2     | 1          | 8349072465255     | BUM                  |
+--------------+------------+-------------------+----------------------+
-----------------------------------------------------------------------------------------------------------------
Summary
  1 multicast-destinations
-----------------------------------------------------------------------------------------------------------------
--{ candidate shared default }--[  ]--
A:leaf1#
```

```
A:leaf1# show network-instance default protocols bgp routes evpn route-type summary
-----------------------------------------------------------------------------------------------------------------
Show report for the BGP route table of network-instance "default"
-----------------------------------------------------------------------------------------------------------------
Status codes: u=used, *=valid, >=best, x=stale
Origin codes: i=IGP, e=EGP, ?=incomplete
-----------------------------------------------------------------------------------------------------------------
BGP Router ID: 10.0.0.1      AS: 101      Local AS: 101
-----------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------
Type 1 Ethernet Auto-Discovery Routes
+--------+--------------------------------+-------------------------------+------------+-------------------------+-------------------------+-------------------------+
| Status |      Route-distinguisher       |              ESI              |   Tag-ID   |        neighbor         |        Next-hop         |           VNI           |
+========+================================+===============================+============+=========================+=========================+=========================+
| u*>    | 10.0.0.2:111                   | 01:11:11:11:11:11:11:00:00:01 | 0          | 10.0.1.1                | 10.0.0.2                | 1                       |
| u*>    | 10.0.0.2:111                   | 01:11:11:11:11:11:11:00:00:01 | 4294967295 | 10.0.1.1                | 10.0.0.2                | -                       |
| u*>    | 10.0.0.2:111                   | 01:11:11:11:11:11:11:00:00:02 | 0          | 10.0.1.1                | 10.0.0.2                | 1                       |
| u*>    | 10.0.0.2:111                   | 01:11:11:11:11:11:11:00:00:02 | 4294967295 | 10.0.1.1                | 10.0.0.2                | -                       |
| u*>    | 10.0.0.2:111                   | 01:11:11:11:11:11:11:00:00:03 | 0          | 10.0.1.1                | 10.0.0.2                | 1                       |
| u*>    | 10.0.0.2:111                   | 01:11:11:11:11:11:11:00:00:03 | 4294967295 | 10.0.1.1                | 10.0.0.2                | -                       |
| u*>    | 10.0.0.2:111                   | 01:11:11:11:11:11:11:00:00:04 | 0          | 10.0.1.1                | 10.0.0.2                | 1                       |
| u*>    | 10.0.0.2:111                   | 01:11:11:11:11:11:11:00:00:04 | 4294967295 | 10.0.1.1                | 10.0.0.2                | -                       |
+--------+--------------------------------+-------------------------------+------------+-------------------------+-------------------------+-------------------------+
Type 3 Inclusive Multicast Ethernet Tag Routes
+--------+------------------------------------------+------------+-----------------+----------------------------------+----------------------------------+
| Status |           Route-distinguisher            |   Tag-ID   |  Originator-IP  |             neighbor             |             Next-Hop             |
+========+==========================================+============+=================+==================================+==================================+
| u*>    | 10.0.0.2:111                             | 0          | 10.0.0.2        | 10.0.1.1                         | 10.0.0.2                         |
+--------+------------------------------------------+------------+-----------------+----------------------------------+----------------------------------+
-----------------------------------------------------------------------------------------------------------------
Type 4 Ethernet Segment Routes
+--------+-------------------------------+-------------------------------+-------------------------------+-------------------------------+-------------------------------+
| Status |      Route-distinguisher      |              ESI              |      originating-router       |           neighbor            |           Next-Hop            |
+========+===============================+===============================+===============================+===============================+===============================+
| u*>    | 10.0.0.2:0                    | 01:11:11:11:11:11:11:00:00:01 | 10.0.0.2                      | 10.0.1.1                      | 10.0.0.2                      |
| u*>    | 10.0.0.2:0                    | 01:11:11:11:11:11:11:00:00:02 | 10.0.0.2                      | 10.0.1.1                      | 10.0.0.2                      |
| u*>    | 10.0.0.2:0                    | 01:11:11:11:11:11:11:00:00:03 | 10.0.0.2                      | 10.0.1.1                      | 10.0.0.2                      |
| u*>    | 10.0.0.2:0                    | 01:11:11:11:11:11:11:00:00:04 | 10.0.0.2                      | 10.0.1.1                      | 10.0.0.2                      |
+--------+-------------------------------+-------------------------------+-------------------------------+-------------------------------+-------------------------------+
-----------------------------------------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
8 Ethernet Auto-Discovery routes 8 used, 8 valid
0 MAC-IP Advertisement routes 0 used, 0 valid
1 Inclusive Multicast Ethernet Tag routes 1 used, 1 valid
4 Ethernet Segment routes 4 used, 4 valid
0 IP Prefix routes 0 used, 0 valid
--------------------------------------------------------------------------------
--{ candidate shared default }--[  ]--
A:leaf1#
```

Ingress and Egress VNI is same. VxLAN tunnel is created. Now we will create ES.

Ethernet Segment

```
A:leaf1# info system network-instance protocols
    system {
        network-instance {
            protocols {
                evpn {
                    ethernet-segments {
                        bgp-instance 1 {
                            ethernet-segment ES-1 {
                                admin-state enable
                                esi 01:11:11:11:11:11:11:00:00:01
                                multi-homing-mode all-active
                                interface lag1 {
                                }
                            }
                            ethernet-segment ES-2 {
                                admin-state enable
                                esi 01:11:11:11:11:11:11:00:00:02
                                multi-homing-mode all-active
                                interface lag2 {
                                }
                            }
                            ethernet-segment ES-3 {
                                admin-state enable
                                esi 01:11:11:11:11:11:11:00:00:03
                                multi-homing-mode all-active
                                interface lag3 {
                                }
                            }
                            ethernet-segment ES-4 {
                                admin-state enable
                                esi 01:11:11:11:11:11:11:00:00:04
                                multi-homing-mode all-active
                                interface lag4 {
                                }
                            }
                        }
                    }
                }
                bgp-vpn {
                    bgp-instance 1 {
                    }
                }
            }
        }
    }
--{ running }--[  ]--
```

The ethernet-segment is created with a name ES-1 under bgp-instance 1 with the all-active mode.

For a multihomed site, each Ethernet segment (ES) is identified by a unique non-zero identifier called an Ethernet Segment Identifier (ESI).

An ESI is encoded as a 10-octet integer in line format with the most significant octet sent first.

The esi and multi-homing-mode must match in all ES peers. At last, we assign the interface lag1 to ES-1, lag2 to ES-2, lag3 to ES3 and lag4 to ES-4 for both leafs.

Besides the ethernet segments, bgp-vpn is also configured with bgp-instance 1 to use the BGP information (RT/RD) for the ES routes exchanged in EVPN to enable multihoming.

To provide the load-balancing for all-active multihoming segments, set ecmp to the expected number of leaves (PE) serving the CE1. Since we have two leaves connected to CE1, we set ecmp 2.

```
A:leaf1# info network-instance mac-vrf-1
    network-instance mac-vrf-1 {
        type mac-vrf
        admin-state enable
        interface lag1.0 {
        }
        interface lag2.0 {
        }
        interface lag3.0 {
        }
        interface lag4.0 {
        }
        vxlan-interface vxlan1.1 {
        }
        protocols {
            bgp-evpn {
                bgp-instance 1 {
                    admin-state enable
                    vxlan-interface vxlan1.1
                    evi 111
                    ecmp 2
                }
            }
            bgp-vpn {
                bgp-instance 1 {
                    route-target {
                        export-rt target:100:111
                        import-rt target:100:111
                    }
                }
            }
        }
    }
--{ running }--[  ]--
```

EVPN in MAC-VRF

To advertise the locally learned MACs to the remote leafs we have to configure EVPN in our mac-vrf-1 network-instance.

EVPN configuration under the mac-vrf network instance will require two configuration containers:

- bgp-vpn - provides the configuration of the bgp-instances where the route-distinguisher and the import/export route-targets used for the EVPN routes exist.
- bgp-evpn - hosts all the commands required to enable EVPN in the network-instance. At a minimum, a reference to bgp-instance 1 is configured, along with the reference to the vxlan-interface and the EVPN Virtual Identifier (EVI).

```
A:leaf1# info network-instance mac-vrf-1
    network-instance mac-vrf-1 {
        type mac-vrf
        admin-state enable
        interface lag1.0 {
        }
        interface lag2.0 {
        }
        interface lag3.0 {
        }
        interface lag4.0 {
        }
        vxlan-interface vxlan1.1 {
        }
        protocols {
            bgp-evpn {
                bgp-instance 1 {
                    admin-state enable
                    vxlan-interface vxlan1.1
                    evi 111
                    ecmp 2
                }
            }
            bgp-vpn {
                bgp-instance 1 {
                    route-target {
                        export-rt target:100:111
                        import-rt target:100:111
                    }
                }
            }
        }
    }
--{ running }--[  ]--
A:leaf1#
```

Once configured, the bgp-vpn instance can be checked to have the RT/RD values set:

```
A:leaf1# show network-instance  mac-vrf-1 protocols bgp-vpn bgp-instance 1
==============================================================================
Net Instance    : mac-vrf-1
    bgp Instance 1
------------------------------------------------------------------------------
        route-distinguisher: 10.0.0.1:111, auto-derived-from-evi
        export-route-target: target:100:111, manual
        import-route-target: target:100:111, manual
==============================================================================
--{ running }--[  ]--
A:leaf1#
```

VNI to EVI mapping: SR Linux supports an interoperability mode in which SR Linux leaf nodes can be attached to VLAN-aware bundle broadcast domains along with other third-party routers.

When the BGP-EVPN is configured in the mac-vrf instance, the leafs start to exchange EVPN routes, which we can verify with the following commands:

```
A:leaf1# show network-instance default protocols bgp neighbor 10.0.0.2
----------------------------------------------------------------------------------------------------------------
BGP neighbor summary for network-instance "default"
Flags: S static, D dynamic, L discovered by LLDP, B BFD enabled, - disabled, * slow
----------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------
+-----------------+---------------------------+-----------------------+-------+----------+------------+------------+----------+---------------------------+
|    Net-Inst     |           Peer            |         Group         | Flags | Peer-AS  |   State    |   Uptime   | AFI/SAFI |      [Rx/Active/Tx]       |
+=================+===========================+=======================+=======+==========+============+============+==========+===========================+
| default         | 10.0.0.2                  | eBGP-overlay          | S     | 501      | established| 0d:16h:0m:14s | evpn  | [13/13/15]                |
+-----------------+---------------------------+-----------------------+-------+----------+------------+------------+----------+---------------------------+
----------------------------------------------------------------------------------------------------------------
Summary:
2 configured neighbors, 2 configured sessions are established,0 disabled peers
0 dynamic peers
--{ running }--[  ]--
```

leaf1 received/sent 1 route out of 13/13/15 is an EVPN Inclusive Multicast Ethernet Tag route (IMET or type 3, RT3). The IMET route is advertised as soon as bgp-evpn is enabled in the MAC-VRF; it has the following purpose:

- Auto-discovery of the remote VTEPs attached to the same EVI
- Creation of a default flooding list in the MAC-VRF so that BUM frames are replicated

The IMET/RT3 routes can be viewed in summary and detailed modes:

```
A:leaf1# show network-instance default protocols bgp routes evpn route-type 3 summary
-------------------------------------------------------------------------------------
Show report for the BGP route table of network-instance "default"
-------------------------------------------------------------------------------------
Status codes: u=used, *=valid, >=best, x=stale
Origin codes: i=IGP, e=EGP, ?=incomplete
-------------------------------------------------------------------------------------
BGP Router ID: 10.0.0.1      AS: 101      Local AS: 101
-------------------------------------------------------------------------------------
Type 3 Inclusive Multicast Ethernet Tag Routes
+--------+--------------------------------+----------+-----------------+-----------------------------+-----------------------------+
| Status |      Route-distinguisher       |  Tag-ID  |  Originator-IP  |          neighbor           |          Next-Hop           |
+========+================================+==========+=================+=============================+=============================+
| u*>    | 10.0.0.2:111                   | 0        | 10.0.0.2        | 10.0.0.2                    | 10.0.0.2                    |
+--------+--------------------------------+----------+-----------------+-----------------------------+-----------------------------+
-------------------------------------------------------------------------------------
1 Inclusive Multicast Ethernet Tag routes 1 used, 1 valid
-------------------------------------------------------------------------------------
--{ running }--[  ]--
A:leaf1# show network-instance default protocols bgp routes evpn route-type 3 detail
-------------------------------------------------------------------------------------
Show report for the EVPN routes in network-instance  "default"
-------------------------------------------------------------------------------------
Route Distinguisher: 10.0.0.2:111
Tag-ID             : 0
Originating router : 10.0.0.2
neighbor           : 10.0.0.2
Received paths     : 1
  Path 1: <Best,Valid,Used,>
    VNI             : 1
    Route source    : neighbor 10.0.0.2 (last modified 15h3m22s ago)
    Route preference: No MED, LocalPref is 100
    Atomic Aggr     : false
    BGP next-hop    : 10.0.0.2
    AS Path         :  i
    Communities     : [target:100:111, bgp-tunnel-encap:VXLAN]
    RR Attributes   : No Originator-ID, Cluster-List is []
    Aggregation     : None
    Unknown Attr    : None
    Invalid Reason  : None
    Tie Break Reason: none
-------------------------------------------------------------------------------------
```

When the IMET routes from leaf2 are imported for mac-vrf-1 network-instance, the corresponding multicast VXLAN destinations are added and can be checked with the following command:

show tunnel-interface vxlan1 vxlan-interface 1 bridge-table multicast-destinations destination *

```
A:leaf1# show tunnel-interface vxlan1 vxlan-interface 1 bridge-table multicast-destinations destination *
-----------------------------------------------------------------------------------------------------------
Show report for vxlan-interface vxlan1.1 multicast destinations (flooding-list)
-----------------------------------------------------------------------------------------------------------
+--------------+------------+-------------------+----------------------+
| VTEP Address | Egress VNI | Destination-index | Multicast-forwarding |
+==============+============+===================+======================+
| 10.0.0.2     | 1          | 8349072465247     | BUM                  |
+--------------+------------+-------------------+----------------------+
-----------------------------------------------------------------------------------------------------------
Summary
  1 multicast-destinations
-----------------------------------------------------------------------------------------------------------
--{ running }--[  ]--
```

This multicast destination means that BUM frames received on a bridged sub-interface are ingress-replicated to the VTEPs for that EVI as per the table above. For example any ARP traffic will be distributed (ingress-replicated) to the VTEPs from multicast destinations table.

As to the unicast destinations there are none so far, and this is because we haven't yet received any MAC/IP RT2 EVPN routes. But before looking into the RT2 EVPN routes, let's zoom into VXLAN tunnels that got built right after we receive the first IMET RT3 routes.

VXLAN tunnels

After receiving EVPN routes from the remote leafs with VXLAN encapsulation, SR Linux creates VXLAN tunnels towards remote VTEP, whose address is received in EVPN IMET routes.

show tunnel vxlan-tunnel all

```
A:leaf1# show tunnel vxlan-tunnel all
-----------------------------------------------------------------
Show report for vxlan-tunnels
-----------------------------------------------------------------
+--------------+---------------+--------------------------+
| VTEP Address |     Index     |       Last Change        |
+==============+===============+==========================+
| 10.0.0.2     | 8349072465238 | 2024-11-15T03:28:27.000Z |
+--------------+---------------+--------------------------+
1 VXLAN tunnels, 1 active, 0 inactive
-----------------------------------------------------------------
--{ running }--[  ]--
```

The VXLAN tunnel is built between the vxlan interfaces in the MAC-VRF network instances, which internally use system interfaces of the default network instance as a VTEP:

Once a VTEP is created in the vxlan-tunnel table with a non-zero allocated index, an entry in the tunnel-table is also created for the tunnel.

show network-instance default tunnel-table all

```
A:leaf1# show network-instance default tunnel-table all
-------------------------------------------------------------------------------------------------------------------------------------------------
IPv4 tunnel table of network-instance "default"
-------------------------------------------------------------------------------------------------------------------------------------------------
+-------------------------------------+--------+----------------+----------+-----+----------+---------+--------------------------+-----------------------+----------------------+
|             IPv4 Prefix             | Encaps |  Tunnel Type   |  Tunnel  | FIB |  Metric  | Prefere |       Last Update        |    Next-hop (Type)    |       Next-hop       |
|                                     |  Type  |                |    ID    |     |          |   nce   |                          |                       |                      |
+=====================================+========+================+==========+=====+==========+=========+==========================+=======================+======================+
| 10.0.0.2/32                         | vxlan  | vxlan          | 1        | Y   | 0        | 0       | 2024-11-15T03:28:27.663Z | 192.168.11.2          | ethernet-1/49.0      |
|                                     |        |                |          |     |          |         |                          | (direct)              |                      |
+-------------------------------------+--------+----------------+----------+-----+----------+---------+--------------------------+-----------------------+----------------------+
-------------------------------------------------------------------------------------------------------------------------------------------------
1 VXLAN tunnels, 1 active, 0 inactive
-------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------
IPv6 tunnel table of network-instance "default"
-------------------------------------------------------------------------------------------------------------------------------------------------
<no_entries>
-------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------------------------------------
--{ running }--[  ]--
```

EVPN MAC/IP routes:

when the leafs exchanged only EVPN IMET routes they build the BUM flooding tree (aka multicast destinations), but unicast destinations are yet unknown, which is seen in the below output:

show tunnel-interface vxlan1 vxlan-interface 1 bridge-table unicast-destinations destination *

```
A:leaf1# show tunnel-interface vxlan1 vxlan-interface 1 bridge-table unicast-destinations destination *
-------------------------------------------------------------------------------------------------------
Show report for vxlan-interface vxlan1.1 unicast destinations
-------------------------------------------------------------------------------------------------------
Destinations
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
Ethernet Segment Destinations
-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
Summary
  0 unicast-destinations, 0 non-es, 0 es
  0 MAC addresses, 0 active, 0 non-active
-------------------------------------------------------------------------------------------------------
--{ running }--[  ]--
A:leaf1#
```

This is due to the fact that no MAC/IP EVPN routes are being advertised yet. If we take a look at the MAC table of the mac-vrf-1, we will see that no local MAC addresses are there, and this is because the servers haven't yet sent any frames towards the leafs.

show network-instance mac-vrf-1 bridge-table mac-table all

```
A:leaf1# show network-instance mac-vrf-1 bridge-table mac-table all
----------------------------------------------------------------------------------------------------------------------------------------
Mac-table of network instance mac-vrf-1
----------------------------------------------------------------------------------------------------------------------------------------
+-------------------+------------------------------------------------------+------------+----------------+---------+--------+------------------------------------+
|      Address      |                     Destination                      | Dest Index |      Type      | Active  | Aging  |            Last Update             |
+===================+======================================================+============+================+=========+========+====================================+
| 00:C1:AB:00:00:11 | lag1.0                                               | 5          | learnt         | true    | 279    | 2024-11-15T15:17:46.000Z           |
| 00:C1:AB:00:00:12 | lag2.0                                               | 9          | learnt         | true    | 300    | 2024-11-15T20:00:06.000Z           |
| 00:C1:AB:00:00:13 | lag3.0                                               | 7          | learnt         | true    | 279    | 2024-11-15T15:17:46.000Z           |
| 00:C1:AB:00:00:14 | lag4.0                                               | 10         | learnt         | true    | 300    | 2024-11-15T20:00:06.000Z           |
+-------------------+------------------------------------------------------+------------+----------------+---------+--------+------------------------------------+
Total Irb Macs               :    0 Total    0 Active
Total Static Macs            :    0 Total    0 Active
Total Duplicate Macs         :    0 Total    0 Active
Total Learnt Macs            :    4 Total    4 Active
Total Evpn Macs              :    0 Total    0 Active
Total Evpn static Macs       :    0 Total    0 Active
Total Irb anycast Macs       :    0 Total    0 Active
Total Proxy Antispoof Macs   :    0 Total    0 Active
Total Reserved Macs          :    0 Total    0 Active
Total Eth-cfm Macs           :    0 Total    0 Active
--{ running }--[  ]--
A:leaf1#
```

When traffic is exchanged between srv1 and srv2, the MACs are learned on the access bridged sub-interfaces and advertised in EVPN MAC/IP routes (type 2, RT2). The MAC/IP routes are imported, and the MACs programmed in the mac-table.

The below output shows the MAC/IP EVPN route that leaf1 received from its neighbor. The NLRI information contains the MAC of the srv2:

show network-instance default protocols bgp routes evpn route-type 2 summary

The MAC/IP EVPN routes also triggers the creation of the unicast tunnel destinations which were empty before:

show tunnel-interface vxlan1 vxlan-interface 1 bridge-table unicast-destinations destination *

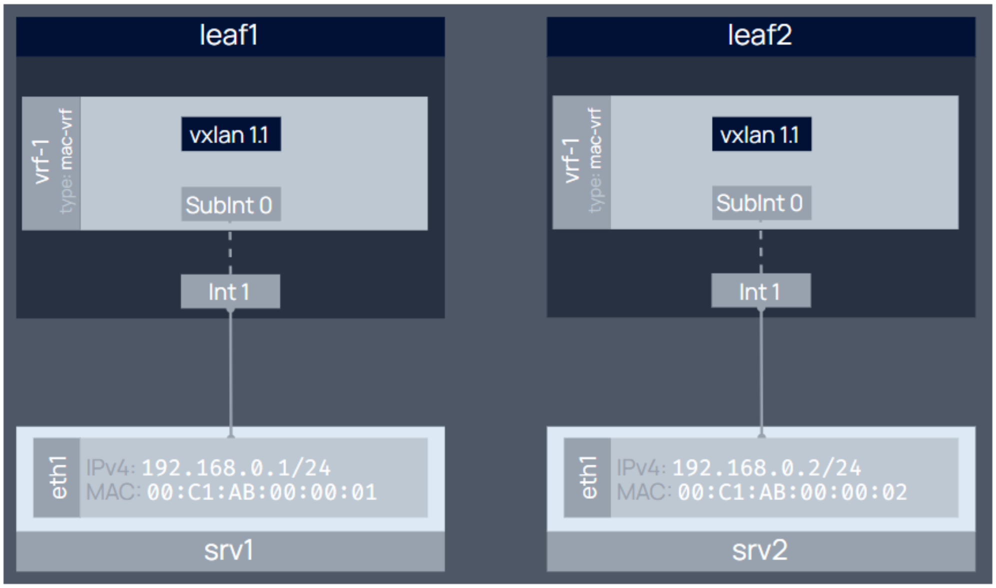
Node configuration

- LACP aware 2 bond interfaces (Access, Trunk) configure with 4 physical links with leaf.

```
ip link add bond0 type bond mode 802.3ad
ip link add bond1 type bond mode 802.3ad
ip link set address 00:c1:ab:00:00:11 dev bond0
ip link set address 00:c1:ab:00:00:12 dev bond1
ip addr add 192.168.0.11/24 dev bond0

ip link set eth1 down
ip link set eth2 down
ip link set eth3 down
ip link set eth4 down
ip link set eth1 master bond0
ip link set eth2 master bond0
ip link set eth3 master bond1
ip link set eth4 master bond1
ip link set eth1 up
ip link set eth2 up
ip link set eth3 up
ip link set eth4 up
ip link set bond0 up
ip link set bond1 up

ip link add link bond1 name bond1.10 type vlan id 10
ip addr add 192.168.1.12/24 dev bond1.10
ip link set dev bond1.10 up
```



This snapshot is a demo for illustration of server to leaf connection. In our case, Bond0 and Bond1 interfaces has MAC and IP, which is connected to lag interface sub-interface. This sub-interface and vxlan1.1 subinterface are in same BD.

Verification

Let's verify that the configuration we have done so far is working as expected.

LAG

let's check the LAG and LACP status on our leaf switches

show interface lag1

```
A:leaf1# show interface lag1
================================================================
lag1 is up, speed None, type None
  lag1.0 is up
    Network-instances:
      * Name: mac-vrf-1
    Encapsulation   : null
    Type            : bridged
----------------------------------------------------------------
================================================================
--{ running }--[  ]--
A:leaf1# show interface lag2
================================================================
lag2 is up, speed None, type None
  lag2.0 is up
    Network-instances:
      * Name: mac-vrf-1
    Encapsulation   : vlan id 10-15
    Type            : bridged
----------------------------------------------------------------
================================================================
--{ running }--[  ]--
A:leaf1# show interface lag3
================================================================
lag3 is up, speed None, type None
  lag3.0 is up
    Network-instances:
      * Name: mac-vrf-1
    Encapsulation   : null
    Type            : bridged
----------------------------------------------------------------
================================================================
--{ running }--[  ]--
A:leaf1# show interface lag4
================================================================
lag4 is up, speed None, type None
  lag4.0 is up
    Network-instances:
      * Name: mac-vrf-1
    Encapsulation   : vlan id 10-15
    Type            : bridged
----------------------------------------------------------------
================================================================
--{ running }--[  ]--
A:leaf1#
```

Lacp status:

show lag lag1 lacp-state

```
A:leaf1# show lag lag1 lacp-state
-----------------------------------------------------------------------------------------------------------------------------------------
LACP State for lag1
-----------------------------------------------------------------------------------------------------------------------------------------
Lag Id         : lag1
Interval       : SLOW
Mode           : ACTIVE
System Id      : 00:00:00:00:00:11
System Priority: 11
+--------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
|   Members    |     Oper state     |      Activity      |      Timeout       |       State        |     System Id      |      Oper key      |     Partner Id     |    Partner Key     |      Port No       |  Partner Port No   |
+==============+====================+====================+====================+====================+====================+====================+====================+====================+====================+====================+
| ethernet-1/1 | up                 | ACTIVE             | LONG               | IN_SYNC/True/True/ | 00:00:00:00:00:11  | 11                 | 00:C1:AB:00:00:11  | 15                 | 1                  | 1                  |
|              |                    |                    |                    | True               |                    |                    |                    |                    |                    |                    |
+--------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
-----------------------------------------------------------------------------------------------------------------------------------------
--{ running }--[  ]--
A:leaf1# show lag lag2 lacp-state
-----------------------------------------------------------------------------------------------------------------------------------------
LACP State for lag2
-----------------------------------------------------------------------------------------------------------------------------------------
Lag Id         : lag2
Interval       : SLOW
Mode           : ACTIVE
System Id      : 00:00:00:00:00:12
System Priority: 12
+--------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
|   Members    |     Oper state     |      Activity      |      Timeout       |       State        |     System Id      |      Oper key      |     Partner Id     |    Partner Key     |      Port No       |  Partner Port No   |
+==============+====================+====================+====================+====================+====================+====================+====================+====================+====================+====================+
| ethernet-1/3 | up                 | ACTIVE             | LONG               | IN_SYNC/True/True/ | 00:00:00:00:00:12  | 12                 | AA:C1:AB:5C:AF:63  | 15                 | 3                  | 1                  |
|              |                    |                    |                    | True               |                    |                    |                    |                    |                    |                    |
+--------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
-----------------------------------------------------------------------------------------------------------------------------------------
--{ running }--[  ]--
A:leaf1# show lag lag3 lacp-state
-----------------------------------------------------------------------------------------------------------------------------------------
LACP State for lag3
-----------------------------------------------------------------------------------------------------------------------------------------
Lag Id         : lag3
Interval       : SLOW
Mode           : ACTIVE
System Id      : 00:00:00:00:00:13
System Priority: 13
+--------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
|   Members    |     Oper state     |      Activity      |      Timeout       |       State        |     System Id      |      Oper key      |     Partner Id     |    Partner Key     |      Port No       |  Partner Port No   |
+==============+====================+====================+====================+====================+====================+====================+====================+====================+====================+====================+
| ethernet-1/2 | up                 | ACTIVE             | LONG               | IN_SYNC/True/True/ | 00:00:00:00:00:13  | 13                 | 00:C1:AB:00:00:13  | 15                 | 2                  | 1                  |
|              |                    |                    |                    | True               |                    |                    |                    |                    |                    |                    |
+--------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
-----------------------------------------------------------------------------------------------------------------------------------------
```

The LAG status and network instance it belongs to can be seen in the show interface lag1 output, and show lag lag1 lacp-state command shows the LACP parameters as well as the status information of the member ports.

The output on leaf1 and leaf2 is expected to be the same, except for the partner port no, which is unique per peer.

Ethernet Segment

show system network-instance ethernet-segments ES-1 detail

```
A:leaf1# show system network-instance ethernet-segments ES-1 detail
==========================================================================
Ethernet Segment
==========================================================================
Name                  : ES-1
Admin State           : enable              Oper State        : up
ESI                   : 01:11:11:11:11:11:11:00:00:01
Multi-homing          : all-active          Oper Multi-homing : all-active
Interface             : lag1
Next Hop              : N/A
EVI                   : N/A
ES Activation Timer   : None
DF Election           : default             Oper DF Election  : default

Last Change           : 2024-11-15T13:36:30.955Z
==========================================================================
MAC-VRF    Actv Timer Rem   DF
ES-1       0                Yes
--------------------------------------------------------------------------
DF Candidates
--------------------------------------------------------------------------
Network-instance        ES Peers
mac-vrf-1               10.0.0.1
mac-vrf-1               10.0.0.2 (DF)
==========================================================================
--{ running }--[  ]--
A:leaf1# show system network-instance ethernet-segments ES-2 detail
==========================================================================
Ethernet Segment
==========================================================================
Name                  : ES-2
Admin State           : enable              Oper State        : up
ESI                   : 01:11:11:11:11:11:11:00:00:02
Multi-homing          : all-active          Oper Multi-homing : all-active
Interface             : lag2
Next Hop              : N/A
EVI                   : N/A
ES Activation Timer   : None
DF Election           : default             Oper DF Election  : default

Last Change           : 2024-11-15T05:41:20.107Z
==========================================================================
MAC-VRF    Actv Timer Rem   DF
ES-2       0                Yes
--------------------------------------------------------------------------
DF Candidates
--------------------------------------------------------------------------
Network-instance        ES Peers
mac-vrf-1               10.0.0.1
mac-vrf-1               10.0.0.2 (DF)
```

The configured ES parameters, operational status, as well as the ES peers and the selected Designated Forwarder (DF) are displayed here.

Traffic test

Let's send some CE to CE traffic to see if multihoming works and traffic is utilizing all available links.

ssh admin@clab-evpn-mh-my-ce1

Credentials admin:srllabs@123

Open 3 terminals on node 1. And one terminal on node 2

sudo tcpdump -ni eth3

sudo tcpdump -ni eth4

We are doing packet capture on bond1 interface two physical interfaces eth3 and eth4.

Basic Ping Sweep To see which hosts are up within a network:

nmap -sn 192.168.1.0/24

Start the iperf3 server on the destination (192.168.1.12):

iperf3 -s

Run the iperf3 client on the source (192.168.1.14), specifying the source IP:

iperf3 -c 192.168.1.12 -B 192.168.1.14

iperf3 -c 192.168.1.12 -B 192.168.1.14 -u -b 30G

```
ce1:~$ iperf3 -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 192.168.1.14, port 48031
[  5] local 192.168.1.12 port 5201 connected to 192.168.1.14 port 60737
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  5]   0.00-1.00   sec  2.74 MBytes  23.0 Mbits/sec  0.072 ms  162749/164734 (99%)
[  5]   1.00-2.00   sec  1.38 MBytes  11.6 Mbits/sec  0.066 ms  175771/176772 (99%)
[  5]   2.00-3.00   sec  1.38 MBytes  11.6 Mbits/sec  0.041 ms  163169/164170 (99%)
[  5]   3.00-4.00   sec  1.38 MBytes  11.6 Mbits/sec  0.048 ms  181278/182278 (99%)
[  5]   4.00-5.00   sec  1.38 MBytes  11.6 Mbits/sec  0.040 ms  175381/176382 (99%)
[  5]   5.00-6.00   sec  1.38 MBytes  11.6 Mbits/sec  0.045 ms  181859/182860 (99%)
[  5]   6.00-7.00   sec  1.38 MBytes  11.6 Mbits/sec  0.020 ms  175884/176885 (99%)
[  5]   7.00-8.00   sec  1.38 MBytes  11.6 Mbits/sec  0.054 ms  170501/171502 (99%)
[  5]   8.00-9.00   sec  1.38 MBytes  11.6 Mbits/sec  0.032 ms  166274/167275 (99%)
[  5]   9.00-10.00  sec  1.38 MBytes  11.6 Mbits/sec  0.045 ms  174538/175539 (99%)
[  5]  10.00-11.00  sec  0.00 Bytes  0.00 bits/sec  0.045 ms  0/0 (0%)
[  5]  11.00-11.78  sec  0.00 Bytes  0.00 bits/sec  0.045 ms  0/0 (0%)
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  5]   0.00-11.78  sec  15.2 MBytes  10.8 Mbits/sec  0.045 ms  1727404/1738397 (99%)  receiver
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

```
ce2:~$ iperf3 -c 192.168.1.12 -B 192.168.1.14 -u -b 30G
Connecting to host 192.168.1.12, port 5201
[  5] local 192.168.1.14 port 60737 connected to 192.168.1.12 port 5201
[ ID] Interval           Transfer     Bitrate         Total Datagrams
[  5]   0.00-1.00   sec   247 MBytes  2.07 Gbits/sec  178603
[  5]   1.00-2.00   sec   245 MBytes  2.05 Gbits/sec  177069
[  5]   2.00-3.00   sec   226 MBytes  1.90 Gbits/sec  163948
[  5]   3.00-4.00   sec   255 MBytes  2.14 Gbits/sec  184390
[  5]   4.00-5.00   sec   242 MBytes  2.03 Gbits/sec  175243
[  5]   5.00-6.00   sec   254 MBytes  2.13 Gbits/sec  183881
[  5]   6.00-7.00   sec   243 MBytes  2.04 Gbits/sec  176096
[  5]   7.00-8.00   sec   237 MBytes  1.99 Gbits/sec  171834
[  5]   8.00-9.00   sec   231 MBytes  1.94 Gbits/sec  167481
[  5]   9.00-10.00  sec   246 MBytes  2.06 Gbits/sec  177910
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  5]   0.00-10.00  sec  2.37 GBytes  2.03 Gbits/sec  0.000 ms  0/1756455 (0%)  sender
[  5]   0.00-11.78  sec  15.2 MBytes  10.8 Mbits/sec  0.045 ms  1727404/1738397 (99%)  receiver

iperf Done.
```

Explanation:

- -u: Enables UDP mode (required for specifying bandwidth).
- -b 30G: Sets the bandwidth to 30 Gbps.
- -B 192.168.1.14: Binds the client to the source IP address 192.168.1.14.
- -c 192.168.1.12: Specifies the destination IP address.

On captured packets we can see traffics on both interfaces.

EVPN Routes

When doing the traffic tests, we triggered some EVPN routes exchange in the fabric.

Let's check which EVPN routes leaf1 and spine1, leaf2 and spine1 (ES peers) advertise to each other:

show network-instance default protocols bgp neighbor 10.0.1.1 advertised-routes evpn

```
A:leaf1# show network-instance default protocols bgp neighbor 10.0.1.1 advertised-routes evpn
----------------------------------------------------------------------------------------------------
Peer        : 10.0.1.1, remote AS: 501, local AS: 501
Type        : static
Description : None
Group       : iBGP-overlay
----------------------------------------------------------------------------------------------------
Origin codes: i=IGP, e=EGP, ?=incomplete
----------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
Type 1 Ethernet Auto-Discovery Routes
+--------------------------------+-------------------------------+------------+---------------------------+---------------------------+---------+---------------------------+
|      Route-distinguisher       |              ESI              |   Tag-ID   |         Next-Hop          |            MED            | LocPref |           Path            |
+================================+===============================+============+===========================+===========================+=========+===========================+
| 10.0.0.1:111                   | 01:11:11:11:11:11:11:00:00:01 | 0          | 10.0.0.1                  | -                         | 100     |                           |
| 10.0.0.1:111                   | 01:11:11:11:11:11:11:00:00:01 | 4294967295 | 10.0.0.1                  | -                         | 100     |                           |
| 10.0.0.1:111                   | 01:11:11:11:11:11:11:00:00:02 | 0          | 10.0.0.1                  | -                         | 100     |                           |
| 10.0.0.1:111                   | 01:11:11:11:11:11:11:00:00:02 | 4294967295 | 10.0.0.1                  | -                         | 100     |                           |
| 10.0.0.1:111                   | 01:11:11:11:11:11:11:00:00:03 | 0          | 10.0.0.1                  | -                         | 100     |                           |
| 10.0.0.1:111                   | 01:11:11:11:11:11:11:00:00:03 | 4294967295 | 10.0.0.1                  | -                         | 100     |                           |
| 10.0.0.1:111                   | 01:11:11:11:11:11:11:00:00:04 | 0          | 10.0.0.1                  | -                         | 100     |                           |
| 10.0.0.1:111                   | 01:11:11:11:11:11:11:00:00:04 | 4294967295 | 10.0.0.1                  | -                         | 100     |                           |
+--------------------------------+-------------------------------+------------+---------------------------+---------------------------+---------+---------------------------+
Type 2 MAC-IP Advertisement Routes
+--------------------------+------------+-------------------+---------------------------+---------------------------+---------------------------+---------+---------------------------+
|   Route-distinguisher    |   Tag-ID   |    MAC-address    |        IP-address         |         Next-Hop          |            MED            | LocPref |           Path            |
+==========================+============+===================+===========================+===========================+===========================+=========+===========================+
| 10.0.0.1:111             | 0          | 00:C1:AB:00:00:11 | 0.0.0.0                   | 10.0.0.1                  | -                         | 100     |                           |
| 10.0.0.1:111             | 0          | 00:C1:AB:00:00:12 | 0.0.0.0                   | 10.0.0.1                  | -                         | 100     |                           |
| 10.0.0.1:111             | 0          | 00:C1:AB:00:00:13 | 0.0.0.0                   | 10.0.0.1                  | -                         | 100     |                           |
| 10.0.0.1:111             | 0          | 00:C1:AB:00:00:14 | 0.0.0.0                   | 10.0.0.1                  | -                         | 100     |                           |
+--------------------------+------------+-------------------+---------------------------+---------------------------+---------------------------+---------+---------------------------+
----------------------------------------------------------------------------------------------------
```

```
Type 3 Inclusive Multicast Ethernet Tag Routes
+-----------------------+--------+--------------+------------+-----+---------+------+
|  Route-distinguisher  | Tag-ID | Originator-IP| Next-Hop   | MED | LocPref | Path |
+=======================+========+==============+============+=====+=========+======+
| 10.0.0.1:111          | 0      | 10.0.0.1     | 10.0.0.1   | -   | 100     |      |
+-----------------------+--------+--------------+------------+-----+---------+------+
-------------------------------------------------------------------------------------
Type 4 Ethernet Segment Routes
+---------------------+-------------------------------+----------------+------------+-----+---------+------+
| Route-distinguisher |              ESI              | Originating-IP | Next-Hop   | MED | LocPref | Path |
+=====================+===============================+================+============+=====+=========+======+
| 10.0.0.1:0          | 01:11:11:11:11:11:11:00:00:01 | 10.0.0.1       | 10.0.0.1   | -   | 100     |      |
| 10.0.0.1:0          | 01:11:11:11:11:11:11:00:00:02 | 10.0.0.1       | 10.0.0.1   | -   | 100     |      |
| 10.0.0.1:0          | 01:11:11:11:11:11:11:00:00:03 | 10.0.0.1       | 10.0.0.1   | -   | 100     |      |
| 10.0.0.1:0          | 01:11:11:11:11:11:11:00:00:04 | 10.0.0.1       | 10.0.0.1   | -   | 100     |      |
+---------------------+-------------------------------+----------------+------------+-----+---------+------+
-------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------
8 advertised Ethernet Auto-Discovery routes
4 advertised MAC-IP Advertisement routes
1 advertised Inclusive Multicast Ethernet Tag routes
4 advertised Ethernet Segment routes
0 advertised IP Prefix routes
-------------------------------------------------------------------------------------
--{ candidate shared default }--[  ]--
```

```
A:leaf2# show network-instance default protocols bgp neighbor 10.0.1.1 advertised-routes evpn
-------------------------------------------------------------------------------------
Peer        : 10.0.1.1, remote AS: 501, local AS: 501
Type        : static
Description : None
Group       : iBGP-overlay
-------------------------------------------------------------------------------------
Origin codes: i=IGP, e=EGP, ?=incomplete
-------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------
Type 1 Ethernet Auto-Discovery Routes
+---------------------+-------------------------------+------------+------------+-----+---------+------+
| Route-distinguisher |              ESI              |   Tag-ID   | Next-Hop   | MED | LocPref | Path |
+=====================+===============================+============+============+=====+=========+======+
| 10.0.0.2:111        | 01:11:11:11:11:11:11:00:00:01 | 0          | 10.0.0.2   | -   | 100     |      |
| 10.0.0.2:111        | 01:11:11:11:11:11:11:00:00:01 | 4294967295 | 10.0.0.2   | -   | 100     |      |
| 10.0.0.2:111        | 01:11:11:11:11:11:11:00:00:02 | 0          | 10.0.0.2   | -   | 100     |      |
| 10.0.0.2:111        | 01:11:11:11:11:11:11:00:00:02 | 4294967295 | 10.0.0.2   | -   | 100     |      |
| 10.0.0.2:111        | 01:11:11:11:11:11:11:00:00:03 | 0          | 10.0.0.2   | -   | 100     |      |
| 10.0.0.2:111        | 01:11:11:11:11:11:11:00:00:03 | 4294967295 | 10.0.0.2   | -   | 100     |      |
| 10.0.0.2:111        | 01:11:11:11:11:11:11:00:00:04 | 0          | 10.0.0.2   | -   | 100     |      |
| 10.0.0.2:111        | 01:11:11:11:11:11:11:00:00:04 | 4294967295 | 10.0.0.2   | -   | 100     |      |
+---------------------+-------------------------------+------------+------------+-----+---------+------+
Type 3 Inclusive Multicast Ethernet Tag Routes
+-----------------------+--------+--------------+------------+-----+---------+------+
|  Route-distinguisher  | Tag-ID | Originator-IP| Next-Hop   | MED | LocPref | Path |
+=======================+========+==============+============+=====+=========+======+
| 10.0.0.2:111          | 0      | 10.0.0.2     | 10.0.0.2   | -   | 100     |      |
+-----------------------+--------+--------------+------------+-----+---------+------+
-------------------------------------------------------------------------------------
Type 4 Ethernet Segment Routes
+---------------------+-------------------------------+----------------+------------+-----+---------+------+
| Route-distinguisher |              ESI              | Originating-IP | Next-Hop   | MED | LocPref | Path |
+=====================+===============================+================+============+=====+=========+======+
| 10.0.0.2:0          | 01:11:11:11:11:11:11:00:00:01 | 10.0.0.2       | 10.0.0.2   | -   | 100     |      |
| 10.0.0.2:0          | 01:11:11:11:11:11:11:00:00:02 | 10.0.0.2       | 10.0.0.2   | -   | 100     |      |
| 10.0.0.2:0          | 01:11:11:11:11:11:11:00:00:03 | 10.0.0.2       | 10.0.0.2   | -   | 100     |      |
| 10.0.0.2:0          | 01:11:11:11:11:11:11:00:00:04 | 10.0.0.2       | 10.0.0.2   | -   | 100     |      |
+---------------------+-------------------------------+----------------+------------+-----+---------+------+
-------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------
8 advertised Ethernet Auto-Discovery routes
```

RT1, RT3 and RT4 routes are triggered by configuration (ES and MAC-VRF), while RT2 routes (MAC-IP) only appear when a MAC is learned or statically configured.

Among these, RT4 is known as ES routes imported by ES peers for DF election and local biasing (split-horizon). It is advertised/received here only by leaf1 and leaf2.

RT1 also advertises ESIs, mainly for two reasons (hence two entries per ESI):

- Aliasing for load balancing (0)
- Mass withdrawal for fast convergence (4294967295)

BGP EVPN route table:

show network-instance default protocols bgp routes evpn route-type summary

```
A:leaf1# show network-instance default protocols bgp routes evpn route-type summary
-----------------------------------------------------------------------------------------
Show report for the BGP route table of network-instance "default"
-----------------------------------------------------------------------------------------
Status codes: u=used, *=valid, >=best, x=stale
Origin codes: i=IGP, e=EGP, ?=incomplete
-----------------------------------------------------------------------------------------
BGP Router ID: 10.0.0.1      AS: 101      Local AS: 101
-----------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------
Type 1 Ethernet Auto-Discovery Routes
+--------+----------------------+-------------------------------+------------+-----------+-----------+-----+
| Status | Route-distinguisher  |              ESI              |   Tag-ID   | neighbor  | Next-hop  | VNI |
+========+======================+===============================+============+===========+===========+=====+
| u*>    | 10.0.0.2:111         | 01:11:11:11:11:11:11:00:00:01 | 0          | 10.0.0.2  | 10.0.0.2  | 1   |
| u*>    | 10.0.0.2:111         | 01:11:11:11:11:11:11:00:00:01 | 4294967295 | 10.0.0.2  | 10.0.0.2  | -   |
| u*>    | 10.0.0.2:111         | 01:11:11:11:11:11:11:00:00:02 | 0          | 10.0.0.2  | 10.0.0.2  | 1   |
| u*>    | 10.0.0.2:111         | 01:11:11:11:11:11:11:00:00:02 | 4294967295 | 10.0.0.2  | 10.0.0.2  | -   |
| u*>    | 10.0.0.2:111         | 01:11:11:11:11:11:11:00:00:03 | 0          | 10.0.0.2  | 10.0.0.2  | 1   |
| u*>    | 10.0.0.2:111         | 01:11:11:11:11:11:11:00:00:03 | 4294967295 | 10.0.0.2  | 10.0.0.2  | -   |
| u*>    | 10.0.0.2:111         | 01:11:11:11:11:11:11:00:00:04 | 0          | 10.0.0.2  | 10.0.0.2  | 1   |
| u*>    | 10.0.0.2:111         | 01:11:11:11:11:11:11:00:00:04 | 4294967295 | 10.0.0.2  | 10.0.0.2  | -   |
+--------+----------------------+-------------------------------+------------+-----------+-----------+-----+
Type 3 Inclusive Multicast Ethernet Tag Routes
+--------+----------------------+--------+---------------+-----------+-----------+
| Status | Route-distinguisher  | Tag-ID | Originator-IP | neighbor  | Next-Hop  |
+========+======================+========+===============+===========+===========+
| u*>    | 10.0.0.2:111         | 0      | 10.0.0.2      | 10.0.0.2  | 10.0.0.2  |
+--------+----------------------+--------+---------------+-----------+-----------+
-----------------------------------------------------------------------------------------
Type 4 Ethernet Segment Routes
+--------+----------------------+-------------------------------+-------------------+-----------+-----------+
| Status | Route-distinguisher  |              ESI              | originating-router| neighbor  | Next-Hop  |
+========+======================+===============================+===================+===========+===========+
| u*>    | 10.0.0.2:0           | 01:11:11:11:11:11:11:00:00:01 | 10.0.0.2          | 10.0.0.2  | 10.0.0.2  |
| u*>    | 10.0.0.2:0           | 01:11:11:11:11:11:11:00:00:02 | 10.0.0.2          | 10.0.0.2  | 10.0.0.2  |
| u*>    | 10.0.0.2:0           | 01:11:11:11:11:11:11:00:00:03 | 10.0.0.2          | 10.0.0.2  | 10.0.0.2  |
| u*>    | 10.0.0.2:0           | 01:11:11:11:11:11:11:00:00:04 | 10.0.0.2          | 10.0.0.2  | 10.0.0.2  |
+--------+----------------------+-------------------------------+-------------------+-----------+-----------+
-----------------------------------------------------------------------------------------
8 Ethernet Auto-Discovery routes 8 used, 8 valid
0 MAC-IP Advertisement routes 0 used, 0 valid
1 Inclusive Multicast Ethernet Tag routes 1 used, 1 valid
4 Ethernet Segment routes 4 used, 4 valid
0 IP Prefix routes 0 used, 0 valid
-----------------------------------------------------------------------------------------
--{ running }--[  ]--
A:leaf1#
```

MAC Table:

show network-instance mac-vrf-1 bridge-table mac-table all

```
A:leaf1# show network-instance mac-vrf-1 bridge-table mac-table all
-----------------------------------------------------------------------------------------
Mac-table of network instance mac-vrf-1
-----------------------------------------------------------------------------------------
+-------------------+-------------+------------+---------+--------+-------+--------------------------+
|      Address      | Destination | Dest Index |  Type   | Active | Aging |       Last Update        |
+===================+=============+============+=========+========+=======+==========================+
| 00:C1:AB:00:00:11 | lag1.0      | 5          | learnt  | true   | 285   | 2024-11-16T18:50:01.000Z |
| 00:C1:AB:00:00:12 | lag2.0      | 9          | learnt  | true   | 285   | 2024-11-16T17:39:15.000Z |
| 00:C1:AB:00:00:13 | lag3.0      | 7          | learnt  | true   | 285   | 2024-11-16T18:50:01.000Z |
| 00:C1:AB:00:00:14 | lag4.0      | 10         | learnt  | true   | 285   | 2024-11-16T17:39:15.000Z |
+-------------------+-------------+------------+---------+--------+-------+--------------------------+
Total Irb Macs              :    0 Total    0 Active
Total Static Macs           :    0 Total    0 Active
Total Duplicate Macs        :    0 Total    0 Active
Total Learnt Macs           :    4 Total    4 Active
Total Evpn Macs             :    0 Total    0 Active
Total Evpn static Macs      :    0 Total    0 Active
Total Irb anycast Macs      :    0 Total    0 Active
Total Proxy Antispoof Macs  :    0 Total    0 Active
Total Reserved Macs         :    0 Total    0 Active
Total Eth-cfm Macs          :    0 Total    0 Active
--{ candidate shared default }--[  ]--
A:leaf1#
```

```
A:leaf2# show network-instance mac-vrf-1 bridge-table mac-table all
-----------------------------------------------------------------------------------------
Mac-table of network instance mac-vrf-1
-----------------------------------------------------------------------------------------
+-------------------+-------------+------------+---------+--------+-------+--------------------------+
|      Address      | Destination | Dest Index |  Type   | Active | Aging |       Last Update        |
+===================+=============+============+=========+========+=======+==========================+
| 00:C1:AB:00:00:11 | lag1.0      | 5          | evpn    | true   | N/A   | 2024-11-16T18:50:01.000Z |
| 00:C1:AB:00:00:12 | lag2.0      | 9          | evpn    | true   | N/A   | 2024-11-16T17:40:24.000Z |
| 00:C1:AB:00:00:13 | lag3.0      | 7          | evpn    | true   | N/A   | 2024-11-16T18:50:01.000Z |
| 00:C1:AB:00:00:14 | lag4.0      | 10         | evpn    | true   | N/A   | 2024-11-16T17:40:24.000Z |
+-------------------+-------------+------------+---------+--------+-------+--------------------------+
Total Irb Macs              :    0 Total    0 Active
Total Static Macs           :    0 Total    0 Active
Total Duplicate Macs        :    0 Total    0 Active
Total Learnt Macs           :    0 Total    0 Active
Total Evpn Macs             :    4 Total    4 Active
Total Evpn static Macs      :    0 Total    0 Active
Total Irb anycast Macs      :    0 Total    0 Active
Total Proxy Antispoof Macs  :    0 Total    0 Active
Total Reserved Macs         :    0 Total    0 Active
Total Eth-cfm Macs          :    0 Total    0 Active
--{ candidate shared default }--[  ]--
A:leaf2#
```

ESI-based Load-Balancing:

MAC addresses learned through EVPN typically show the VTEP (PE) router in the destination column, while MAC addresses of multihomed devices are instead assigned the EVPN Segment Identifier (ESI), which can refer to multiple VTEP destinations.

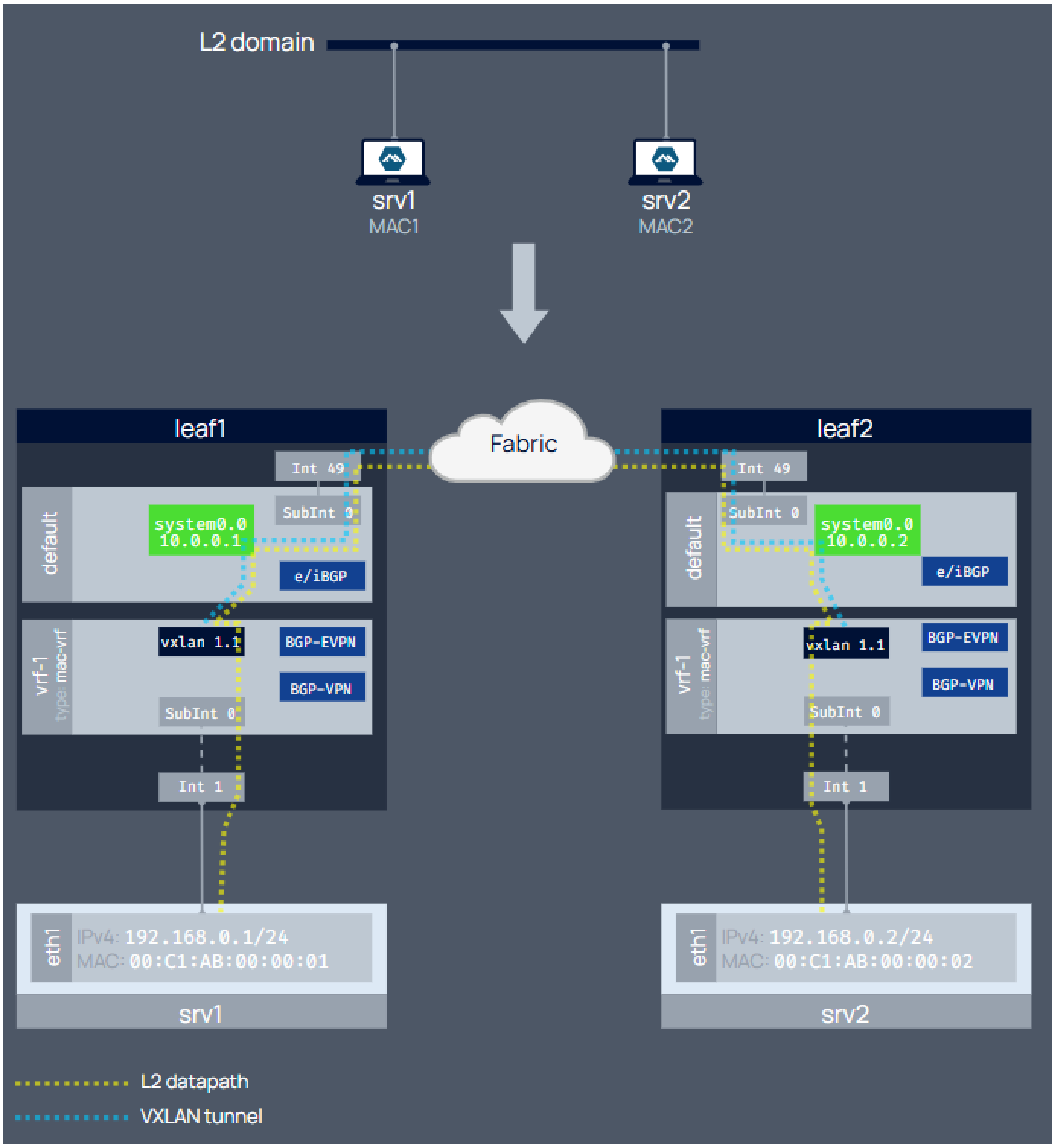
The use of ESIs here ensures the load balancing as it refers to multiple VTEPs if ECMP is enabled.

show tunnel-interface vxlan1 vxlan-interface 1 bridge-table unicast-destinations destination

show tunnel-interface vxlan1 vxlan-interface 1 bridge-table multicast-destinations destination

```
A:leaf1# show tunnel-interface vxlan1 vxlan-interface 1 bridge-table unicast-destinations destination

-------------------------------------------------------------------------------------------------------------------
Show report for vxlan-interface vxlan1.1 unicast destinations
-------------------------------------------------------------------------------------------------------------------
Destinations
-------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------
Ethernet Segment Destinations
-------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------------------
Summary
  0 unicast-destinations, 0 non-es, 0 es
  0 MAC addresses, 0 active, 0 non-active
-------------------------------------------------------------------------------------------------------------------
--{ candidate shared default }--[  ]--
A:leaf1# show tunnel-interface vxlan1 vxlan-interface 1 bridge-table multicast-destinations destination
-------------------------------------------------------------------------------------------------------------------
Show report for vxlan-interface vxlan1.1 multicast destinations (flooding-list)
-------------------------------------------------------------------------------------------------------------------
+--------------+------------+-------------------+----------------------+
| VTEP Address | Egress VNI | Destination-index | Multicast-forwarding |
+==============+============+===================+======================+
| 10.0.0.2     | 1          | 8349072465255     | BUM                  |
+--------------+------------+-------------------+----------------------+
-------------------------------------------------------------------------------------------------------------------
Summary
  1 multicast-destinations
-------------------------------------------------------------------------------------------------------------------
--{ candidate shared default }--[  ]--
A:leaf1#
```



The highly detailed configuration & verification steps helped us achieve the goal of creating an overlay Layer 2 broadcast domain for the two servers in our topology. So that the high level service diagram transformed into a detailed map of configuration constructs and instances.

During the verification phases we observed the following packet captures to prove the control/data plane behavior:

- Exchange of the IMET/RT3 EVPN routes. IMET/RT3 routes are the starting point in the L2 EVPN-VXLAN services, as they are used to dynamically discover the VXLAN VTEPs participating in the same EVI.

- Exchange of MAC-IP/RT2 EVPN routes which convey the MAC information of the attached servers. These routes are used to create unicast tunnel destinations that the dataplane frames will use.