Pluribus NETWORKS

Pluribus Data Center IP-Fabric

Validated Design & Deployment Guide

Table of Contents

Terminology Reference

This is a glossary of acronyms and terms used throughout this document.

Cluster	A pair of adjacent Netvisor ONE enabled switches acting as one logical unit for high availability typically used with Multi Chassis LAG also called vLAG
Fabric	A set of Netvisor ONE enabled switches that operate and are managed as a single entity
In-band Interface	A fabric communication interface in a Pluribus fabric built using an in-band (data) network
Insight Analytics	Insight Analytics is Network Performance Management (NPM) module as part of UNUM
L2VPN	Layer2 Virtual Private Network services in the overlay
L3VPN	Layer3 Virtual Private Network services in the overlay
Overlay	All the elements built on top of the generic IP transport infrastructure with packets carried inside a VxLAN encapsulation
UNUM	Pluribus UNUM™ Unified Management, Automation and Analytics Platform software
Underlay	In the VXLAN context, this refers to the generic IP transport infrastructure used to ensure IP communication among all data centers
vLAG	Virtual Link Aggregation Group is a Netvisor ONE technology for connecting multiple switches to other devices or to other switches for resiliency and high availability
vNET	A Virtual NETwork is a partition of the Adaptive Cloud Fabric. A vNET is defined by a group of network objects that can operate independently and have dedicated resources, providing multi- tenancy and network segmentation.
VRF	A distributed (Pluribus) implementation of Virtual Routing and Forwarding L3 tenant isolation, also called a dVRF
vRouter	An instance of a routing abstraction in a Pluribus Network Operating System. The vRouter runs in a dedicated operating system container

Abstract

This guide provides design and implementation guidance for enterprises, cloud and manage service provider networks.

The Pluribus Networks Validated Designs are a collection of technology focused guides that include Adaptive Cloud Fabric™ (ACF) technology, recommendations for designing, planning, configuration, and deployment best practices. Together these guides comprise a reference model for understanding Pluribus Adaptive Cloud technology and designs for common deployment scenarios. ACF network design is built and validated to address specific use cases and deployment scenarios. Network engineers can use these proven designs to rapidly deploy Pluribus solutions with the assurance that they will perform, and scale as expected.

Key Audience

This document is intended for network architects, network engineers, systems engineers, who want to understand how to deploy Pluribus IP-Fabric in data center infrastructure.

This design and deployment guide assume that the reader is familiar with traditional networking protocols.

Adaptive Cloud Fabric Overview

Pluribus Networks offers a unique and highly effective implementation of software-defined networking (SDN) called the Adaptive Cloud Fabric. This section summarizes important elements of the Adaptive Cloud Fabric architecture and capabilities that are relevant to building leaf-spine datacenter IP fabrics.

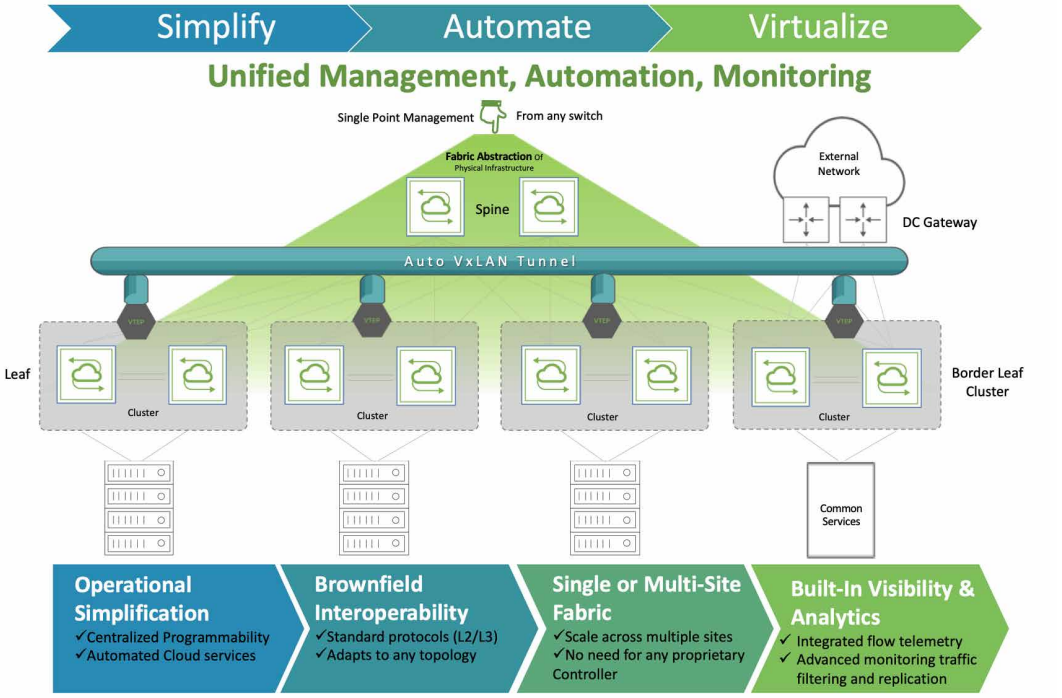
The foundation of the Adaptive Cloud Fabric is open networking switches running the Pluribus Netvisor® ONE network operating system (OS). Netvisor ONE software supports a variety of standard layer 2 (L2) and layer 3 (L3) protocols for flexibility in designing resilient underlay networks including interoperability with other network elements. A key attribute of the Netvisor ONE OS is the ability to support a variety of network topologies, including data center leaf-spine, aggregation tree and ring networks. Netvisor ONE also incorporates configuration automation and advanced traffic visibility based on built-in flow monitoring and telemetry.

Building on this foundation, the Adaptive Cloud Fabric software enables creation and automation of distributed network fabrics thanks to several important technologies:

- **Distributed, controllerless architecture:** the Adaptive Cloud Fabric software runs directly on open networking switches and implements an innovative distributed control plane and distributed database, removing the need for an external controller. Every fabric node has full network state information and can act as a single point of management for the entire fabric. An advanced transactional model guarantees that device configuration is maintained consistently across network nodes and supports configuration rollback capabilities. The operational efficiency benefits of this controllerless architecture are even more pronounced in highly geo-distributed networks.
- **Network virtualization and segmentation:** The Adaptive Cloud Fabric creates highly automated virtualized overlay networks based on standard Virtual Extensible LAN (VXLAN) tunnels with high availability (HA) configuration, switch-based tunnel termination and hardware acceleration for resilient, wirespeed performance. This overlay approach supports the automated creation of rich virtualized services and enables multi-tenant network segmentation for both physical and virtual devices and applications.
- **Virtualized services:** The Adaptive Cloud Fabric implements a range of powerful virtualized services as fabric-wide objects that are simple to configure and manage. Examples include layer 1 emulated virtual links, multipoint layer 2 bridge domains and distributed virtual routing and forwarding instances (VRFs) with anycast gateways for efficient distributed routing. These service types can be mixed within a fabric, enabling a single network to efficiently support a wide variety of users and applications.

- **Fabric-wide network and service automation:** Network, service and policy configuration and provisioning are dramatically simplified using fabric-wide service objects and commands and built-in automation, allowing a single command to configure the entire network. One important example of built-in automation is automatic VXLAN tunnel creation to enable a newly provisioned service.

Figure 1:
Key benefits and foundational pillars of Adaptive Cloud Fabric



Organization of This Document

The IP-Fabric guide focuses on aspects of Pluribus technologies and deployment models. The ACF IP-Fabric is structured in four separate phases:

Planning Phase – explains the core technologies of an ACF network and describes different aspects of planning, Implementation, and best-practice deployments.

Underlay Implementation Phase – discusses deployments in conditions that differ from the common base design deployment models, with respect to scaling-out while maintaining reliability and flexibility.

Virtualized Services (Overlay) Implementation Phase – builds on the base designs delivering mission-critical application specific design and deployment content for services like Virtualized L2/3-VNI, Bridge-Domain, Security, Multi-tenant services, monitoring, and visibility.

Pluribus IP-Fabric Design Objectives

When building an IP fabric, the main design objectives considered are:

- Achieving a scale-out, modular architecture easy to grow and expand
- Non-blocking IP fabric using standard IP protocols
- High availability with fast re-convergence in case of a single or multiple failure events
- Highly scalable Layer 2 VPN and Layer 3 VPN segmentation services
- Enable control-plane traffic protection to protect the fabric CPU from mis-behaving endpoints or security threats such as DDoS attacks
- Forwarding efficiency for traffic hair pinning avoidance, loop avoidance, BUM optimization
- Operational simplification with a single unified management plane for the entire fabric
- End-to-end visibility of network and application traffic

Planning Phase

The design phase is critical to ensure the architecture will deliver a robust solution fulfilling all the requirements in terms of functionality, redundancy, and bandwidth availability.

The design phase is also intended to be a dynamic iterative process. For instance, a specific deployment scenario could hit a limitation or constraint, such as port availability, which could trigger a re-evaluation of previous higher-level design decisions.

Using a top-down approach, the design steps are ordered hierarchically from the end goals of the solution down to the details of logical and physical connectivity.

Figure 2:
Steps in building an Adaptive Cloud Fabric

Step 1	Design the Scale-Out Leaf and Spine Architecture for the Data Center
Step 2	Plan the IP-Fabric layout and identify the Layer 2 extension requirements
Step 3	Plan the "Underlay" connectivity (Layer 3 » Layer 1 Architecture)
Step 4	Re-evaluate / validate the previous designs based on eventual per-case limitations

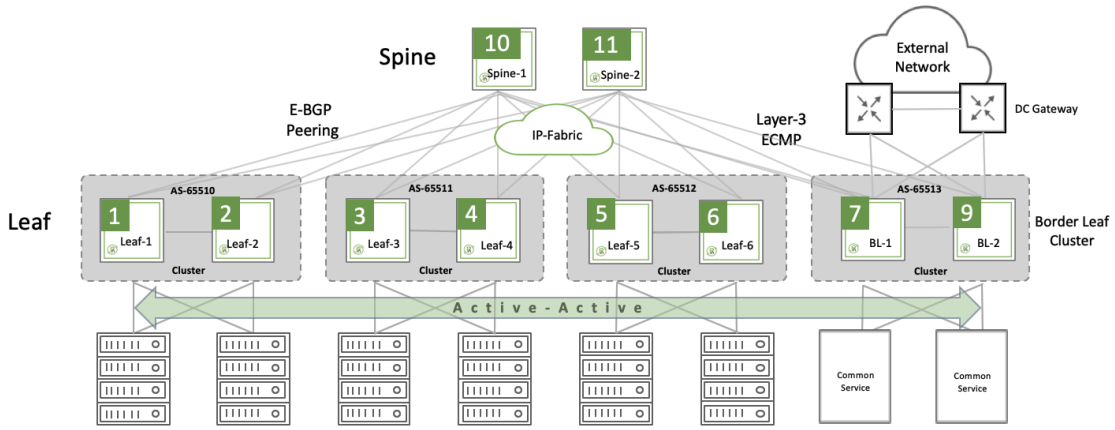
Step 1 – Design the High-level Data Center IP-Fabric Architecture

Figure 3 below depicts the high-level representation of the set-up that it is going to be used throughout this document as an example of the Pluribus IP-Fabric implementation.

There are eight Leaf switches and two Spine switches in the data center, connected through a mix of 10/40/100G redundant links. There are two Border Leaf switches (BL-1 & BL-2) connected to the spine layer and to the external DC Gateway for internal and external connectivity.

Switches powered by Netvisor ONE (green icons) act as Top of Rack (ToR) switches, providing connectivity to hosts, VMs, or containers. Synthetic traffic generator ports may also use to emulate these endpoint devices for validation purposes.

Figure 3:
Pluribus IP-Fabric CLOS Architecture Overview

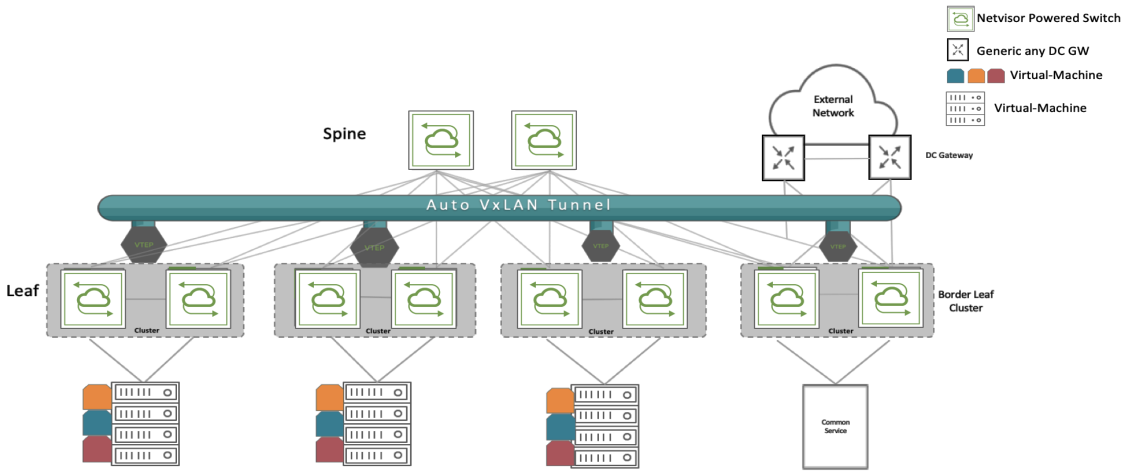


Step 2 – Plan the IP-Fabric layout and identify the overlay services

In Figure 3, all the switches in the design will be part of the same Adaptive Cloud Fabric instance, thus bringing to the network operator benefits such as end-to-end visibility, single-point-of-management, security, and automation.

In Figure 4, the VMs located on the host servers, are represented in different colors as they belong to different logical networks stretched across host/racks. In the rest of the document these logical networks will be often referred to as overlays. This planning stage focuses on identifying these overlay services, how they are distributed physically and which isolation/communication policies they comply with.

Figure 4:
Virtualized Overlay Networking Over IP Underlay Network



Step 3 – Plan the underlay connectivity (Layer 3 » Layer 1 Architecture)

The fabric underlay refers to the physical and logical connectivity between the switches. An underlay transport network needs to be present to exchange data between any two endpoints (for example VMs) that are in different racks. The fabric control-plane, when deployed in-band, also relies on the same underlay transport.

This step lays out the physical connectivity of the Clos architecture and defines the VLAN and IP address schema used for switch-to-switch communication, including the routing protocols chosen for this design.

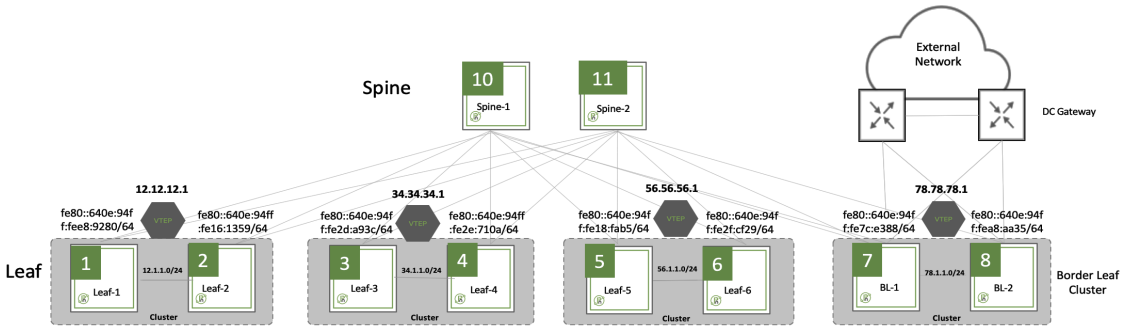
Pluribus switches support layer-3 interfaces (optionally) implemented with BGP unnumbered: as such, they don't require to be provisioned with an IPv4 address, because BGP unnumbered automatically advertises a link-local IPv6 address as a next-hop. This approach has the dual advantage of reducing the provisioning effort complexity and saving IPv4 address space.

The fabric underlay is a private IPv4 based network transport designed for internal fabric connectivity. The underlay is in fact completely isolated from the overlay network services that provide connectivity to devices attached to the fabric. As first benefit of this decoupled underlay-overlay approach, the underlay IPv4 prefixes may overlap with those used for endpoint (overlay) services and in general with networks outside the fabric. The second advantage is the fact that IPv6 endpoints are also supported on the overlay networks without the need to configure IPv6 in the underlay.

The comprehensive IP address plan used in the topology has the following attributes:

- For simplicity all prefix masks used are either /24 for subnets or /32 for loopback addresses
- Each device has its own ID (represented in the white circle)
- Each link between two adjacent leaf switches uses a subnet according to the following IP address convention:
  - < SW1_ID><SW2_ID>.1.1.0/24, the last octet of the IP is either <SW1_ID> or <SW2_ID>
- Each point-to-point link between a leaf and spine switch is implemented using eBGP unnumbered: as such, there is no need to define a specific subnet for it. The design can also cover situations where eBGP unnumbered may not be an option: in that case, each subnet for interconnecting a leaf to a spine, or in general any two Layer 3 speakers, may use the same IP address convention at previous point.
- Each device uses three local interfaces:
  - Loopback interface (used for routing): <SW_ID>.<SW_ID>.<SW_ID>.>.<SW_ID>/32
  - In-band interface (used to fabric in-band communication): 1.1.<SW_ID>.1/ 24
  - VTEP interface (used for VXLAN tunnel termination): <SW1SW2_IDs>.<SW1SW2_IDs>.<SW1SW2_IDs>.<SW_ID>/24
- VRRP: a Virtual IP (VIP) is used in each subnet to provide redundancy to the devices connected using <SW1SW2_IDs> as the first 3 octet for the VIP

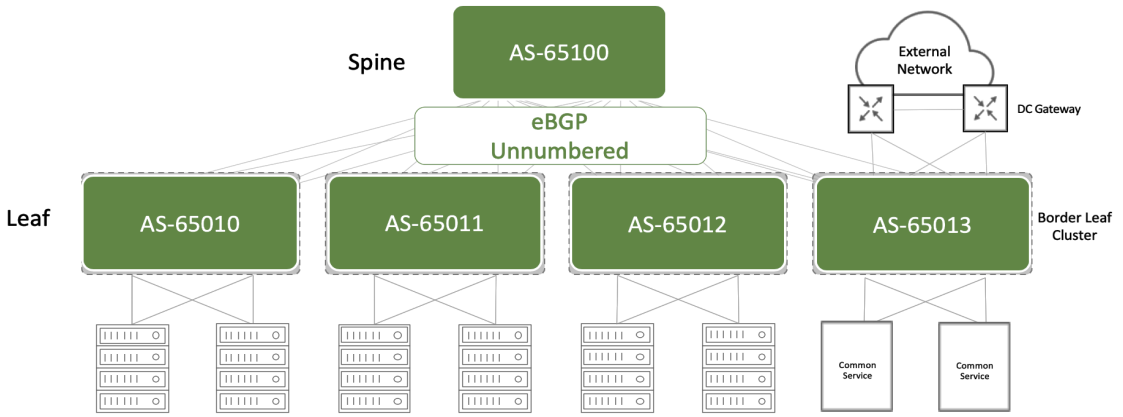
Figure 5:
Underlay IP Addressing



The details on the AS numbering are presented in Figure 6 below. The design includes the use of ECMP to load-balance the traffic on all the available connections within the setup. BFD is also enabled to enhance the failover convergence times in case of a link-down/node-down event.

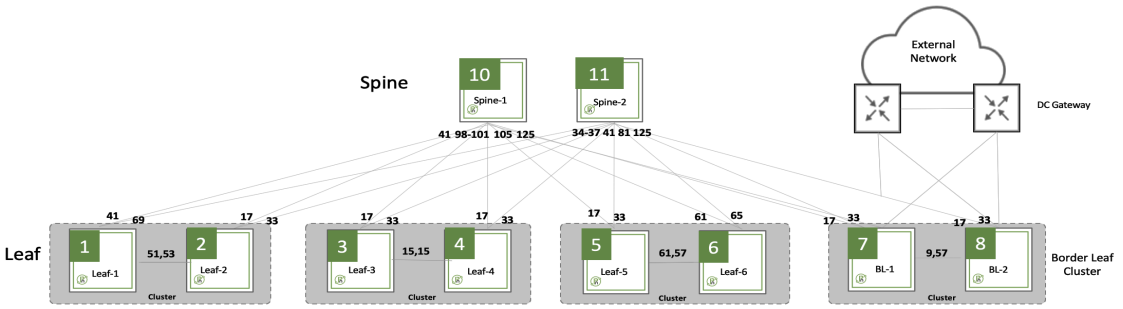
NOTE: The use of BGP is not mandatory; OSPF or Static Routes could be also used if the IP reachability among the fabric nodes is ensured. BGP offers more scalability and flexibility and thus is the preferred option for this design guide.

Figure 6:
The AS numbering of the BGP underlay



From a Layer 1 perspective, the physical connectivity and the detailed port numbers used in the current setup are documented in Figure 7:

Figure 7:
IP-Fabric physical port connectivity details



Step 4 – Re-evaluate and re-validate the previous steps based on any possible limitations

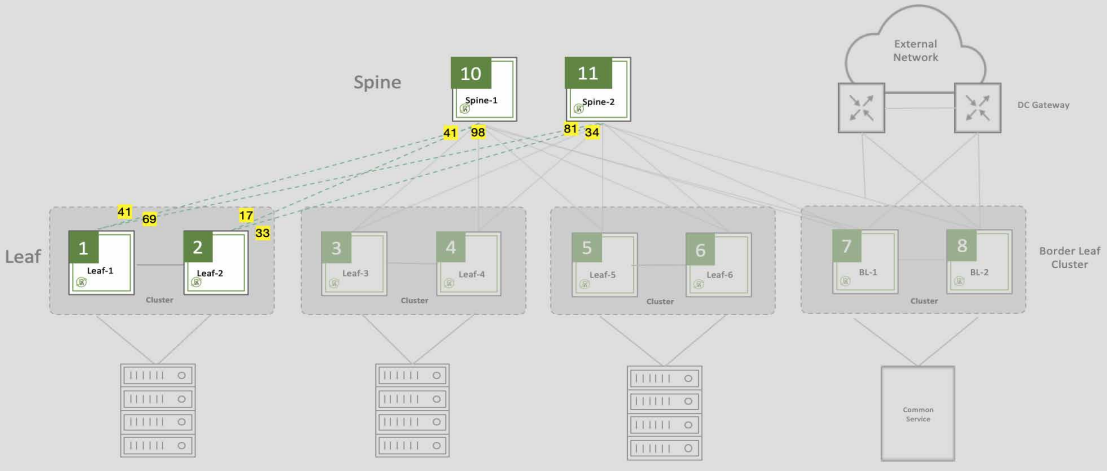
As mentioned above, the design phase is usually an iterative process. Once all design steps are complete, it is a good time to revisit the design goals of the architecture and make sure all the requirements are fulfilled. The sooner the need for a change in the design gets identified, the easier is to manage and mitigate its implications.

Underlay Implementation Phase

The IP Fabric in Figure 8 uses spine and leaf three-stage CLOS architecture (leaf switches as ingress+egress stages and spine switches as a middle stage); all the switches in Figure 8 are running Pluribus Netvisor operating system except for the DC Gateway devices.

In this configuration, spine devices are typically Layer 3 switches that provide the underlay IP connectivity between leaf devices, and leaf devices are top-of-rack (TOR) switches that provide connectivity to the servers, storage device and other common service devices in the data center.

Figure 8:
Three-Stage CLOS based IP-Fabric underlay network



Configuring the underlay network involves the configuration of (at least) the following base features:

Layer-2	VLAN	Leafs for SVI
Layer-3	VRRP	VRRP between Leaf Nodes for VTEP-HA
	BGP & BFD	iBGP between Leafs (Cluster) and eBGP between Leaf and Spine
Cluster	Multi-Homing	Between Leaf Nodes
vLAG	Active-Active Path	Between Leaf Nodes and Connected Host

Step 1 - Enable physical port, speed, and adjust the MTU for jumbo frames

After plugging in the cables, login to each switch and use the following command to enable port, speed, and increase the MTU value to jumbo.

NOTE: VXLAN encapsulation adds between 50 and 54 bytes of additional header information to the original Ethernet frame. Because this can result in Ethernet frames that exceed the default 1514-byte MTU, best practice is to implement jumbo frames throughout the network.

Follow the configuration steps as shown below

Spine-1

```
CLI (network-admin@Spine-1) > port-config-modify port 41 speed 100g enable jumbo
CLI (network-admin@Spine-1) > port-config-modify port 98 speed 100g enable jumbo
```

Spine-2

```
CLI (network-admin@Spine-2) > port-config-modify port 81 speed 100g enable jumbo
CLI (network-admin@Spine-2) > port-config-modify port 34 speed 100g enable jumbo
```

Leaf-1

```
CLI (network-admin@Leaf-1) > port-config-modify port 41 speed 100g enable jumbo
CLI (network-admin@Leaf-1) > port-config-modify port 69 speed 100g enable jumbo
```

Leaf-2

```
CLI (network-admin@Leaf-2) > port-config-modify port 17 speed 100g enable jumbo
CLI (network-admin@Leaf-2) > port-config-modify port 33 speed 100g enable jumbo
```

Verify physical port information at Layer 2 using the port-phy-show command. The command displays information about the default VLAN, link quality, maximum frame size, Ethernet mode, speed, and status.

Spine-1

```
CLI (network-admin@Spine-1) > port-phy-show port 41,98
port state speed eth-mode max-frame learning def-vlan
---- ----- ----- -------- --------- -------- --------
41   up    100000 kr4      9412      off      4092
98   up    100000 kr4    9412      off      4091
```

Spine-2

```
CLI (network-admin@Spine-2) > port-phy-show port 81,34
port state speed eth-mode max-frame learning def-vlan
---- ----- ----- -------- --------- -------- --------
34   up    100000 kr4   9412      off      4091
81   up    100000 kr4     9412      off      4092
```

Leaf-1

```
CLI (network-admin@Leaf-1) > port-phy-show port 41,69
port state speed eth-mode max-frame learning def-vlan
---- ----- ----- -------- --------- -------- --------
41   up    100000 kr4      9412      off      4092
69   up    100000 kr4      9412      off      4091
```

Leaf-2

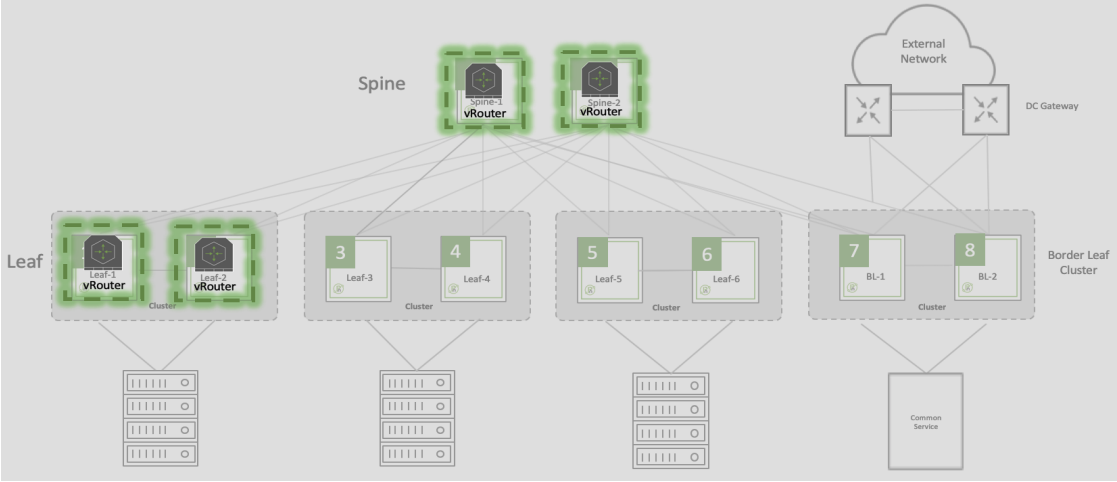
```
CLI (network-admin@Leaf-2) > port-phy-show port 17,33
port state speed eth-mode max-frame learning def-vlan
---- ----- ----- -------- --------- -------- --------
17   up    100000 kr4      9412      off      4092
33   up    100000 kr4      9412      off      4091
```

Step 2: Create a vRouter for fabric communication

Create a vRouter for fabric communication and configure all the required interfaces and ensure IP reachability across the leaf-spine eBGP fabric.

A vRouter runs in a dedicated container on each switch and provides static and dynamic routing capabilities. The vRouters also provide connectivity among the in-band management interface of the switches to enable the SDN Fabric control plane communication.

Figure 9:
Create vRouter and attach all the required interfaces



Creating a vRouter with the specific attribute `fabric-comm`, signals to Netvisor to use this router for fabric control plane communication.

To create vRouter on a spine and a leaf, use the following command

Spine-1

```
CLI (network-admin@Spine-1) > vrouter-create name Spine-1 fabric-comm
```

Spine-2

```
CLI (network-admin@Spine-2) > vrouter-create name Spine-2 fabric-comm
```

Leaf-1

```
CLI (network-admin@Leaf-1) > vrouter-create name Leaf-1 fabric-comm
```

Leaf-2

```
CLI (network-admin@Leaf-2) > vrouter-create name Leaf-2 fabric-comm
```

Verify the vRouter configuration, use the vrouter-show or vrouter-show format name, type, scope, state, router command:

Spine-1

```
CLI (network-admin@Spine-1) > vrouter-show format name,type,scope,state,
router-type,hw-router-mac
name    type    scope state   router-type hw-router-mac
------- ------- ----- ------- ----------- -----------------
Spine-1 vrouter local enabled hardware    66:0e:94:17:61:93
```

Spine-2

```
CLI (network-admin@Spine-2) > vrouter-show format name,type,scope,state,
router-type,hw-router-mac
name    type    scope state   router-type hw-router-mac
------- ------- ----- ------- ----------- -----------------
Spine-2 vrouter local enabled hardware    66:0e:94:30:83:94
```

Leaf-1

```
CLI (network-admin@Leaf-1) > vrouter-show format name,type,scope,state,
router-type,hw-router-mac
name   type    scope state   router-type hw-router-mac
------ ------- ----- ------- ----------- -----------------
Leaf-1 vrouter local enabled hardware    66:0e:94:e8:92:80
```

Leaf-2

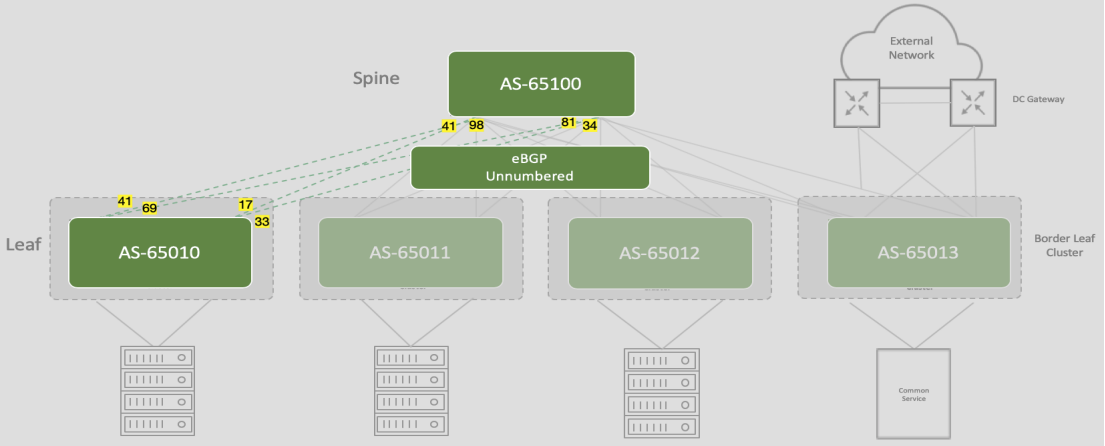
```
CLI (network-admin@Leaf-2) > vrouter-show format name,type,scope,state,
router-type,hw-router-mac
name   type    scope state   router-type hw-router-mac
------ ------- ----- ------- ----------- -----------------
Leaf-2 vrouter local enabled hardware    66:0e:94:16:13:59
```

Step 3 – Enable BGP routing and BFD protocol

To minimize the IP address space assigned on each link between leaf and spine, Netvisor ONE supports BGP Unnumbered. BGP unnumbered advertises IPv4 network layer reachability Information with an IPv6 link-local address as next-hop.

In this design, switches in the spine layer share the same autonomous system number (ASN); also leaf switches part of the same cluster share the same ASN; each leaf cluster uses a unique ASN. As such, leaf and spine use eBGP and switches part of the same cluster use iBGP.

Figure 10:
Enable BGP and BFD with unique ASN



First create an interface with a link-local IPv6 address between Spine-1 and each Leaf switch:

```
CLI (network-admin@Spine-1) > vrouter-interface-add vrouter-name Spine-1 l3-port
41 ipv6-unnumbered

CLI (network-admin@Spine-1) > vrouter-interface-add vrouter-name Spine-1 l3-port
98 ipv6-unnumbered
```

Modify the vRouter BGP configuration to enable ECMP load balancing across all the available connections between leaf and spine.

```
CLI (network-admin@Spine-1) > vrouter-modify name Spine-1 bgp-as 65100 router-id
100.1.1.1 bgp-redistribute connected bgp-max-paths 16 bgp-bestpath-as-path
multipath-relax
```

Configure the eBGP unnumbered adjacencies on the first spine switch towards the leaf switches and for fast convergence enable BFD.

```
CLI (network-admin@Spine-1) > vrouter-bgp-add vrouter-name Spine-1 remote-as
external l3-port 41 bfd

CLI (network-admin@Spine-1) > vrouter-bgp-add vrouter-name Spine-1 remote-as
external l3-port 98 bfd
```

Repeat the configuration steps for Spine-2:

```
CLI (network-admin@Spine-2) > vrouter-interface-add vrouter-name Spine-2 l3-port
81 ipv6-unnumbered

CLI (network-admin@Spine-2) > vrouter-interface-add vrouter-name Spine-2 l3-port
34 ipv6-unnumbered

CLI (network-admin@Spine-2) > vrouter-modify name Spine-2 bgp-as 65100 router- id
200.1.1.1 bgp-redistribute connected bgp-max-paths 16 bgp-bestpath-as-path
multipath-relax

CLI (network-admin@Spine-2) > vrouter-bgp-add vrouter-name Spine-2 remote-as
external l3-port 81 bfd

CLI (network-admin@Spine-2) > vrouter-bgp-add vrouter-name Spine-2 remote-as
external l3-port 34 bfd
```

Next, we start configuring the leaf switches starting with Leaf-1. As each leaf pair use a different ASN, there is no need to configure allow-AS-in property for the eBGP adjacency.

```
CLI (network-admin@Leaf-1) > vrouter-interface-add vrouter-name Leaf-1 l3-port 41
ipv6-unnumbered

CLI (network-admin@Leaf-1) > vrouter-interface-add vrouter-name Leaf-1 l3-port 69
ipv6-unnumbered

CLI (network-admin@Leaf-1) > vrouter-modify name Leaf-1 bgp-as 65010 router-id
1.1.1.1 bgp-redistribute connected bgp-max-paths 16 bgp-bestpath-as-path
multipath-relax

CLI (network-admin@Leaf-1) vrouter-bgp-add vrouter-name Leaf-1 remote-as external
l3-port 41 bfd

CLI (network-admin@Leaf-1) vrouter-bgp-add vrouter-name Leaf-1 remote-as external
l3-port 69 bfd
```

Repeat the configuration steps for Leaf-2:

```
CLI (network-admin@Leaf-2) > vrouter-interface-add vrouter-name Leaf-2 l3-port 17
ipv6-unnumbered

CLI (network-admin@Leaf-2) > vrouter-interface-add vrouter-name Leaf-2 l3-port 33
ipv6-unnumbered

CLI (network-admin@Leaf-2) > vrouter-modify name Leaf-2 bgp-as 65010 router-id
2.2.2.2 bgp-redistribute connected bgp-max-paths 16 bgp-bestpath-as-path
multipath-relax

CLI (network-admin@Leaf-2) > vrouter-bgp-add vrouter-name Leaf-2 remote-as
external l3-port 17 bfd

CLI (network-admin@Leaf-2) > vrouter-bgp-add vrouter-name Leaf-2 remote-as
external l3-port 33 bfd
```

Verify the vRouter port configuration, use the vrouter-interface show or vrouter-show l3-port command:

Spine-1

```
CLI (network-admin@Spine-1) > vrouter-interface-show l3-port 41,98
vrouter-name nic       ip                          linklocal
mac               vlan vlan-type nic-state l3-port mtu  priority-tag
------------ --------- --------------------------- -------------------------
---------------- ---- --------- --------- ------- ---- ------------
Spine-1      eth0.4092 fe80::640e:94ff:fe17:6193/64 fe80::640e:94ff:fe17:6193
 66:0e:94:17:61:93 4092 public    up        41      1500 off
Spine-1      eth0.4091 fe80::640e:94ff:fe17:6193/64 fe80::640e:94ff:fe17:6193
 66:0e:94:17:61:93 4091 public    up        98      1500 off
```

Spine-2

```
CLI (network-admin@Spine-2) > vrouter-interface-show l3-port 81,34
vrouter-name nic       ip                          linklocal
mac               vlan vlan-type nic-state l3-port mtu  priority-tag
------------ --------- --------------------------- -------------------------
---------------- ---- --------- --------- ------- ---- ------------
Spine-2      eth0.4092 fe80::640e:94ff:fe30:8394/64 fe80::640e:94ff:fe30:8394
 66:0e:94:30:83:94 4092 public    up        81      1500 off
Spine-2      eth0.4091 fe80::640e:94ff:fe30:8394/64 fe80::640e:94ff:fe30:8394
 66:0e:94:30:83:94 4091 public    up        34     1500 off
```

Leaf-1

```
CLI (network-admin@Leaf-1) > vrouter-interface-show l3-port 41,69
vrouter-name nic       ip                          linklocal
mac               vlan vlan-type nic-state l3-port mtu  priority-tag
------------ --------- --------------------------- -------------------------
---------------- ---- --------- --------- ------- ---- ------------
Leaf-1       eth0.4092 fe80::640e:94ff:fee8:9280/64 fe80::640e:94ff:fee8:9280
 66:0e:94:e8:92:80 4092 public    up        41      1500 off
Leaf-1       eth0.4091 fe80::640e:94ff:fee8:9280/64 fe80::640e:94ff:fee8:9280
 66:0e:94:e8:92:80 4091 public    up        69      1500 off
```

Leaf-2

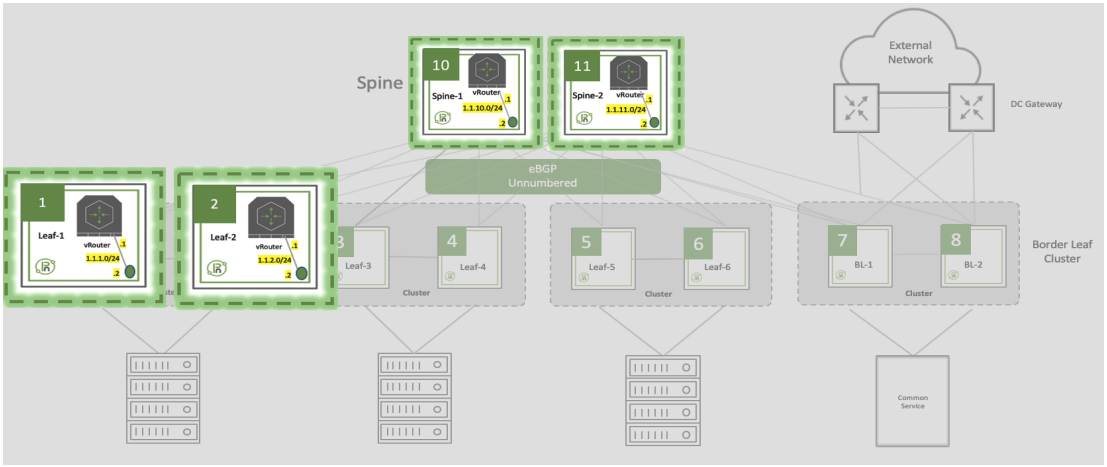
```
CLI (network-admin@Leaf-2) > vrouter-interface-show l3-port 17,33
vrouter-name nic       ip                          linklocal
mac               vlan vlan-type nic-state l3-port mtu  priority-tag
------------ --------- --------------------------- -------------------------
---------------- ---- --------- --------- ------- ---- ------------
Leaf-2       eth0.4092 fe80::640e:94ff:fe16:1359/64 fe80::640e:94ff:fe16:1359
 66:0e:94:16:13:59 4092 public    up        17      1500 off
Leaf-2       eth0.4091 fe80::640e:94ff:fe16:1359/64 fe80::640e:94ff:fe16:1359
 66:0e:94:16:13:59 4091 public    up        33      1500 off
```

Step 4: Configuring fabric In-Band communication

This step focuses on setting up the Adaptive Cloud Fabric control plane communication using the in-band management interface which is reachable through the switch front panel ports carrying the network traffic.

To behave as a unified single-logical system, all switches that are part of an Adaptive Cloud Fabric need to exchange fabric-related messages such as: control-plane messages, configuration transactions, network and endpoint state notifications, remote procedure calls and file system replication messages. This communication is always encrypted and can be enabled via an out-of-band management network or via the in-band network. In-band communication has two main advantages: first it removes the dependence on an external management network, second it leverages the high-availability and resilience of production network.

Figure 11:
In-band fabric communication over layer-3



The following steps configure a dedicated in-band management network on each switch using an internal VLAN. Even though we show a /24 network below, even a /29 will be sufficient for in-band-ip interface.

Spine-1

```
CLI (network-admin@Spine-1) > switch-setup-modify in-band-ip 1.1.100.1/24
CLI (network-admin@Spine-1) > vlan-create id 1000 scope local
```

Configure IP and switch-route to enable IP reachability between in-band interfaces of all the nodes.

```
CLI (network-admin@Spine-1) > vrouter-interface-add vrouter-name Spine-1 ip
1.1.100.2/24 vlan 1000 fabric-nic

CLI (network-admin@Spine-1) > switch-route-create network 1.1.0.0/16
gateway-ip 1.1.100.2
```

Spine-2

```
CLI (network-admin@Spine-2) > switch-setup-modify in-band-ip 1.1.200.1/24

CLI (network-admin@Spine-2) > vlan-create id 1000 scope local

CLI (network-admin@Spine-2) > vrouter-interface-add vrouter-name Spine-2 ip
1.1.200.2/24 vlan 1000 fabric-nic

CLI (network-admin@Spine-2) > switch-route-create network 1.1.0.0/16
gateway-ip 1.1.200.2
```

Leaf-1

```
CLI (network-admin@Leaf-1) > switch-setup-modify in-band-ip 1.1.1.1/24

CLI (network-admin@Leaf-1) > vlan-create id 1000 scope local

CLI (network-admin@Leaf-1) > vrouter-interface-add vrouter-name Leaf-1 ip
1.1.1.2/24 vlan 1000 fabric-nic

CLI (network-admin@Leaf-1) > switch-route-create network 1.1.0.0/16 gateway-
ip 1.1.1.2
```

Leaf-2

```
CLI (network-admin@Leaf-2) > switch-setup-modify in-band-ip 1.1.2.1/24

CLI (network-admin@Leaf-2) > vlan-create id 1000 scope local

CLI (network-admin@Leaf-2) > vrouter-interface-add vrouter-name Leaf-2 ip
1.1.2.2/24 vlan 1000 fabric-nic

CLI (network-admin@Leaf-2) > switch-route-create network 1.1.0.0/16 gateway-
ip 1.1.2.2
```

NOTE: After configuring in-band properties on all switches, validate the in-band management connectivity using `ping command`

Spine-1

```
CLI (network-admin@Spine-1) > ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=62 time=1.15 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=62 time=1.23 ms ^C
--- 1.1.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms rtt min/avg/max/mdev =
1.153/1.194/1.235/0.041 ms

CLI (network-admin@Spine-1) > ping 1.1.2.1
PING 1.1.2.1 (1.1.2.1) 56(84) bytes of data.
64 bytes from 1.1.2.1: icmp_seq=1 ttl=62 time=1.00 ms
64 bytes from 1.1.2.1: icmp_seq=2 ttl=62 time=1.00 ms ^C
--- 1.1.2.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms rtt min/avg/max/mdev =
1.000/1.001/1.003/0.031 ms CLI (network-admin@Spine-1) >
```

Spine-2

```
CLI (network-admin@Spine-2) > ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=2 ttl=62 time=1.01 ms ^C
--- 1.1.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1017ms rtt min/avg/max/mdev =
0.832/0.925/1.019/0.098 ms

CLI (network-admin@Spine-2) > ping 1.1.2.1
64 bytes from 1.1.2.1: icmp_seq=2 ttl=62 time=0.979 ms ^C
--- 1.1.2.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms rtt min/avg/max/mdev =
0.979/0.984/0.989/0.005 ms CLI (network-admin@Spine-2)
```

Leaf-1

```
CLI (network-admin@Leaf-1) > ping 1.1.100.1
PING 1.1.100.1 (1.1.100.1) 56(84) bytes of data.
64 bytes from 1.1.100.1: icmp_seq=1 ttl=61 time=1.17 ms
64 bytes from 1.1.100.1: icmp_seq=2 ttl=61 time=1.01 ms ^C
--- 1.1.100.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms rtt min/avg/max/mdev =
1.016/1.096/1.177/0.087 ms

CLI (network-admin@Leaf-1) > ping 1.1.200.1
PING 1.1.200.1 (1.1.200.1) 56(84) bytes of data.
64 bytes from 1.1.200.1: icmp_seq=1 ttl=62 time=1.15 ms
64 bytes from 1.1.200.1: icmp_seq=2 ttl=62 time=1.05 ms ^C
--- 1.1.200.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms rtt min/avg/max/mdev =
1.052/1.101/1.151/0.059 ms CLI (network-admin@Leaf-1) >
```

Leaf-2

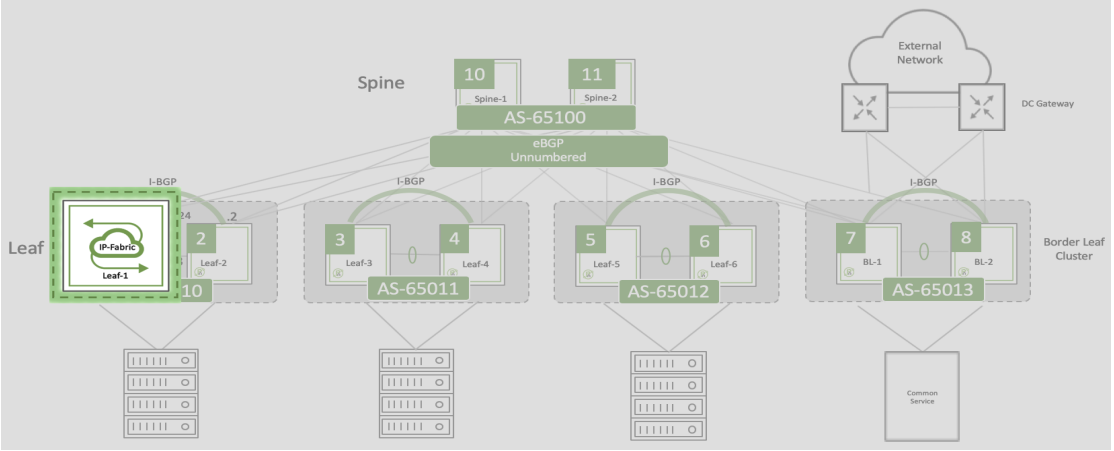
```
CLI (network-admin@Leaf-2) > ping 1.1.100.1
PING 1.1.100.1 (1.1.100.1) 56(84) bytes of data.
64 bytes from 1.1.100.1: icmp_seq=1 ttl=61 time=1.09 ms
64 bytes from 1.1.100.1: icmp_seq=2 ttl=61 time=0.979 ms ^C
--- 1.1.100.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms rtt min/avg/max/mdev =
0.979/1.039/1.099/0.060 ms

CLI (network-admin@Leaf-2) > ping 1.1.200.1
PING 1.1.200.1 (1.1.200.1) 56(84) bytes of data.
64 bytes from 1.1.200.1: icmp_seq=1 ttl=62 time=1.10 ms
64 bytes from 1.1.200.1: icmp_seq=2 ttl=62 time=1.00 ms ^C
--- 1.1.200.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms rtt min/avg/max/mdev =
1.002/1.053/1.105/0.060 ms
```

Step 5 – Create a new fabric instance

After completing the initial setup, the BGP underlay, and the in-band properties, it is now time to create a new fabric instance to subsequently add all other switches.

Figure 12:
Creating fabric over Layer-3(BGP)



Choose any switch to create the fabric instance and use the optional password option, so that other switches can join the fabric only by providing the assigned password.

To create a new fabric instance, use the following command:

Leaf-1

```
CLI (network-admin@Leaf-1*) > fabric-create name IP-Fabric vlan 1000 fabric-network
in-band control-network in-band fabric-advertisement-network inband-only password
```

Adding members to the IP-Fabric.

Each new member joins the fabric by using an existing member IP address and the fabric internal VLAN. To add a switch to an available fabric instance, use the following `fabric-join` command and type the required password twice as prompted.

Leaf-2

```
CLI (network-admin@Leaf-2) > fabric-join switch-ip 1.1.1.1 vlan1000 password
fabric password: Password
confirm fabric password: Password
```

Leaf-3

```
CLI (network-admin@Leaf-3) > fabric-join switch-ip 1.1.1.1 vlan1000 password
fabric password: Password
confirm fabric password: Password
```

Leaf-4

```
CLI (network-admin@Leaf-4) > fabric-join switch-ip 1.1.1.1 vlan1000 password
fabric password: Password
confirm fabric password: Password
```

Leaf-5

```
CLI (network-admin@Leaf-5) > fabric-join switch-ip 1.1.1.1 vlan1000 password
fabric password: Password
confirm fabric password: Password
```

Leaf-6

```
CLI (network-admin@Leaf-6) > fabric-join switch-ip 1.1.1.1 vlan1000 password
fabric password: Password
confirm fabric password: Password
```

BL-1

```
CLI (network-admin@BL-1) > fabric-join switch-ip 1.1.1.1 vlan1000 password
fabric password: Password
confirm fabric password: Password
```

BL-2

```
CLI (network-admin@BL2-2) > fabric-join switch-ip 1.1.1.1 vlan1000 password
fabric password: Password
confirm fabric password: Password
```

Spine-1

```
CLI (network-admin@Spine-1) > fabric-join switch-ip 1.1.1.1 vlan1000 password
fabric password: Password
confirm fabric password: Password
```

Spine-2

```
CLI (network-admin@Spine-2) > fabric-join switch-ip 1.1.1.1 vlan1000 password
fabric password: Password
confirm fabric password: Password
```

With the command fabric-node-show, verify for each fabric node that the state value is online, and that the fab-tid values are the same.

NOTE: The fab-tid (fabric transaction identifier) represents a configuration transaction index in the database used to ensure all the switches of the Fabric have a consistent configuration w.r.t. fabric-wide objects (e.g. VLANs, policies etc.)

```
CLI (network-admin@Leaf-1) > fabric-node-show layout horizontal format name,fab-name,mgmt-
ip,in-band-ip,fab-id,state,firmware-upgrade,device-state,
name    fab-name   mgmt-ip       in-band-ip    fab-id state    firmware-upgrade    device-state
-----   --------   ----------    ----------    ------ -----    ----------------    -------------
Leaf-1  IP-Fabric  10.36.10.34/24 1.1.1.1/24   136    online   5.2.0-5020015631    ok
Leaf-2  IP-Fabric  10.36.10.35/24 1.1.2.1/24   136    online   5.2.0-5020015631    ok
Leaf-3  IP-Fabric  10.36.10.87/24 1.1.3.1/24   136    online   5.2.0-5020015631    ok
Leaf-4  IP-Fabric  10.36.10.82/24 1.1.4.1/24   136    online   5.2.0-5020015631    ok
Leaf-5  IP-Fabric  10.36.10.36/24 1.1.5.1/24   136    online   5.2.0-5020015631    ok
Leaf-6  IP-Fabric  10.36.10.83/24 1.1.6.1/24   136    online   5.2.0-5020015631    ok
BL-1    IP-Fabric  10.36.10.69/24 1.1.7.1/24   136    online   5.2.0-5020015631    ok
BL-2    IP-Fabric  10.36.10.43/24 1.1.8.1/24   136    online   5.2.0-5020015631    ok
Spine-1 IP-Fabric  10.36.10.46/24 1.1.100.1/24 136    online   5.2.0-5020015631    ok
Spine-2 IP-Fabric  10.36.10.84/24 1.1.200.1/24 136    online   5.2.0-5020015631    ok
```

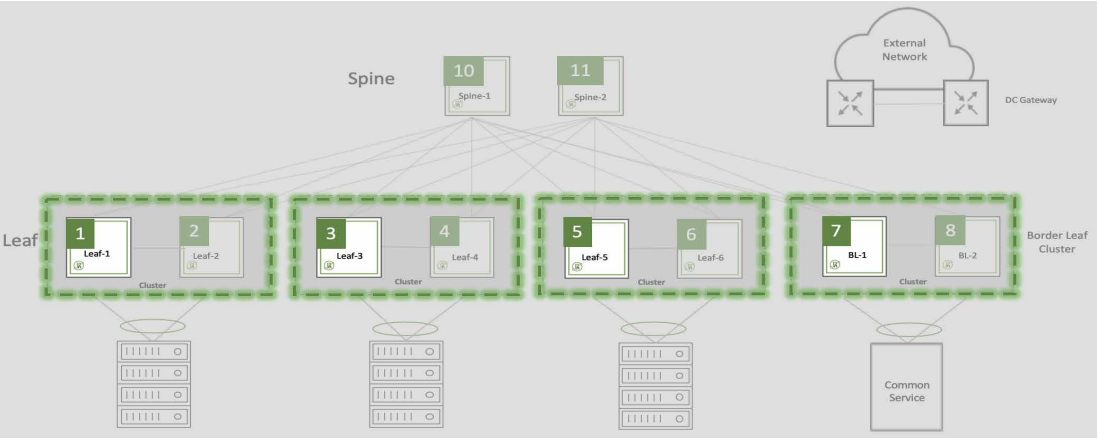
Once the switches have joined the Fabric, the switches can be provisioned and monitored with a single management session (SSH) to any switch in the fabric.

Step 6 – Configure High-Availability Clusters for dual-home server connectivity

Cluster objects provide redundant multihoming connectivity to any server, storage, and any other endpoint.

A cluster, in Pluribus parlance, indicates a high-availability (HA) pair of switches that function as a single virtual chassis to provide continuity of service and traffic load balancing when any component or an entire device fails. Clusters are used for critical applications when all single points of failure must be eliminated.

Figure 13:
Netvisor cluster (High-Availability switch pairs)



Create a cluster configuration between Leaf-1 and Leaf-2. On Leaf-1 enter the cluster-create command:

Leaf-1 & 2

```
CLI (network-admin@Leaf-1) > cluster-create name Leaf-1-to-Leaf-2-Cluster cluster-
node-1 Leaf-1 cluster-node-2 Leaf-2
```

To create a cluster configuration on rest of the leaf switches, use the following command:

Leaf-3 & 4

```
CLI (network-admin@Leaf-1) > cluster-create name Leaf-3-to-Leaf-4-Cluster cluster-
node-1 Leaf-3 cluster-node-2 Leaf-4
```

Leaf-5 & 6

```
CLI (network-admin@Leaf-1) > cluster-create name Leaf-5-to-Leaf-6-Cluster cluster-
node-1 Leaf-5 cluster-node-2 Leaf-6
```

BL-1 & 2

```
CLI (network-admin@Leaf-1) > cluster-create name BL-1-to-BL-2-Cluster cluster-node-1
BL-1 cluster-node-2 BL-2
```

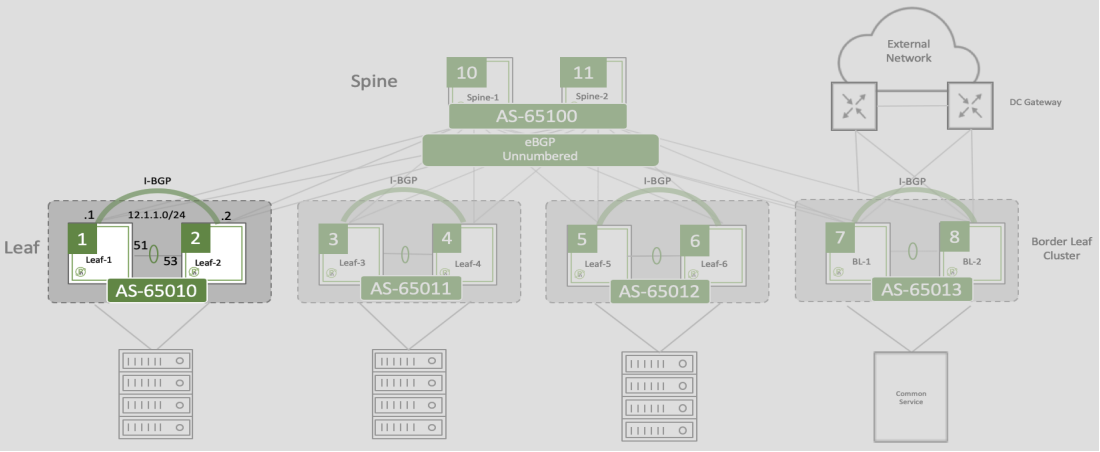
To verify the status of the cluster, use the cluster-show command:

```
CLI (network-admin@Leaf-1) > cluster-show
switch name  state  cluster-node-1   cluster-node-2   tid  mode    ports   remote-ports.  cluster-
sync-timeout(ms)  cluster-sync-offline-count
--------     --------      ------  --------------  --------------   ---       -----   -------
-------------  -------------------------  --------------------------
Leaf-1   Leaf-1-to-Leaf-2-Cluster online.   Leaf-1      Leaf-2   0     master    51      53
         2000                       3
Leaf-2   Leaf-1-to-Leaf-2-Cluster online.   Leaf-1      Leaf-2   0     salve     53      51
         2000                       3
CLI (network-admin@Leaf-1) >
```

Step 7 – Enable iBGP between Cluster-pair Leaf to provide dual-homing to host

Leaf switches that are part of the same high-availability cluster use SVI (VLAN 12) address (12.1.1.0/24) for iBGP peering and use the same ASN.

Figure 14:
iBGP peering between cluster nodes



Create VLAN interfaces (a.k.a. SVIs), on the Leaf switches and assign IP addresses.

Leaf-1

```
CLI (network-admin@Leaf-1) > vlan-create id 12 scope local
CLI (network-admin@Leaf-1) > vlan-port-add vlan-id 12 ports 51 tagged
CLI (network-admin@Leaf-1) > vrouter-interface-add vrouter-name Leaf-1 vlan 12 ip
12.1.1.1/24
```

Configure the iBGP adjacency between the leaf cluster peers, set the next-hop-self property and enable BFD.

```
CLI (network-admin@Leaf-1) > vrouter-bgp-add vrouter-name Leaf-1 neighbor
12.1.1.2 remote-as 65010 next-hop-self bfd
```

Leaf-2

```
CLI (network-admin@Leaf-2) > vlan-create id 12 scope local
CLI (network-admin@Leaf-2) > vlan-port-add vlan-id 12 ports 53 tagged
CLI (network-admin@Leaf-2) > vrouter-interface-add vrouter-name Leaf-2 vlan 12
ip 12.1.1.2/24
CLI (network-admin@Leaf-2) > vrouter-bgp-add vrouter-name Leaf-2 neighbor
12.1.1.1 remote-as 65010 next-hop
```

QUICK CHECK: After configuring eBGP on both Leaf and Spine, a BGP session is established and can be verified by running the `vrouter-bgp-neighbor-show` command. It is also possible to validate the routes received from this neighbor by running a `vrouter-routes-show` command and verify if they are installed in hardware routing tables with `vrouter-fib-routes-show` command.

Spine-1

```
CLI (network-admin@Spine-1) > vrouter-bgp-neighbor-show
vrouter-name l3-port nic       ver remote-as msg_rcvd msg_sent up/down  state/pfxrcd
------------ ------- --------- --- --------- -------- -------- -------- ------------
Spine-1      98      eth0.4091 4   65010     15887    15938    01w3d18h 18
Spine-1      41      eth0.4092 4   65010     18949    18978    4d02h53m 18
```

```
CLI (network-admin@Spine-1) > vrouter-routes-show
vrouter-name network       type      interface next-hop                  distance metric
------------ ------------- --------- --------- ------------------------- -------- ------
Spine-1       12.1.1.0/24  bgp       eth0.4091 fe80::640e:94ff:fe16:1359 20       0
Spine-1       12.1.1.0/24  bgp       eth0.4092 fe80::640e:94ff:fee8:9280 20       0
Spine-1       fe80::/64    connected eth0.4092
```

```
CLI (network-admin@Spine-1) > vrouter-fib-routes-show
vrid ip         prelen intf-id bd vlan port nexthop-mac       flags            egress-id ecmp-group
---- ---------- ------ ------- -- ---- ---- ----------------- ---------------- --------- ----------
0    12.1.1.0   24     0          4092 41   66:0e:94:e8:92:80 ECMP              100008   200256
0    12.1.1.0   24     2          4091 98   66:0e:94:16:13:59 ECMP              100009   200256
0    fe80::     64     8191            0                     local-subnet,hit  100002
0    fe80::     10     8191            0                     local-subnet,hit  100002
```

Spine-2

```
CLI (network-admin@Spine-2) > vrouter-bgp-neighbor-show
vrouter-name l3-port nic       ver remote-as msg_rcvd msg_sent up/down  state/pfxrcd
------------ ------- --------- --- --------- -------- -------- -------- ------------
Spine-2      34      eth0.4091 4   65010     15599    15655    01w3d18h 18
Spine-2      81      eth0.4092 4   65010     15629    15668    4d02h55m 18
```

```
CLI (network-admin@Spine-2) > vrouter-routes-show
vrouter-name network       type      interface next-hop                  distance metric
------------ ------------- --------- --------- ------------------------- -------- ------
Spine-2      1.1.1.0/24    bgp       eth0.4091 fe80::640e:94ff:fe16:1359 20       0
Spine-2      1.1.1.0/24    bgp       eth0.4092 fe80::640e:94ff:fee8:9280 20       0
Spine-2      fe80::/64     connected eth0.4087
```

```
CLI (network-admin@Spine-2) > vrouter-fib-routes-show
vrid ip          prelen intf-id bd vlan port nexthop-mac       flags            egress-id ecmp-group
---- ---------- ------ ------- -- ---- ---- ----------------- ---------------- --------- ----------
0    12.1.1.0   24     0          4092 81   66:0e:94:e8:92:80 ECMP              100010   200256
0    12.1.1.0   24     1          4091 34   66:0e:94:16:13:59 ECMP              100013   200256
0    fe80::     64     8191            0                     local-subnet,hit   100002
0    fe80::     10     8191            0                     local-subnet,hit   100002
```

Leaf-1

```
CLI (network-admin@Leaf-1) > vrouter-bgp-neighbor-show
vrouter-name neighbor l3-port. Nic.      ver remote-as msg_rcvd msg_sent up/down  state/pfxrcd
------------ -------- ------- --------- --- --------- -------- -------- -------- ------------
Leaf-1                69      eth0.4091 4   65100     15773    15742    4d02h56m 18
Leaf-1                41      eth0.4092 4   65100     15762    15753    4d02h56m 18
Leaf-1       12.1.1.2                   4   65010     15718    15734    01w3d17h 17
```

```
CLI (network-admin@Leaf-1) > vrouter-routes-show
vrouter-name network       type      interface next-hop                   distance metric
------------ ------------- --------- --------- -------------------------- -------- ------
Leaf-1       12.1.1.0/24   connected eth0.12   12.1.1.2                   200      0
Leaf-1       fe80::/64     connected eth0.4092
```

```
CLI (network-admin@Leaf-1) > vrouter-fib-routes-show
vrid ip          prelen intf-id bd vlan port nexthop-mac       flags                 egress-id ecmp-group
---- ---------- ------ ------- -- ---- ---- ----------------- --------------------- --------- ----------
12.1.1.0   24   8191       0    0  00:00:00:00:00:00 ECMP,local-subnet,hit   100002     200001
12.1.1.0   24   2          4091 69 66:0e:94:30:83:94 ECMP                    100008     200001
fe80::     64   8191            0                    local-subnet,hit      100002
fe80::     10   8191            0                    local-subnet          100002
```

Leaf-2

```
CLI (network-admin@Leaf-2) > vrouter-bgp-neighbor-show
vrouter-name neighbor l3-port nic.      ver remote-as msg_rcvd msg_sent up/down  state/pfxrcd
------------ -------- ------- --------- --- --------- -------- -------- -------- ------------
Leaf-2                33      eth0.4091 4   65100     15744    15692    01w3d18h 18
Leaf-2                17      eth0.4092 4   65100     15731    15692    01w3d18h 18
```

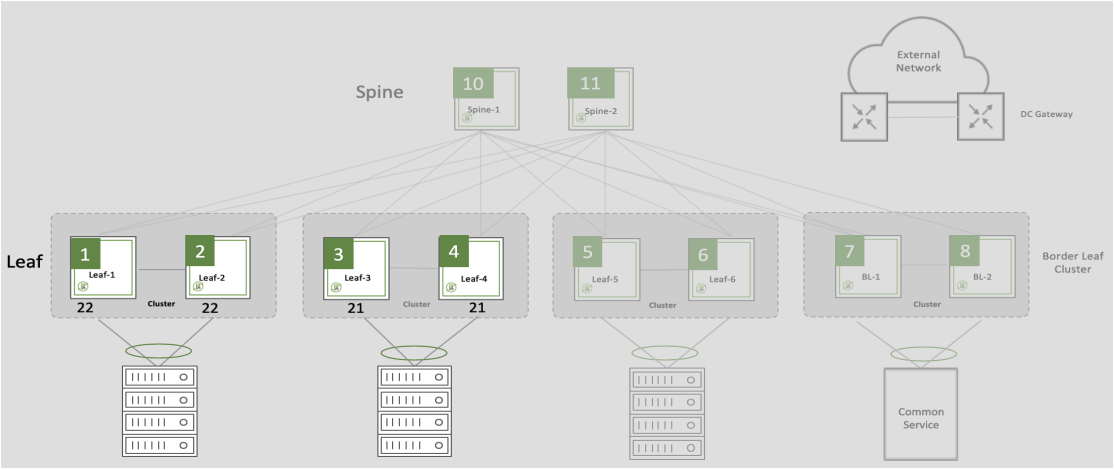
```
CLI (network-admin@Leaf-2) > vrouter-routes-show
vrouter-name network       type      interface next-hop                   distance metric
------------ ------------- --------- --------- -------------------------- -------- ------
Leaf-2       12.1.1.0/24   connected eth0.12
Leaf-2       fe80::/64     connected eth0.4092
```

```
CLI (network-admin@Leaf-2) > vrouter-fib-routes-show
vrid ip          prelen intf-id bd vlan port nexthop-mac       flags                 egress-id ecmp-group
---- ---------- ------ ------- -- ---- ---- ----------------- --------------------- --------- ----------
0    12.1.1.0   24     8191    0    0  00:00:00:00:00:00     ECMP,local-subnet,hit 100002    200001
0    12.1.1.0   24     0       4092 17 66:0e:94:17:61:93     ECMP                  100008    200001
0    12.1.1.0   24     2       4091 33 66:0e:94:30:83:94     ECMP                  100009    200001
0    fe80::     64     8191          0                       local-subnet,hit      100002
0    fe80::     10     8191          0                       local-subnet          100002
```

Step 8 – Configure Active-Active vLAGs

Pluribus implementation of multi-chassis Link Aggregation Group (M-LAG) is called vLAG (Virtual Link Aggregation Group) and can be created on a switch cluster. This provides active-active high availability and very fast traffic steering in case of link failure.

Figure 15:
Active-Acitve vLAG towards a dual-homed host



Create the server-facing active-active vLAG. This example uses a dynamic LAG protocol (LACP mode active).

Leaf-1

```
CLI (network-admin@Leaf-1) > switch Leaf-1 vlag-create name L1-VLAG-L2-To-Host port 22
peer-port 22 mode active-active lacp-mode active
```

Leaf-3

```
CLI (network-admin@Leaf-3) > switch Leaf-3 vlag-create name L3-VLAG-L4-To-Host port 21
peer-port 21 mode active-active lacp-mode active
```

Then verify the vLAG status for each cluster with the command vlag-show:

```
CLI (network-admin@Leaf-1*) > vlag-show

name  cluster mode peer-port status local-state peer-state lacp-mode lack-fallback lacp-fallback-
timeout lace-individual lacp-port-priority
----- ------- ---- --------- ------ ----------- ---------- --------- ------------- ---------------
--------------- ------------------
VLAG-To-Host L1-VLAG-L2-To-Host active-active Leaf-1 22 Leaf-2 22 normal enabled enabled active
bundle 50 none 32768
CLI (network-admin@Leaf-1*) >
```

```
CLI (network-admin@Leaf-1*) > switch Leaf-3
CLI (network-admin@Leaf-3*) > vlag-show

name  cluster mode peer-port status local-state peer-state lacp-mode lack-fallback lacp-fallback-
timeout lace-individual lacp-port-priority
----- ------- ---- --------- ------ ----------- ---------- --------- ------------- ---------------
--------------- ------------------
VLAG-To-Host L3-VLAG-L4-To-Host active-active Leaf-3 21 Leaf-4 21 normal enabled enabled active
bundle 50 none 32768
CLI (network-admin@Leaf-3*) >
```

Step 9 – Configure VRRP for VTEP High Availability

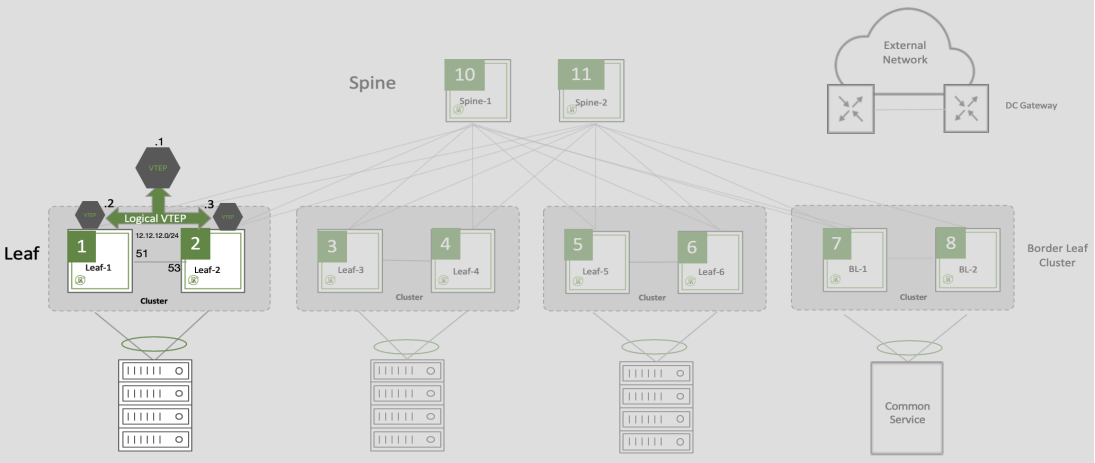
In order to traverse a layer-3 underlay network between a source switch and a destination switch, the overlay traffic needs to be VxLAN encapsulated in an IP-UDP header envelope. This operation technically creates a VxLAN tunnel from the source switch to the destination switch.

A Virtual Tunnel Endpoint (VTEP) terminates VxLAN tunnels on a switch. The VTEP performs the encapsulation and decapsulation of VxLAN packets and it is implemented on Pluribus leaf (Top-of-Rack) switches. Each VTEP has a unique IP address that is used for routing the UDP packets between VTEPs.

Pluribus forwarding logic and distributed control plane employ the standard split horizon algorithm to create a loop-free logical forwarding topology in the fabric overlay network to steer any VXLAN traffic over it without creating potential switching loops. Pluribus uses an address registration and tracking technology, called vPort database, to optimize MAC address which is particularly useful to track host moves.

A critical design requirement for data center networks is to deploy the VXLAN transport in conjunction with an active-active high-availability configuration. This guarantees path redundancy and service continuity in case of link or device failure.

Figure 16:
Redundant logical VIP use for Inter VTEP communication



Pluribus uses a standard VRRP protocol to define a single virtual IP address for the VTEP HA pair, whose subnet is announced from both eBGP speakers. In addition, Pluribus implementation optimizes the performance of this technology combination by supporting active-active Layer 3 forwarding on both VRRP routers in a cluster pair.

The configuration steps of the VRRP protocol for a set of vRouters consists of:

- Defining the VRRP group identifier for all vRouters participating in the VRRP protocol: this group ID determines the value of the VRRP virtual MAC address
- Defining a VLAN and providing continuity for this VLAN between the vRouters
- Defining a VTEP subnet according to the IP address plan (VTEP interface)
- Designating a vRouter interface on such VLAN on each vRouter, with a unique priority value: the priority value is used (in decreasing order) to elect the master vRouter that activates the VRRP virtual IP address
- Defining the VRRP virtual IP address, consistent with the VTEP subnet: this IP address will be used as VxLAN tunnel termination

For Leaf-1 and Leaf-2, for example:

- VRRP group ID: 12
- VLAN ID: 102
- Subnet: 12.12.12.0/24 (at least a /29 subnet is recommended)
- VRRP VIP: 12.12.12.1

To configure VRRP, start with Leaf-1 and modify a vRouter that is associated with the VRRP ID of 12.

Leaf-1: Follow the configuration steps as shown below

```
CLI (network-admin@Leaf-1*) > vlan-create id 102 scope local ports 51
CLI (network-admin@Leaf-1*) > vrouter-modify name Leaf-1 hw-vrrp-id 11
```

Add a vRouter interface that corresponds to the router's own real IP address:

```
CLI (network-admin@Leaf-1*) > vrouter-interface-add vrouter-name Leaf-1 ip
12.12.12.2/24 vlan 102 mtu 9216
```

Create the VRRP interface on the master switch with virtual IP 12.12.12.1, VRRP ID 11 and default priority (100). It is recommended to explicitly configure vrrp-priority. This way, the interface with higher priority will be the master.

```
CLI (network-admin@Leaf-1*) > vrouter-interface-add vrouter-name Leaf-1 ip
12.12.12.1/24 vrrp-id 11 vrrp-primary eth0.102 mtu 9216 vlan 102
```

Leaf-2

```
CLI (network-admin@Leaf-2*) > vlan-create id 102 scope local ports 53 CLI (network-
admin@Leaf-2*) > vrouter-modify name Leaf-2 hw-vrrp-id 11

CLI (network-admin@Leaf-2*) > vrouter-interface-add vrouter-name Leaf-2 ip
12.12.12.3/24 vlan 102 mtu 9216

CLI (network-admin@Leaf-2*) > vrouter-interface-add vrouter-name Leaf-2 ip
12.12.12.1/24 vrrp-id 11 vrrp-primary eth0.102 mtu 9216 vlan
```

Then verify the VRRP status(es) with the command vrouter-interface-show vrrp-id:

```
CLI (network-admin@Leaf-1) > vrouter-interface-show vrrp-id 11
```

```
switch vrouter-name nic        ip    mac      vlan nic-state is-vip vrrp-id vrrp-primary vrrp-state
------ ------------ ---------- ----  -----    ---- --------- ------ --------- ---------- ----------
Leaf-1 Leaf-1       eth1.102 12.12.12.1/24 00:00:5e:00:01:0b 102   up    true  11   eth0.102   master
Leaf-2 Leaf-2       eth1.102 12.12.12.1/24 00:00:5e:00:01:0b 102   down  true  11   eth0.102   slave
```

Step 10 – Repeat similar steps on all other nodes in the fabric

Leaf-3

```
port-config-modify port 15 speed 10g enable jumbo
port-config-modify port 17 speed 10g enable jumbo
port-config-modify port 33 speed 10g enable jumbo
port-config-modify port 57 speed 40g enable jumbo

vrouter-create name Leaf-3 fabric-comm

vrouter-modify name Leaf-3 bgp-as 65111 router-id 3.3.3.3 bgp-redistribute connected bgp-max-paths
16 bgp-bestpath-as-path multipath-relax

vrouter-interface-add vrouter-name Leaf-3 l3-port 17 ipv6-unnumbered
vrouter-bgp-add vrouter-name Leaf-3 remote-as external l3-port 17 allowas-in next-hop-self bfd

vrouter-interface-add vrouter-name Leaf-3 l3-port 33 ipv6-unnumbered
vrouter-bgp-add vrouter-name Leaf-3 remote-as external l3-port 33 allowas-in next-hop-self bfd

vlan-create id 1000 scope local
vrouter-interface-add vrouter-name Leaf-3 ip 1.1.3.2/24 vlan 1000 fabric-nic
switch-route-create network 1.1.0.0/16 gateway-ip 1.1.3.2

vlan-create id 34 scope local
vlan-port-add vlan-id 34 ports 15,57 tagged
vrouter-interface-add vrouter-name Leaf-3 vlan 34 ip 34.1.1.1/24
vrouter-bgp-add vrouter-name Leaf-3 neighbor 34.1.1.2 remote-as 65111 next-hop-self bfd allowas-in
```

Leaf-4

```
port-config-modify port 15 speed 10g enable jumbo
port-config-modify port 17 speed 10g enable jumbo
port-config-modify port 33 speed 10g enable jumbo
port-config-modify port 57 speed 40g enable jumbo

vrouter-create name Leaf-4 fabric-comm

vrouter-modify name Leaf-4 bgp-as 65111 router-id 4.4.4.4 bgp-redistribute connected bgp-max-paths
16 bgp-bestpath-as-path multipath-relax

vrouter-interface-add vrouter-name Leaf-4 l3-port 17 ipv6-unnumbered
vrouter-bgp-add vrouter-name Leaf-4 remote-as external l3-port 17 allowas-in next-hop-self bfd

vrouter-interface-add vrouter-name Leaf-4 l3-port 33 ipv6-unnumbered
vrouter-bgp-add vrouter-name Leaf-4 remote-as external l3-port 33 allowas-in next-hop-self bfd

vlan-create id 1000 scope local
vrouter-interface-add vrouter-name Leaf-4 ip 1.1.4.2/24 vlan 1000 fabric-nic
switch-route-create network 1.1.0.0/16 gateway-ip 1.1.4.2

vlan-create id 34 scope local
vlan-port-add vlan-id 34 ports 15,57 tagged
vrouter-interface-add vrouter-name Leaf-4 vlan 34 ip 34.1.1.2/24
vrouter-bgp-add vrouter-name Leaf-4 neighbor 34.1.1.1 remote-as 65111 next-hop-self bfd allowas-in
```

Leaf-5

```
port-config-modify port 17 speed 10g enable jumbo
port-config-modify port 33 speed 10g enable jumbo
port-config-modify port 61 speed 40g enable jumbo

vrouter-create name Leaf-5 fabric-comm

vrouter-modify name Leaf-5 bgp-as 65112 router-id 5.5.5.5 bgp-redistribute connected bgp-max-paths
16 bgp-bestpath-as-path multipath-relax

vrouter-interface-add vrouter-name Leaf-5 l3-port 17 ipv6-unnumbered
vrouter-bgp-add vrouter-name Leaf-5 remote-as external l3-port 17 allowas-in next-hop-self bfd

vrouter-interface-add vrouter-name Leaf-5 l3-port 33 ipv6-unnumbered
vrouter-bgp-add vrouter-name Leaf-5 remote-as external l3-port 33 allowas-in next-hop-self bfd

vlan-create id 1000 scope local
vrouter-interface-add vrouter-name Leaf-5 ip 1.1.5.2/24 vlan 1000 fabric-nic
switch-route-create network 1.1.0.0/16 gateway-ip 1.1.5.2

vlan-create id 56 scope local
vlan-port-add vlan-id 56 ports 61 tagged
vrouter-interface-add vrouter-name Leaf-5 vlan 56 ip 56.1.1.1/24
vrouter-bgp-add vrouter-name Leaf-5 neighbor 56.1.1.2 remote-as 65112 next-hop-self bfd allowas-in
```

Leaf-6

```
port-config-modify port 57 speed 40g enable jumbo
port-config-modify port 61 speed 10g enable jumbo
port-config-modify port 65 speed 100g enable jumbo

vrouter-create name Leaf-6 fabric-comm

vrouter-modify name Leaf-6 bgp-as 65112 router-id 6.6.6.6 bgp-redistribute connected bgp-max-paths
16 bgp-bestpath-as-path multipath-relax

vrouter-interface-add vrouter-name Leaf-6 l3-port 61 ipv6-unnumbered
vrouter-bgp-add vrouter-name Leaf-6 remote-as external l3-port 61 allowas-in next-hop-self bfd

vrouter-interface-add vrouter-name Leaf-6 l3-port 65 ipv6-unnumbered
vrouter-bgp-add vrouter-name Leaf-6 remote-as external l3-port 65 allowas-in next-hop-self bfd

vlan-create id 1000 scope local
vrouter-interface-add vrouter-name Leaf-6 ip 1.1.6.2/24 vlan 1000 fabric-nic
switch-route-create network 1.1.0.0/16 gateway-ip 1.1.6.2

vlan-create id 56 scope local
vlan-port-add vlan-id 56 ports 57 tagged
vrouter-interface-add vrouter-name Leaf-6 vlan 56 ip 56.1.1.2/24
vrouter-bgp-add vrouter-name Leaf-6 neighbor 56.1.1.1 remote-as 65112 next-hop-self bfd allowas-in
```

BL-1

```
port-config-modify port 9 speed 100g enable jumbo
port-config-modify port 17 speed 40g enable jumbo
port-config-modify port 57 speed 100g enable jumbo

vrouter-create name BL-1 fabric-comm

vrouter-modify name BL-1 bgp-as 65113 router-id 7.7.7.7 bgp-redistribute connected bgp-max-paths 16
bgp-bestpath-as-path multipath-relax

vrouter-interface-add vrouter-name BL-1 l3-port 17 ipv6-unnumbered
vrouter-bgp-add vrouter-name BL-1 remote-as external l3-port 17 allowas-in next-hop-self bfd

vrouter-interface-add vrouter-name BL-1 l3-port 69 ipv6-unnumbered
vrouter-bgp-add vrouter-name BL-1 remote-as external l3-port 69 allowas-in next-hop-self bfd

vlan-create id 1000 scope local
vrouter-interface-add vrouter-name BL-1 ip 1.1.7.2/24 vlan 1000 fabric-nic
switch-route-create network 1.1.0.0/16 gateway-ip 1.1.7.2

vlan-create id 78 scope local
vlan-port-add vlan-id 78 ports 9,57 tagged
vrouter-interface-add vrouter-name BL-1 vlan 78 ip 78.1.1.1/24
vrouter-bgp-add vrouter-name BL-1 neighbor 78.1.1.2 remote-as 65113 next-hop-self bfd allowas-in
```

BL-2

```
port-config-modify port 9 speed 100g enable jumbo
port-config-modify port 17 speed 40g enable jumbo
port-config-modify port 57 speed 100g enable jumbo

vrouter-create name BL-2 fabric-comm

vrouter-modify name BL-2 bgp-as 65113 router-id 8.8.8.8 bgp-redistribute connected bgp-max-paths 16
bgp-bestpath-as-path multipath-relax

vrouter-interface-add vrouter-name BL-2 l3-port 17 ipv6-unnumbered
vrouter-bgp-add vrouter-name BL-2 remote-as external l3-port 17 allowas-in next-hop-self bfd

vrouter-interface-add vrouter-name BL-2 l3-port 69 ipv6-unnumbered
vrouter-bgp-add vrouter-name BL-2 remote-as external l3-port 69

vlan-create id 1000 scope local
vrouter-interface-add vrouter-name BL-2 ip 1.1.8.2/24 vlan 1000 fabric-nic
switch-route-create network 1.1.0.0/16 gateway-ip 1.1.8.2

vlan-create id 78 scope local
vlan-port-add vlan-id 78 ports 9,57 tagged
vrouter-interface-add vrouter-name BL-2 vlan 78 ip 78.1.1.2/24
vrouter-bgp-add vrouter-name BL-2 neighbor 78.1.1.1 remote-as 65113 next-hop-self bfd allowas-in
```

Virtualized Services (Overlay) Implementation Phase

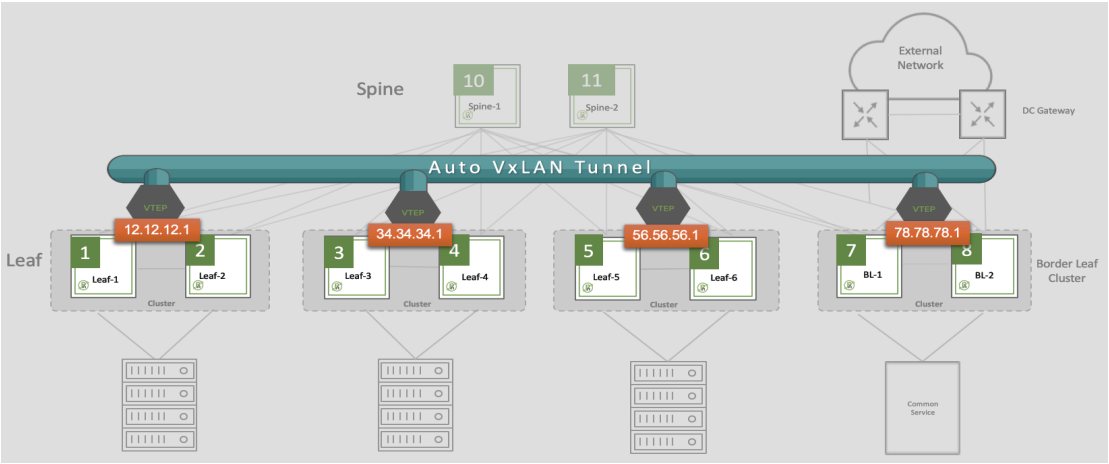
Configuring the Overlay network involves the configuration of (at least) the following base features:

- VTEP (automatic VxLAN tunnels)
- VxLAN Layer-2 VPN Services (VLAN and Bridge Domain)
- VxLAN Layer-3 VPN Services (VRF routing with distributed anycast gateway)

Configure Overlay VTEP with Automatic VxLAN Tunnel

Each Leaf is configured with a VTEP, VxLAN uses VxLAN tunnel endpoint (VTEP) to map tenants to segments and performs encapsulate and de-capsulate packets. To provide VTEP-HA (high-availability) the VTEP uses VIP (VRRP virtual-ip) and reachable other VTEP-HA over the underlay IP Network.

Figure 17:
VTEP High Availability



A VTEP has one or more VNIs associated with it. When frames from these VNIs arrive at the ingress VTEP, the receiving VTEP Leaf encapsulates packet in UDP and IP headers then forwards over the IP network to the egress VTEP. The egress VTEP then decapsulate the IP and UDP headers and delivers the original frame.

To create a single VTEP object per endpoint use the vtep-create command, which triggers the automatic creation of all the required VXLAN tunnels in both directions among all leaf clusters in the fabric, thus yielding a significant amount of configuration and time savings.

The VTEP object created on a cluster Leaf uses the the common VRRP VIP previously defined.

Leaf-1

```
CLI (network-admin@Leaf-1*) > vtep-create name VTEP-Leaf-1 location Leaf-1 vrouter-
name Leaf-1 ip 12.12.12.2 virtual-ip 12.12.12.1
```

Leaf-2

```
CLI (network-admin@Leaf-2*) > vtep-create name VTEP-Leaf-2 location Leaf-2 vrouter-
name Leaf-2 ip 12.12.12.3 virtual-ip 12.12.12.1
```

Leaf-3

```
CLI (network-admin@Leaf-3*) > vtep-create name VTEP-Leaf-3 location Leaf-3 vrouter-
name Leaf-3 ip 34.34.34.2 virtual-ip 34.34.34.1
```

Leaf-4

```
CLI (network-admin@Leaf-4*) > vtep-create name VTEP-Leaf-4 location Leaf-4 vrouter-
name Leaf-4 ip 34.34.34.3 virtual-ip 34.34.34.1
```

Leaf-5

```
CLI (network-admin@Leaf-5*) > vtep-create name VTEP-Leaf-5 location Leaf-5 vrouter-
name Leaf-5 ip 56.56.56.2 virtual-ip 56.56.56.1
```

Leaf-6

```
CLI (network-admin@Leaf-6*) > vtep-create name VTEP-Leaf-6 location Leaf-6 vrouter-
name Leaf-6 ip 56.56.56.3 virtual-ip 56.56.56.1
```

BL-1

```
CLI (network-admin@BL-1*) > vtep-create name VTEP-BL-1 location BL-1 vrouter-name
BL-1 ip 78.78.78.2 virtual-ip 78.78.78.1
```

BL-2

```
CLI (network-admin@BL-2*) > vtep-create name VTEP-BL-2 location BL-2 vrouter-name
BL-2 ip 78.78.78.3 virtual-ip 78.78.78.1
```

Then verify the VTEP status(es) with the command vtep-show:

```
CLI (network-admin@Leaf-1) > vtep-show
scope  name         location vrouter-name ip         virtual-ip
------ ------------ -------- ------------ ---------- ----------
fabric VTEP-Leaf-1 Leaf-1   Leaf-1       12.12.12.2 12.12.12.1
fabric VTEP-Leaf-2 Leaf-2   Leaf-2       12.12.12.3 12.12.12.1
fabric VTEP-Leaf-3 Leaf-3   Leaf-3       34.34.34.2 34.34.34.1
fabric VTEP-Leaf-4 Leaf-4   Leaf-4       34.34.34.3 34.34.34.1
fabric VTEP-Leaf-5 Leaf-5   Leaf-5       56.56.56.2 56.56.56.1
fabric VTEP-Leaf-6 Leaf-6   Leaf-6       56.56.56.3 56.56.56.1
fabric VTEP-BL-1   BL-1     BL-1         78.78.78.2 78.78.78.1
fabric VTEP-BL-2   BL-2     BL-2         78.78.78.3 78.78.78.1
```

VxLAN L2VPN Services

Layer-2 VLAN-to-VNI Mapping

As discussed earlier, in VxLAN networks, each VLAN is mapped to a VNI number representing a VxLAN segment.

In this multitenant environment, endpoints belonging to the same Layer-2 virtual network can be located anywhere in the fabric and communication among endpoints part of the same Layer-2 virtual network is bridged.

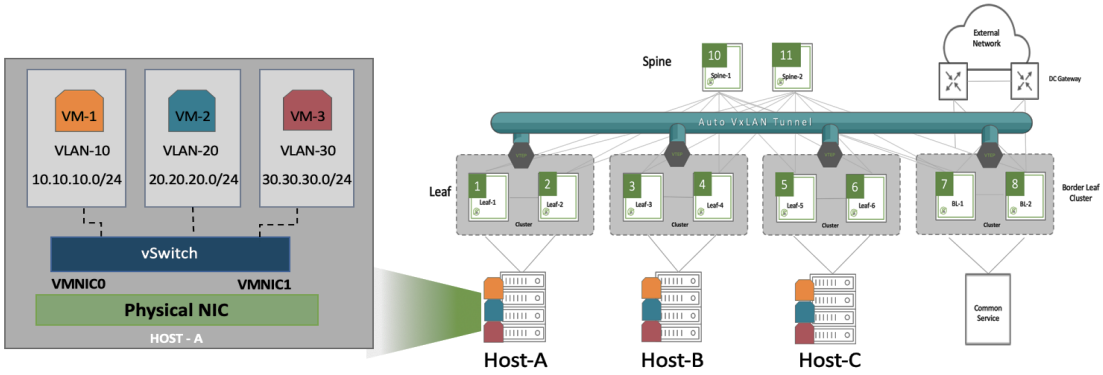


Figure 18: Multitenancy Data Center Network

In the figure below, the tenant VLANs (10,20,30) are mapped to L2 VNI (100010, 100020, 100030). Then those VNIs are automatically associated with the VTEPs of each leaf pair.

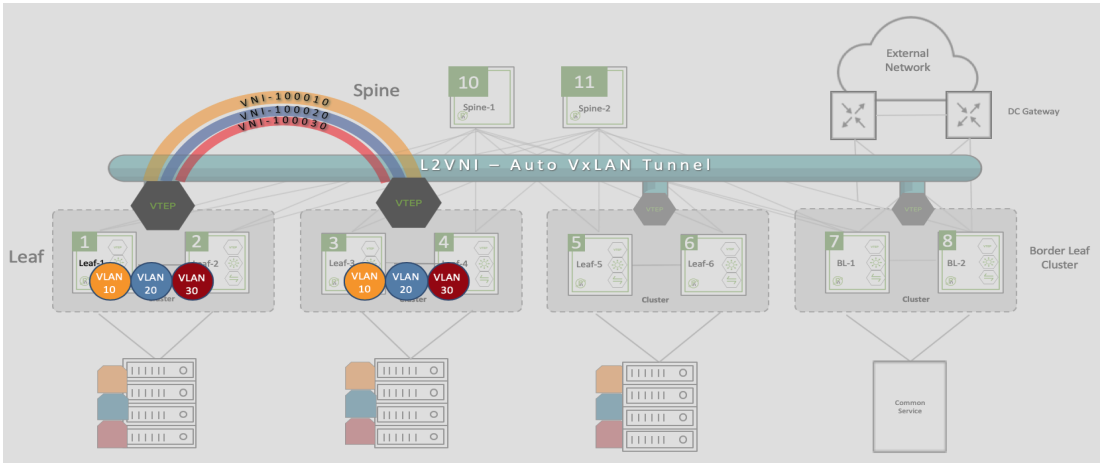


Figure 19: Mapping Layer-2 VLAN with VxLAN Network Identifier (VNI)

One creates an overlay VLAN by specifying a VLAN ID, vxlan (VNI) and auto-vlxan attribute. This automatically assigns the vxlan to all the auto tunnels. The figure above shows the VNIs added to just the two unidirectional tunnels between Leaf-1-Leaf-2 pair and Leaf-3-Leaf-4 pair, but the same extends to the full mesh of tunnels in the fabric.

```
CLI (network-admin@Leaf-1*) > vlan-create id 10 scope fabric auto-vxlan vxlan
100010 ports none
CLI (network-admin@Leaf-1*) > vlan-create id 20 scope fabric auto-vxlan vxlan
100020 ports none
CLI (network-admin@Leaf-1*) > vlan-create id 30 scope fabric auto-vxlan vxlan
100030 ports none
```

Add ports to VLANs, noting port 48 is tagged for recirculation. Adding ports to a VLAN is local to a switch.

```
CLI (network-admin@Leaf-1*) > vlan-port-add vlan-id 10 ports 22,51,48
CLI (network-admin@Leaf-1*) > vlan-port-add vlan-id 20 ports 22,51,48
CLI (network-admin@Leaf-1*) > vlan-port-add vlan-id 20 ports 22,51,48
```

Then verify the VTEP status(es) and VxLAN mapping with the command vtep-show, vtep-vxlan-show tunnel-vxlan-show:

```
CLI (network-admin@Leaf-1) > vtep-show
scope  name        location vrouter-name ip         virtual-ip
------ ----------- -------- ------------ ---------- ----------
fabric VTEP-Leaf-1 Leaf-1   Leaf-1       12.12.12.2 12.12.12.1
fabric VTEP-Leaf-2 Leaf-2   Leaf-2       12.12.12.3 12.12.12.1
fabric VTEP-Leaf-3 Leaf-3   Leaf-3       34.34.34.2 34.34.34.1
fabric VTEP-Leaf-4 Leaf-4   Leaf-4       34.34.34.3 34.34.34.1
fabric VTEP-Leaf-5 Leaf-5   Leaf-5       56.56.56.2 56.56.56.1
fabric VTEP-Leaf-6 Leaf-6   Leaf-6       56.56.56.3 56.56.56.1
fabric VTEP-BL-1   BL-1     BL-1         78.78.78.2 78.78.78.1
fabric VTEP-BL-2   BL-2     BL-2         78.78.78.3 78.78.78.1
```

```
CLI (network-admin@Leaf-1) > vtep-vxlan-show
name        vxlan
----------- ------
VTEP-Leaf-1 100020
VTEP-Leaf-1 100030
VTEP-Leaf-1 100010
```

```
CLI (network-admin@Leaf-1*) > tunnel-show format name,type,vrouter-
name,local-ip,remote-ip,next-hop,tunnelID,state,
name type  vrouter-name local-ip   remote-ip  next-hop
tunnelID   state
------------------------------- ----- ------------ ---------- ---------- --
---------------------- ---------- -----
auto-tunnel-12.12.12.1_34.34.34.1 vxlan Leaf-1       12.12.12.1 34.34.34.1
fe80::640e:94ff:fe30:8394 1275068416 ok
auto-tunnel-12.12.12.1_56.56.56.1 vxlan Leaf-1       12.12.12.1 56.56.56.1
fe80::640e:94ff:fe30:8394 1275068417 ok
auto-tunnel-12.12.12.1_78.78.78.1 vxlan Leaf-1       12.12.12.1 78.78.78.1
fe80::640e:94ff:fe30:8394 1275068418 ok
CLI (network-admin@Leaf-1*) >
```

```
CLI (network-admin@Leaf-1) > tunnel-vxlan-show
switch name                              vxlan
------ --------------------------------- ------
Leaf-1 auto-tunnel-12.12.12.1_34.34.34.1 100010
Leaf-1 auto-tunnel-12.12.12.1_34.34.34.1 100020
Leaf-1 auto-tunnel-12.12.12.1_34.34.34.1 100030
Leaf-1 auto-tunnel-12.12.12.1_56.56.56.1 100010
Leaf-1 auto-tunnel-12.12.12.1_56.56.56.1 100020
Leaf-1 auto-tunnel-12.12.12.1_56.56.56.1 100030
Leaf-1 auto-tunnel-12.12.12.1_78.78.78.1 100010
Leaf-1 auto-tunnel-12.12.12.1_78.78.78.1 100020
Leaf-1 auto-tunnel-12.12.12.1_78.78.78.1 100030
```

Layer-2-to-Bridge-Domain Mapping

A bridge domain (BD) is a set of logical ports that share the single flooding or broadcast domain and mapped to single or multiple 802.1Q VLANs as well as to double tagged IEEE 802.1ad VLAN tags, covering all possible design requirements. There are many options available with bridge domains and this topic is covered extensively in a separate document.

VxLAN L3VPN Services

Adaptive Cloud Fabric adds Layer 3 segmentation to VXLAN interconnections with the support of VRF (Virtual Routing and Forwarding) instances. As part of the Fabric VRFs configuration, you can create IPv4 and IPv6 subnets, which are atomic objects in the Fabric data plane to associate to the VRF instances in order to implement distributed traffic segmentation.

In addition, Anycast Gateway routing function for the Fabric VRFs to enable distributed first-hop routing, redundancy and mobility. This capability uses a dedicated virtual MAC address, called the anycast gateway MAC address, which gets associated with configurable anycast gateway IP addresses as part of the subnet object configuration.

Step 1 – Multitenancy using VRF to isolate tenant traffic from each other

The following commands are used with fabric scope for configuring the subnet objects for the associated anycast gateway addresses and the associated VNIs:

Use the following `vrf-create` command can be used to create fabric wider

Leaf-1

```
CLI (network-admin@Leaf-1) > vrf-create name VRF-1 scope fabric
CLI (network-admin@Leaf-1) > vrf-create name VRF-2 scope fabric
CLI (network-admin@Leaf-1) > vrf-create name VRF-3 scope fabric
```

```
CLI (network-admin@Leaf-1) > subnet-create name subnet-vxlan-100010 scope fabric
vxlan 100010 network 10.10.10.0/24 anycast-gw-ip 10.10.10.254 vrf VRF-1
```

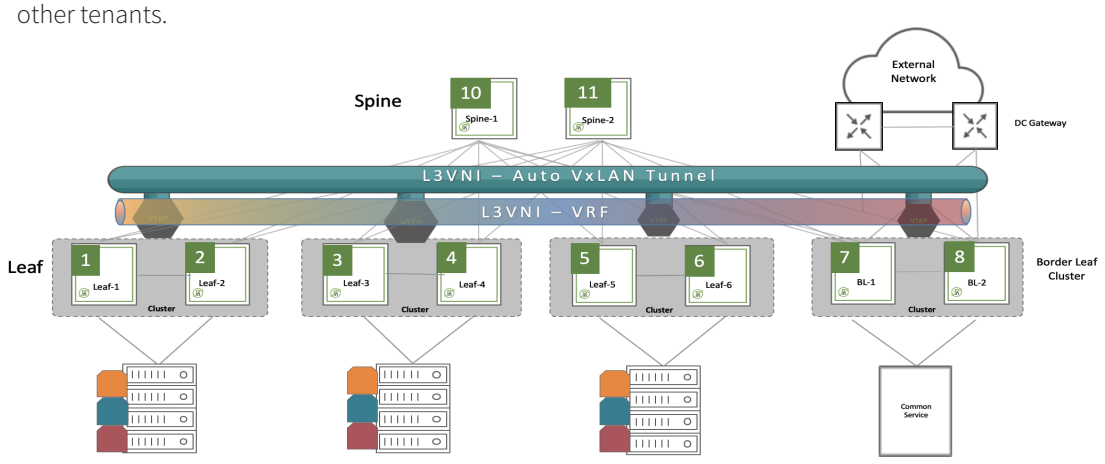
```
CLI (network-admin@Leaf-1) > subnet-create name subnet-vxlan-100020 scope fabric
vxlan 100020 network 20.20.20.0/24 anycast-gw-ip 20.20.20.254 vrf VRF-2
```

```
CLI (network-admin@Leaf-1) > subnet-create name subnet-vxlan-100030 scope fabric
vxlan 100030 network 30.30.30.0/24 anycast-gw-ip 30.30.30.254 vrf VRF-3
```

Step 2 – VxLAN Layer-3 Service

In multitenant data center with VxLAN as a *virtualization overlay* provide virtual network that runs over an IP underlay network. This building block enables multitenancy in a network and allows to share the same physical network across multiple tenants, while keeping each tenant's network traffic isolated from the other tenants.

Figure 20:
Layer-3 Multi-tenancy - VxLAN

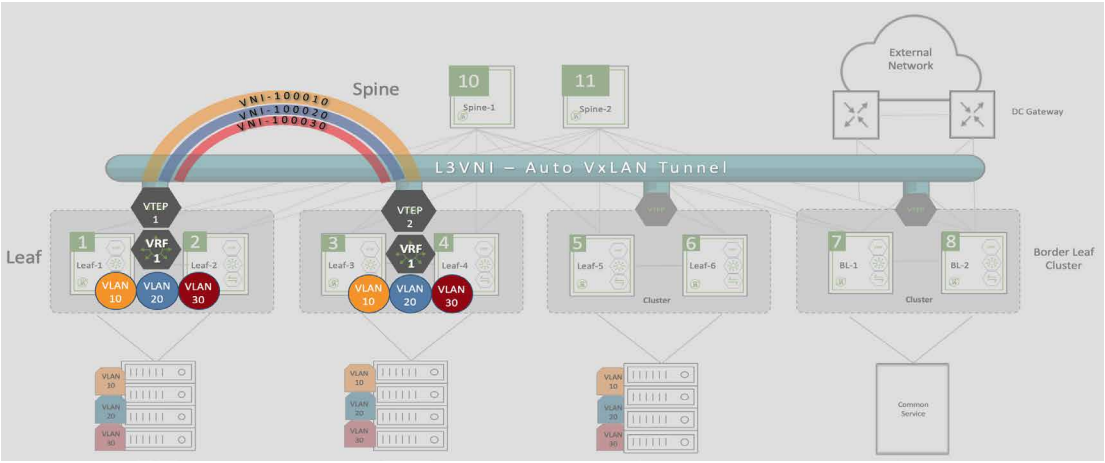


A tenant is a group such as a business unit, department, security, workgroup, or application. This groups may need to communicate with other groups in the same tenancy, and tenants may communicate with other tenants if permitted by network policies.

This group are typically in same subnet (VLAN) that can communicate with other devices and reach external groups and endpoints by way of a virtual routing and forwarding (VRF) instance.

Inter-VLAN routing happens at the Leaf using VxLAN recirculation technique with the dedicated vxlan-loopback-trunk configuration.

Figure 21:
Layer-3 VRF Mapping



In most designs the VXLAN overlay network requires that overlay traffic be routed as well as bridged.

This is typically required for the host traffic to be routed across VNI-mapped VLANs and/or to reach the data center gateways, firewalls (or other L3/L4 appliances), so as to be forwarded to the Internet. The reverse path from the Internet also requires that the traffic be routed as well to reach the hosts.

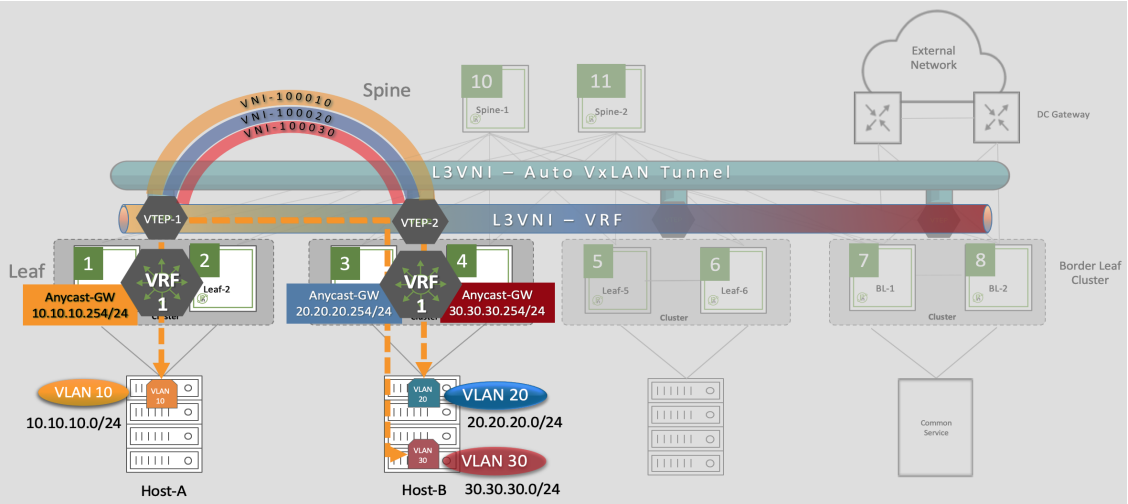
Any fabric packets that need to be routed between two hosts in different VNI-mapped VLANs are sent to a distributed anycast gateway and then VXLAN-encapsulated or - decapsulated depending on source and/or destinations host location.

In particular, Fabric leaf switches use subnet objects for management purposes to represent groups of directly connected hosts with a fabric wide scope across the VXLAN interconnect. Netvisor ONE also uses them to program subnet routes into the hardware to send Layer 3 packets corresponding to unresolved adjacencies to the software so that next-hop resolution through ARP requests can be performed. When a host responds to the ARP request(s), more specific Layer 2 and Layer 3 host entries are configured in the hardware so that end-to-end forwarding ensues.

In addition, Netvisor ONE supports the anycast gateway routing function for the Fabric VRFs to enable distributed first-hop routing, redundancy and mobility. This capability uses a dedicated virtual MAC address, called the anycast gateway MAC address, which gets associated with configurable anycast gateway IP addresses as part of the subnet object configuration.

The default MAC address for the anycast gateway function is 64:0e:94:40:00:02. It can be displayed with the fabric-anycast-gateway-show command. If necessary, you can also modify it using the fabric-anycast-gateway-modify command.

Figure 22:
Layer-3 VRF Mapping with Multiple Tenant



As show in above figure, tenant in Host A is connected to VTEP-1 and wants to communicate with another tenant (which is in different VLAN) on Host B and attached to VTEP-2. Host A sends data traffic to the anycast gateway associated with the local subnet in VLAN 10. From VLAN 10, traffic is routed based on the destination IP lookup. This result indicates the traffic needs to be VXLAN encapsulated and sends traffic toward VTEP-2, below which Host B resides. The encapsulated VXLAN traffic is sent from VTEP-1 to VTEP-2 in VNI 100010, where it is associated with L3VNI the VRF-1 in which Host A and Host B reside. Once the encapsulated VXLAN traffic arrives at VTEP-2, the traffic is decapsulated and routed within the VRF toward VLAN 20 & 30 where Host B resides. In this way, for traffic from Host A to Host B, a bridge–route–route–bridge symmetric sequence is performed.

Follow the configuration steps as shown below to configure Inter-VLAN routing and bridging.

Leaf-1

Use the following `vrf-create` command can be used to create fabric wider

```
CLI (network-admin@Leaf-1) > vrf-create name VRF-1 scope fabric
```

The following commands are used with `fabric scope` for the configuring the subnet objects for the associated anycast gateway addresses and the associated VNIs:

```
CLI (network-admin@Leaf-1*) > subnet-create name subnet-vxlan-100010 scope
fabric vxlan 100010 network 10.10.10.0/24 anycast-gw-ip 10.10.10.254 vrf VRF-1

CLI (network-admin@Leaf-1*) > subnet-create name subnet-vxlan-100020 scope
fabric vxlan 100020 network 20.20.20.0/24 anycast-gw-ip 20.20.20.254 vrf VRF-1

CLI (network-admin@Leaf-1*) > subnet-create name subnet-vxlan-100030 scope
fabric vxlan 100030 network 30.30.30.0/24 anycast-gw-ip 30.30.30.254 vrf VRF-1
```

Pluribus supports the recirculation technique with the dedicated for any fabric packets that need to be routed between two hosts in different VNI-mapped VLANs.

```
CLI (network-admin@Leaf-1*) > trunk-modify name vxlan-loopback-trunk ports 48
CLI (network-admin@Leaf-2*) > trunk-modify name vxlan-loopback-trunk ports 48
CLI (network-admin@Leaf-3*) > trunk-modify name vxlan-loopback-trunk ports 48
CLI (network-admin@Leaf-4*) > trunk-modify name vxlan-loopback-trunk ports 48
```

The following commands can be seeing the subnet objects associated with VRF-1 for East-West traffic segmentation

```
CLI (network-admin@Leaf-1*) > subnet-show vrf VRF-1
name                scope  vlan vxlan  vrf   network       anycast-gw-ip
packet-relay forward-proto state enable
------------------- ------ ---- ------ ----- ------------- -------------
------------ ------------- ----- ------
subnet-vxlan-100010 fabric 10   100010 VRF-1 10.10.10.0/24 10.10.10.254
disable      dhcp          ok    yes
subnet-vxlan-100020 fabric 20   100020 VRF-1 20.20.20.0/24 20.20.20.254
disable      dhcp          ok    yes
subnet-vxlan-100030 fabric 30   100030 VRF-1 30.30.30.0/24 30.30.30.254
disable      dhcp          ok    yes
```

Single-Pass RIOT

Most switches need to two passes for routing traffic from subnet to the other. For this reason, we specified the vxlan-loopback-trunk earlier which has ports for the second pass. The two passes are for routing and VxLAN tunnel encapsulation. In the latest release, these two operations can be done in a single pass on Dell S52xx switches.

The following CLI enables the single pass. Both options should be enabled for single pass routing and single pass flooding such as needed for ARP.

```
switch * system-settings-modify single-pass-riot single-pass-flood
```

Control Plane Protection

In the older switch models, the CPU port only had eight class queues, much like the front panel ports. However, Netvisor One makes full use of the extended CPU queues available in the newer switch models. In all 43 queues are available including a few unused ones that are customizable by the end-user. With such fine granularity, each protocol such as OSPF, BGP, VRRP or ARP can be given its own queue with special options to protect against rogue hosts hogging the CPU (say with ARP flood for instance). Each queue can be individually rate limited as well. Extended cpu queues are enabled in the newer switches with ONIE install of Netvisor One. However, if they are not enabled, “system-settings-modify cpu-class-enable” will enable extended queues. The CLI “cpu-class-modify” can be used to set the rate for a specific CPU queue.

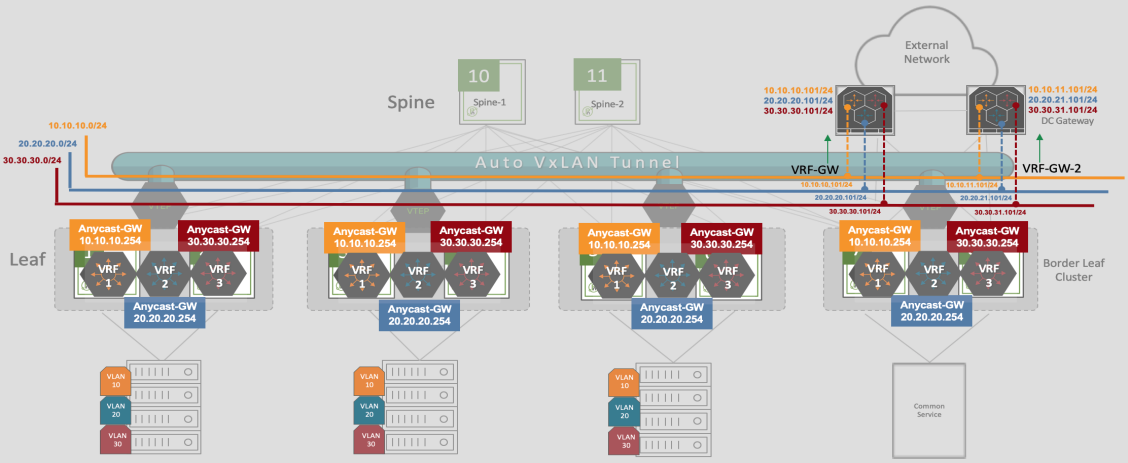
Intra-Extranet Connection

Step 1 – North-South traffic into and out of the datacenter

The previous section addressed the traffic paths within the same, or between different, extended subnets within the data center pod or E-W. The last component of the distributed VRF configuration enables north-to-south interconnection of L3VPN services, allowing hosts connected to subnets behind the VRFs in the fabric to access services that are outside of the data centers, including the public internet.

This is achieved by pointing each VRF to a next-hop gateway, or two for equal cost redundancy. The VRF attribute defining its northbound gateways has local switch scope relevance, allowing this way to have diversity of northbound connectivity for a certain VRF based on physical location, which is very useful for L3VPN services dispersed over multiple data center locations.

Figure 23:
Internal and External Connectivity



As represented in above, the overlay VRF discussed is now provisioned with two additional northbound subnet, which provide reachability in and out of the VRF from/to the DC gateway:

```
CLI (network-admin@Leaf-1) > subnet-create name subnet-vxlan-100010 scope
fabric vxlan 100010 network 10.10.10.0/24 anycast-gw-ip 10.10.10.254 vrf VRF-1

CLI (network-admin@Leaf-1) > subnet-create name subnet-vxlan-100011 scope
fabric vxlan 100011 network 10.10.11.0/24 anycast-gw-ip 10.10.11.254 vrf VRF-1

CLI (network-admin@Leaf-1) > subnet-create name subnet-vxlan-100011 scope
fabric vxlan 100011 network 10.10.12.0/24 anycast-gw-ip 10.10.12.254 vrf VRF-1

CLI (network-admin@Leaf-1) > subnet-create name subnet-vxlan-100020 scope
fabric vxlan 100020 network 20.20.20.0/24 anycast-gw-ip 20.20.20.254 vrf VRF-2

CLI (network-admin@Leaf-1) > subnet-create name subnet-vxlan-100021 scope
fabric vxlan 100021 network 20.20.21.0/24 anycast-gw-ip 20.20.21.254 vrf VRF-2

CLI (network-admin@Leaf-1) > subnet-create name subnet-vxlan-100021 scope
fabric vxlan 100022 network 20.20.22.0/24 anycast-gw-ip 20.20.22.254 vrf VRF-2

CLI (network-admin@Leaf-1) > subnet-create name subnet-vxlan-100030 scope
fabric vxlan 100030 network 30.30.30.0/24 anycast-gw-ip 30.30.30.254 vrf VRF-3

CLI (network-admin@Leaf-1) > subnet-create name subnet-vxlan-100031 scope
fabric vxlan 100031 network 30.30.31.0/24 anycast-gw-ip 30.30.31.254 vrf VRF-3

CLI (network-admin@Leaf-1) > subnet-create name subnet-vxlan-100032 scope
fabric vxlan 100032 network 30.30.32.0/24 anycast-gw-ip 30.30.32.254 vrf VRF-3
```

The VRF is also configured with the IP address of each DC gateway as default next-hop:

```
CLI (network-admin@Leaf-1) > switch * vrf-modify name VRF-1 vrf-gw 10.10.10.101
vrf-gw2 10.10.11.101

CLI (network-admin@Leaf-1) > switch * vrf-modify name VRF-2 vrf-gw 20.20.20.101
vrf-gw2 20.20.21.101

CLI (network-admin@Leaf-1) > switch * vrf-modify name VRF-3 vrf-gw 30.30.30.101
vrf-gw2 30.30.31.101
```

To provide inbound reachability for the VRF inside the pod, the DC gateways will need to be provisioned with a static route for the VRF subnets that need to receive traffic from external networks, using the adjacent Anycast Gateway addresses as next hop:

```
DC-Gateway-1: ip route vrf VRF_1 10.10.12.0/24 10.10.10.254
DC-Gateway-2: ip route vrf VRF_1 10.10.12.0/24 10.10.11.254

DC-Gateway-1: ip route vrf VRF_2 20.20.22.0/24 20.20.20.254
DC-Gateway-2: ip route vrf VRF_2 20.20.22.0/24 20.20.21.254

DC-Gateway-1: ip route vrf VRF_3 30.30.32.0/24 30.30.30.254
DC-Gateway-2: ip route vrf VRF_3 30.30.32.0/24 30.30.31.254
```

The number of routes that are required in order to implement L3VPNs on leaf switches and border leaf nodes is strictly limited to the number of connected subnets in each active VRF, plus two default routes per VRF. As the typical hardware routing scale in modern leaf switches is of the same order as the Layer 2 domain's capacity, this design scales well as it adapts to the hardware architectural model available in most open networking top-of-rack switches and can thus provide thousands of L3VPN services with full horizontal scale with merchant silicon economics.

Appendix -I

This design guide showed several fabric parameters that should be decided and used throughout the fabric especially for building the underlay and the VxLAN tunnels. The table below summarizes these design parameters.

Fabric attributes for all Leaves

Attribute	Value
Fabric VLAN	1000
Fabric Password	<mypwd>

Leaf specific attributes for Leaf-1

Attribute	Value
BGP-AS number	65010
Router-id	1.1.1.1
In-band-IP/vrouter-interface in Fabric VLAN	1.1.1.0/30
Cluster VLAN	12
Cluster IP in Cluster VLAN	12.1.1.0/29
Hw-vrrp-id	12
VTEP VLAN	102
VTEP IP in VTEP VLAN	12.12.12.0/24
VRRP Priority – Primary and Secondary	100, xx

The above table can be repeated with variation for other Leaf switches. Any locally scoped VLAN such as the cluster VLAN or the VTEP VLAN can be the same value across the fabric for simplicity. Creating the above table upfront will make simplify the task of building the fabric.