NVIDIA CUMULUS LINUX

Network Reference Design Guide

Introduction

Overview

In the current era of networking, network simplicity, agility and scale are essential. In the past, applications were designed to function within the same layer 2 (L2) domain. This caused problems because protocols like Spanning Tree (STP) are fragile and noisy. Layer 3 (L3) protocols are increasingly popular as they can scale more easily and efficiently.
Ethernet VPN (EVPN) is a technology that connects L2 network segments separated by an L3 network. It is an extension to Border Gateway Protocol (BGP) that enables the network to carry endpoint reachability information such as L2 MAC addresses and L3 IP addresses.
In the Data Center, EVPN enables optimal east-west and south-north traffic forwarding. It supports Integrating Routing and Bridging for routing between subnets routing and multi-tenancy. In the virtualization scenario, it also supports MAC mobility, so virtual machines can be moved within or across racks. As EVPN is multi-transport, it can run over VXLAN and enables scalable service fabrics.

Key Use Cases

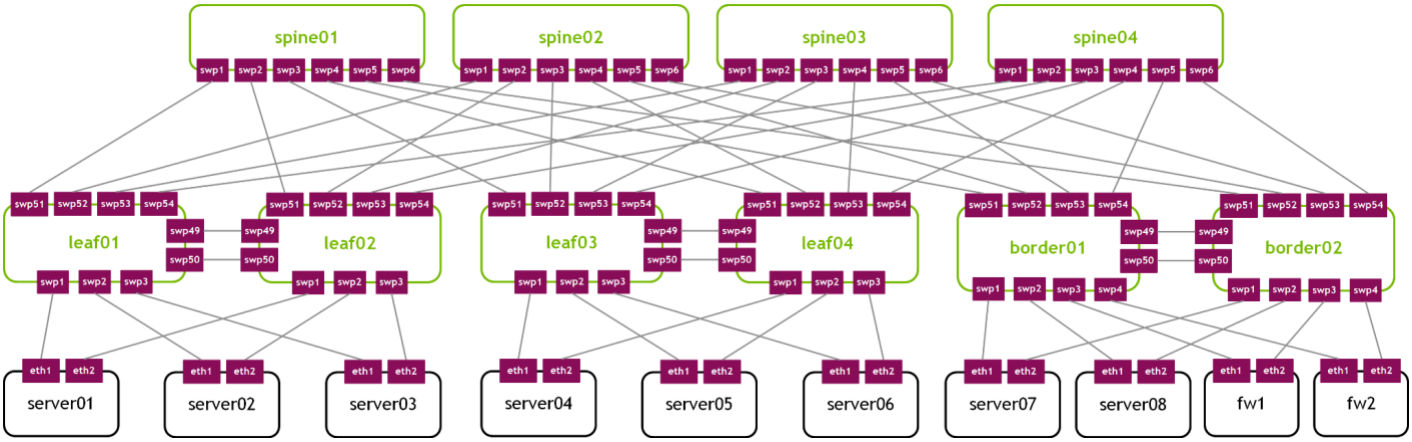
1. L2 and L3 VPN for tenancy
   You can choose to segment your networks at L2, L3, or both. By carrying L2 and L3 endpoint reachability information, EVPN supports integrated bridging and routing in overlay networks.
2. Scaling out the access layer
   a. Active-active L2 and EVPN multihoming at the access is critical for high availability.
   b. Optimal forwarding of east-west and north-south traffic.
   c. Minimizes flooding within the network through protocol-based host MAC, and IPv4 and IPv6 route distribution, additionally providing early Address Resolution Protocol (ARP) termination at the local access switches.
   d. Allows full network bandwidth utilization by running VXLAN at the access layer and eliminating the need to run STP.
3. Extending services beyond a single data center
   VXLAN EVPN enabled the industry to use new data center deployment approaches and optimized ingress routing. As the EVPN control plane evolves, L2 extensions can not only cross physical rack boundaries but also stretch across data centers.

Intended Audience

This document is intended for networking professionals and discusses EVPN-VXLAN cloud data center architecture concepts and tools, and the best practices to consider for deployment.

Topology

The following topology is used throughout the document.



Basic Terminology

1. Leaf

Also referred to as an *access* switch, where servers connect to the network. Servers and storage connect to leaf switches that aggregate the network traffic.

2. Spine

Also referred to as an *aggregation* switch, *end-of-row* switch, or *distribution* switch, where leaf switches connect into the spine, forming the access layer that delivers network connection points for servers.

3. Border leaf

A leaf that connects external services, such as firewalls, load balancers and internet routers, for north-south traffic typically. This acts as a demarcation zone between the underlying fabric and the outside world. The border leaf is responsible for announcing the prefix of the fabric to the outside world and determining how to join the internet and other data centers.

4. Super spine

Sometimes referred to as a *spine aggregation* switch, *end-of-row* switch, or *data center core* switch.

Common Data Center Architectures

Over the past decade, data centers have increased in size; they require applications that are vastly different from the traditional client–server applications and need much faster deployment speeds (seconds instead of days). This changes how networks must be designed and deployed. Data centers now have a growing demand for server-to-server communication at high scale and high resilience.

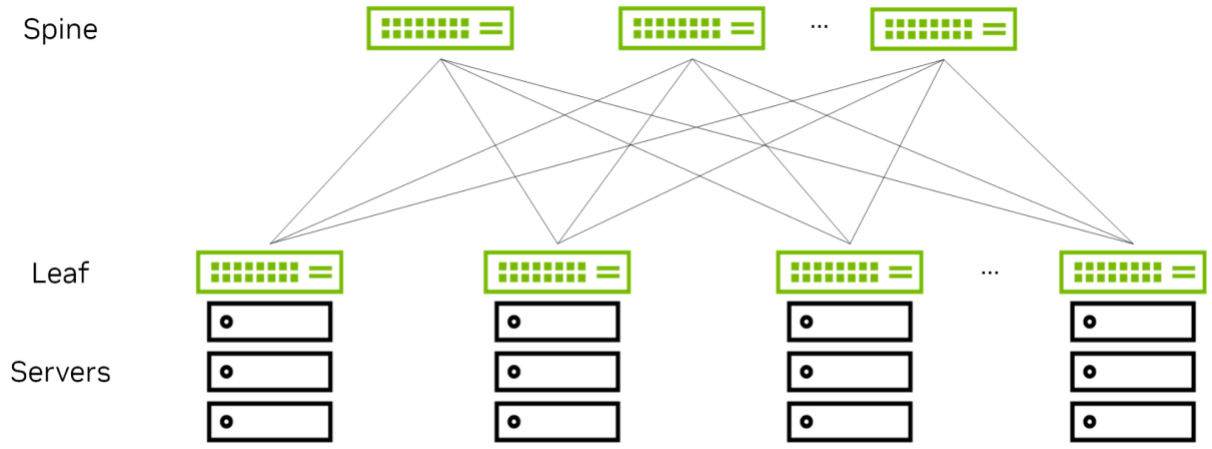
The traditional standard access, aggregation, and core layer architecture was suitable for north-to-south traffic flows that go in and out of a data center. This architecture was based on the L2 switching model. However, this model surfaced many disadvantages to network redundancy, scale, lack of TTL field in the L2 frames, and reliability in contrast to the robust L3 network

model. In addition, many applications operating inside of the data center required cross communication and the ability to talk amongst themselves. A cleaner design is to move from the L2 switching model to IP and network routing protocols.
The cloud native data center infrastructure pioneers picked a network topology called *CLOS* to fashion their data centers. Clos networks are named after their inventor, Charles Clos. The design allows you to build a network that is not limited by the scale of a single unit but can scale to a number of tiers. However, two tiers are used in a majority of the use cases.

Two-Tier Clos Architecture (Leaf-Spine)

Figure 1 shows the most common Clos two-tier topology.
The green nodes represent the switches and the black nodes the servers. There are two layers of switches: spine and leaf, therefore, the topology is commonly called a leaf-spine topology.

FIGURE 1 - TWO-TIER CLOS TOPOLOGY



The spine nodes connect the leaf nodes with one another and the leaf nodes connect the servers to the network. Every leaf is connected to every spine node.
This topology produces a high-capacity network because there are more than two paths between any two servers. Adding more spines or multiple links to the spine increases the available bandwidth between leaves, thanks to equal-cost multipath (ECMP).
The endpoints are all connected to leaves and the spines merely act as connectors. In this model, the functionality is pushed out to the edges instead of pulled into the spines. This model of scaling is called a *scale-out* model.
Typically, servers are interconnected to the leaf through lower-speed links and the switches are interconnected by higher-speed links. A common deployment is to interconnect servers to leaves through 25 Gbps links, while interconnecting switches with one another through 100 Gbps links. In AI/ML environments, you can also have higher bandwidth servers with 50/100G network interface cards and 200/400G interconnects.

Data Center Networking Concepts

Overlay, Underlay, and Tunneling

Network virtualization is the carving up of a single physical network into many virtual networks. Virtualizing a resource allows it to be shared by multiple users. In the case of virtual networks, each user is under the illusion that there are no other users of the network. To preserve the illusion, virtual networks are isolated from one another.

A virtual network implemented with protocols that leave the transit nodes unaware of it is called a *virtual network overlay*. An *underlay* is the network that transports the overlay network. Underlay networks can be L2 or L3 networks. L2 underlay networks today are typically based on Ethernet, with segmentation accomplished through VLANs. Internet is an example of an L3 underlay network, where autonomous systems use interior gateway protocols (IGPs) such as OSPF and IS-IS to run control planes, and use BGP as the Internet-wide routing protocol Multi-Protocol Label Switched (MPLS) networks are a legacy underlay WAN technology that falls between L2 and L3.Overlay networks implement network virtualization concepts, where L2 and L3 tunneling encapsulation (VXLAN, GRE, and IPSec) serves as the transport overlay protocol, sometimes referred to as OTV (Overlay Transport Virtualization).

In data centers, the role of the underlay is to provide reachability for the entire network. The underlay doesn't actually have any *intelligence* to keep track of the endpoints or define the end-to-end networking. It provides the ability for all devices in the network to communicate with each other. In overlay environments, routing information is typically aggregated in top-of-rack switches (for bare-metal endpoints) or server hypervisors (for virtualized workloads).
In an overlay virtual network, a tunnel endpoint is termed a *network virtualization edge* (NVE). The *ingress NVE*, which marks the start of the virtual network overlay, adds the tunnel header. The *egress NVE*, which marks the end of the virtual network overlay, strips off the tunnel header.

The tunnel header can be constructed using an L2 header or an L3 header. Examples of L2 tunnels include double VLAN tag (Q-in-Q or double-Q), TRILL, and Mac-in-Mac (IEEE 802.1ah). Popular L3 tunnel headers include VXLAN, IP Generic Routing Encapsulation (GRE) and Multiprotocol Label Switching (MPLS).

Benefits of using overlay

Some of the benefits of the virtual network overlays over non-overlays include:

- Scalability
  Virtual network overlays scale much better. Because the network core does not have to store forwarding table state for the virtual networks, it operates with much less state. As a consequence, a single physical network can support a larger number of virtual networks.

- Rapid provisioning
  Virtual network overlays allow for rapid provisioning of virtual networks. Rapid provisioning is possible because you configure only the affected edges, not the entire network.
- Reuse of existing equipment
  Only the edges participating in the virtual networks need to support the semantics of virtual networks. This also makes overlays extremely cost effective. If you want to try out an update to the virtual network software, only the edges need to be touched, whereas the rest of the network can hum along just fine.
- Independence from geographical location
  As long as end to end MTU permits, overlay networks can transport endpoint traffic across domains as if they're directly attached to each other. This makes disaster recovery and data replication very easy. As most of the modern overlay technologies are pure IP based and the whole internet supports IP, overlay networks allow you to interconnect domains over a shared environment.

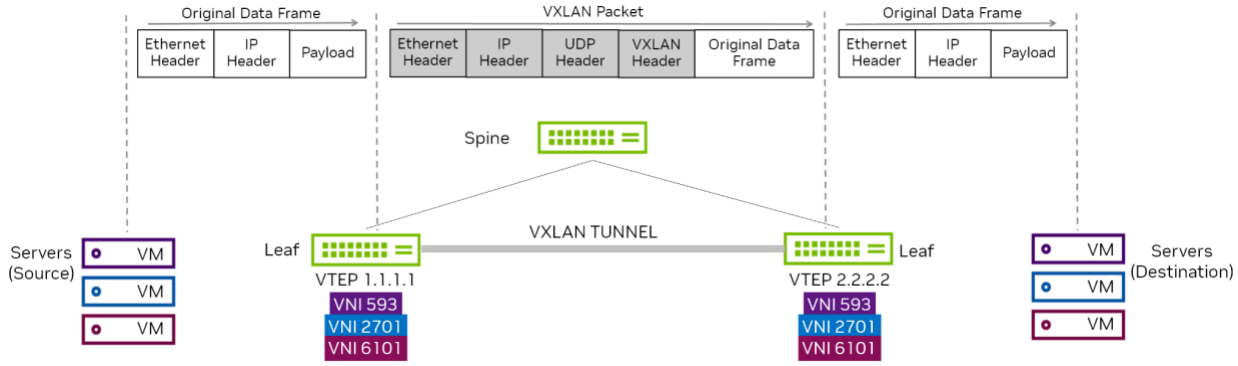
Virtual Extensible LAN

Many data centers today have moved from a legacy L2 design to a modern L3 architecture. L3 designs allow simplified troubleshooting, clear upgrade strategies, multi-vendor support, small failure domains, and less vendor lock-in. However, many applications, storage appliances and tenant considerations still require L2 adjacency.

Virtual Extensible LAN (VXLAN) is widely deployed in many L3 data centers to provide L2 connectivity between hosts for specific applications. This is done by encapsulating L2 frames in L3 packets. VXLAN is an *Overlay Technology* as it allows you to stretch L2 connections over an intervening L3 network by encapsulating (tunneling) Ethernet frames in an IP-UDP packet with a VXLAN header.

When a host sends traffic that belongs to a VNI (VXLAN Network Identifier), as shown in figure 2, the traffic is encapsulated in UDP and IP headers. This is then sent across the underlay network, just like normal IP traffic. When the packet reaches the destination switch, the packet is decapsulated and delivered to the destination server.

FIGURE 2 - VXLAN COMMUNICATION



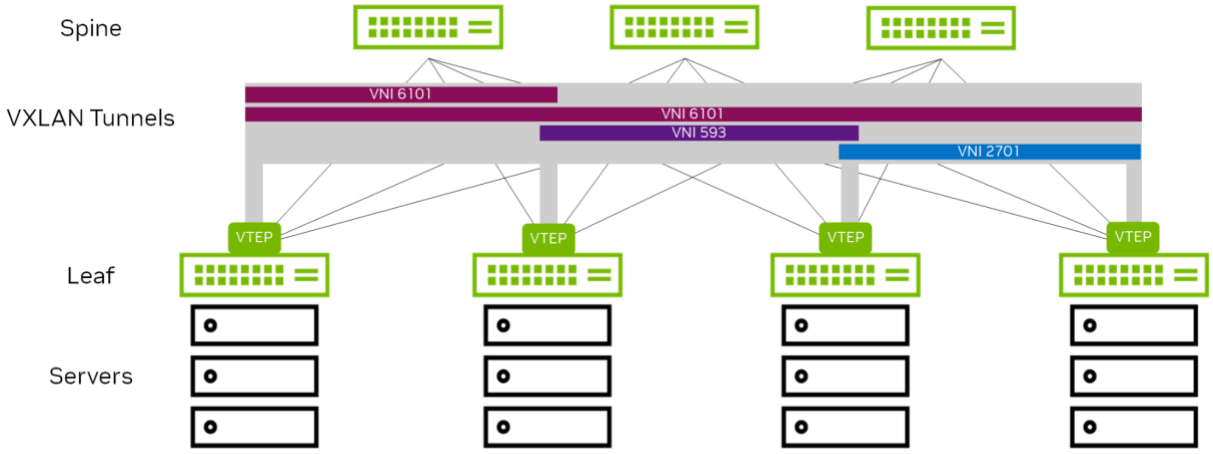
In the 2 tier leaf-spine topology, the leaf switches handle all the VXLAN functions that include creating the virtual networks and mapping VLANs to VNIs. The spine switches just pass traffic and are unaware that VXLAN even exists. By using VXLAN here, scaling the overlay network does not affect the underlay, and the other way round.

Virtual Tunnel Endpoints

Virtual Tunnel Endpoints (VTEPs) are used to originate and terminate the VXLAN tunnel, and map a VLAN to a VNI and a VNI to a VLAN. As shown above in Figure 2, a VTEP is an edge device on a VXLAN network. It is either the start of a VXLAN tunnel, where the user data frame is encapsulated, or the end point of a VXLAN tunnel, where the user data frame is decapsulated.

These are the top-of-rack switches (for bare-metal endpoints) and server hypervisors (for virtualized workloads). A VTEP requires an IP address (often a loopback address) and uses this address as the source/destination tunnel IP address. The VTEP IP address must be advertised into the routed domain so the VXLAN tunnel endpoints can reach each other. You can have multiple VNIs (VXLANs) using one VTEP IP address. Each switch that hosts a VTEP must have a VXLAN-supported chipset, such as Spectrum. VXLAN is a point-to-multipoint tunnel. Multicast or broadcast packets can be sent from a single VTEP to multiple VTEPs in the network.

FIGURE 3 - VXLAN TUNNEL AND VTEPs WITHIN A DATA CENTER



Benefits of using VXLAN

VXLAN is an overlay technology that uses encapsulation to allow L2 overlay VLANs to span across L3 networks. L2 networks have some inherent disadvantages:

- Because they rely on STP, the capability for redundancy and multiple paths is limited by the functionality of spanning tree.
- Using STP downscales the size and radius of an L2 segment due to characteristics and limitations of STP itself.
- Due to its characteristics, an L2 broadcast domain also defines the blast radius of a network. The larger L2 domain (with or without STP being used) also refers to the larger blast-radius.
- STP convergence is very slow.
- Redundancy is normally only limited to two devices due to MLAG.

VXLAN overcomes these deficiencies and allows the network operator to optimize on an L3 routed fabric. A L2 overlay can still be accomplished, but no longer requires STP for control plane convergence due to the reliance of EVPN as the control plane. EVPN exchanges MAC information through a BGP address family, instead of relying on the inefficiencies of broadcast flood and learn. Plus, VXLAN uses a 24-bit ID that can define up to 16 million virtual networks, whereas VLAN only has a 12-bit ID and is limited to 4094 virtual networks.

Border Gateway Protocol

Border Gateway Protocol (BGP) is the routing protocol that runs the Internet. It manages how packets get routed from network to network by exchanging routing and reachability information. BGP is an increasingly popular protocol for use in the data center as it lends itself well to the rich interconnections in a Clos topology.

BGP directs packets between autonomous systems (AS), which are a set of routers under a common administration. Each router maintains a routing table that controls how packets are

forwarded. Because BGP was originally designed to peer between independently managed enterprises and service providers, each such enterprise is treated as an AS responsible for a set of network addresses. Each such AS is given a unique number called an autonomous system number (ASN).

The ASN is central to how BGP builds a forwarding topology. A BGP route advertisement carries with it not only the ASN of the originator, but also the list of ASNs that this route advertisement passes through. When forwarding a route advertisement, a BGP speaker adds itself to this list. This list of ASNs is called the AS path. BGP uses the AS path to detect and avoid loops.

When you use BGP to peer between autonomous systems, the peering is eBGP. When you use BGP within an autonomous system, the peering is iBGP. eBGP peers have different ASNs while iBGP peers have the same ASN. The recommendation is to use eBGP for EVPN deployments.

Auto BGP

In a two-tier leaf and spine environment, you can use Auto BGP to generate 32-bit ASNs automatically so that you do not have to think about which numbers to configure. Auto BGP helps build optimal ASN configurations in your data center to avoid suboptimal routing and path hunting, which occurs when you assign the wrong spine ASNs. Auto BGP makes no changes to standard BGP behavior or configuration.

To use auto BGP to assign an ASN automatically on the leaf:

```
nv set router bgp autonomous-system leaf
```

To use auto BGP to assign an ASN automatically on the spine:

```
nv set router bgp autonomous-system spine
```

The auto BGP `leaf` and `spine` keywords are only used to configure the ASN. The configuration files and `nv show` commands display the AS number.

BGP unnumbered

One of the requirements for BGP to establish peering is to have IP addresses configured for L3 communication. This requires IPv4 and IPv6 address configuration on links connecting neighboring routers, which in a large network can consume a lot of address space and can require a separate IP address for each peer-facing interface. The BGP neighbors only connect and exchange routes when all of the configurations are correct. Configuring BGP in large data centers can be repetitive, time-consuming, and error-prone. BGP unnumbered helps to avoid these issues.

The BGP unnumbered standard in RFC 5549 uses ENHE and does not require that you advertise an IPv4 prefix together with an IPv4 next hop. You can configure BGP peering

between your Cumulus Linux switches and exchange IPv4 prefixes without having to configure an IPv4 address on each switch; BGP uses unnumbered interfaces.

The next hop address for each prefix is an IPv6 link-local address, which BGP assigns automatically to each interface. Using the IPv6 link-local address as a next hop instead of an IPv4 unicast address, BGP unnumbered saves you from having to configure IPv4 addresses on each interface.

The following example commands show a basic BGP unnumbered configuration for two switches, leaf01 and spine01, which are eBGP peers. As seen below, the only difference in a BGP unnumbered configuration is that the BGP neighbor is an interface and not an IP address.

```
# leaf01 configuration
nv set router bgp autonomous-system 65101
nv set router bgp router-id 10.10.10.1
nv set vrf default router bgp neighbor swp51 remote-as external
nv set vrf default router bgp address-family ipv4-unicast network
10.10.10.1/32
nv set vrf default router bgp address-family ipv4-unicast network
10.1.10.0/24
nv config apply
```

```
# spine01 configuration
nv set router bgp autonomous-system 65199
nv set router bgp router-id 10.10.10.101
nv set vrf default router bgp neighbor swp1 remote-as external
nv set vrf default router bgp address-family ipv4-unicast network
10.10.10.101/32
nv config apply
```

Design Considerations

- Use auto BGP in new deployments to avoid conflicting ASNs in an existing configuration.
- BGP unnumbered simplifies configuration and is recommended to be used in data center deployments.

Route Distinguisher (RD) and Route Target (RT)

Route Distinguisher

Virtual networks allow the reuse of an address. In other words, an address is unique only within a virtual network. A common, well understood example of this is the use of the 10.x.x.x subnet in IPv4. The 10.x address space is a private address space, so different organizations can reuse the address with impunity. Similarly, different virtual networks can reuse the same 10.x IPv4 address. This is true also for L2 addresses. So, BGP needs to keep the Building Blocks of Ethernet VPN separate from the advertisement of an address in one virtual network from the

advertisement of the same address in a different virtual network. That is the job of a Route Distinguisher (RD).

When exchanging VPN addresses, BGP prepends an 8-byte RD to every address. This combination of RD plus address makes the address globally unique. The RD format used for this purpose (RFC4364) is IP address + a unique number in the format of x.x.x.x:y this gives overlapping IP space uniqueness across EVPN domain, hence RD used in an EVPN domain must be unique.

Route Target

BGP advertisements carry path attributes, which provide extra information about a network address. They carry information such as the next hop IP address for a prefix, whether to propagate an advertisement, and so on, as encoded bits. Path attributes take several forms, including well-known attributes, communities, and extended communities. Route Target (RT) is a specific path attribute that encodes the virtual network it represents. A BGP speaker advertising virtual networks and their addresses uses a specific RT called the export RT. A BGP speaker receiving and using the advertisement uses this RT to decide into which local virtual network to add the routes. This is called the import RT. In a typical VPN configuration, you must configure both import and export RTs.

RD, RT, and BGP Processing

RD and RT both identify the virtual network from which a packet arrives. Every BGP implementation maintains two kinds of routing tables: a global one and one per virtual network. BGP runs the best-path algorithm on the global table to pick a single path to advertise for each prefix to its peers. Because the RD is unique to each originator, all copies of a route are advertised to a neighbor. To install routes into a virtual network's routing table, BGP first uses the import RT clause to select specific candidate routes from the global table to import into this virtual network. Then, it runs the best-path algorithm again on the imported candidate routes, but this time within the context of the virtual network's routing table. If the same address is advertised with multiple RTs, the best-path algorithm selects the best one.

Auto RD/RT

RD and RT are automatically generated using VLAN/VXLAN and VRF. When Free Range Routing (FRR) learns about a local VNI and there is no explicit configuration for that VNI in FRR, the switch derives the RD, and import and export RTs for this VNI automatically. The RD uses *RouterId:VNI-Index* and the import and export RTs use *AS:VNI*. For routes that come from an L2 VNI (type-2 and type-3), the RD uses the VXLAN local tunnel IP address (VXLAN-local-tunnelip) from the L2 VNI interface instead of the RouterId (VXLAN-local-tunnelip:VNI). EVPN route exchange uses the RD and RTs.
Cumulus Linux treats the import RT as *:VNI to determine which received routes apply to a particular VNI. This only applies when the switch auto-derives the import RT. If you do not want to derive RDs and RTs automatically or have an existing deployment with non-Cumulus

switches that do not support auto RTs or work with a different schema, you can define them manually.

Ethernet Virtual Private Network

Ethernet Virtual Private Network (EVPN) is a feature offered with Cumulus Linux that provides a scalable, interoperable end-to-end control-plane solution using Border Gateway Protocol (BGP). It is a standards-based protocol that can carry both L2 MAC and L3 IP information simultaneously to optimize routing and switching decisions. This control plane technology uses Multiprotocol BGP (MP-BGP) for MAC and IP address endpoint distribution, in turn minimizing flooding.

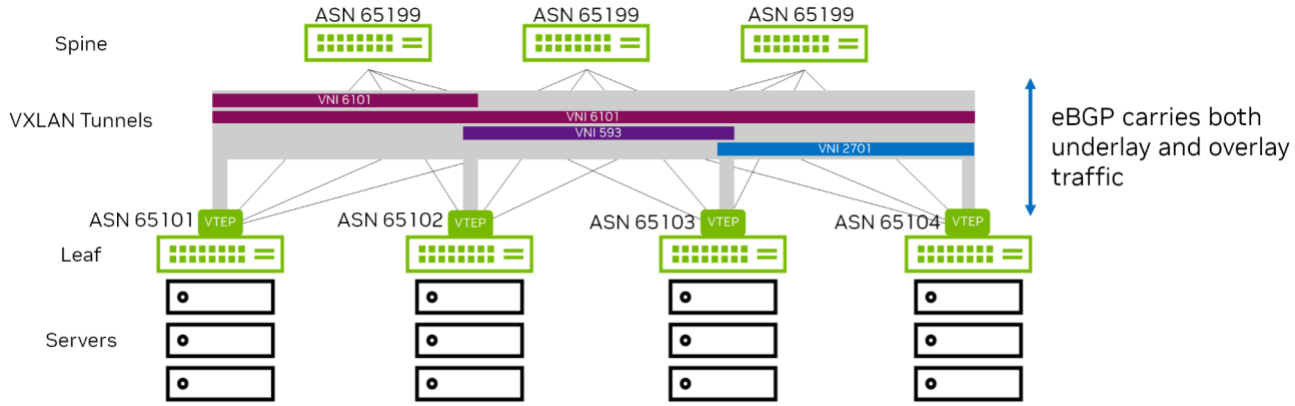
EVPN supports redundancy, load sharing, and multi-tenant segmentation. It provides virtual multi-point bridged connectivity between different L2 domains over an IP or IP/MPLS backbone network. EVPN also provides the benefit of fast convergence for host and VM mobility over VXLAN tunnels and ARP suppression.

With the advent of VXLAN in the data center, EVPN was adopted as the solution for network virtualization in the data center. Cumulus Linux supports the EVPN address family with both eBGP and iBGP peering. In a typical 2-tier Clos network where the leafs are VTEPs, if you use BGP sessions between the leafs and spines for underlay routing, the same sessions exchange EVPN routes. The spine switches act as route forwarders and do not install any forwarding state as they are not VTEPs. When the switch exchanges EVPN routes over iBGP peering, you can use OSPF as the IGP or resolve next hops using iBGP.

As shown in Figure 4, a typical data center uses just a few eBGP elements:

- Assign a unique identifier (AS number) to each logical group of devices.
    - Assign each leaf its own identifier (AS number)
    - Assign each group of spines (all the spines connected to the same set of leafs) a common identifier (AS number).
    - Assign each group of super-spines (connected to the same set of spines) a common identifier (AS number).
- Set up peering between spine and leaf devices
- Add a simple routing policy to share loopback address information
- Add a routing policy to disallow leaf switches as transit devices
- Enable load balancing

FIGURE 4 - EVPN DEPLOYMENT WITH eBGP



Benefits of deploying EVPN

EVPN is a standardized control plane protocol that offers controller-less VXLAN tunnels. It also offers scale, redundancy, fast convergence and robustness while reducing broadcast, unknown unicast, and multicast (BUM) traffic across a data center core. Deploying EVPN provides many advantages to an L3 data center:

- Simplicity
  EVPN uses the BGP routing protocol. BGP is also the preferred routing protocol for data center infrastructures. The same routing protocol can be used for both infrastructure and virtual topologies.
- Controller-less VXLAN tunnels
  No controller is needed for VXLAN tunnels, as EVPN provides peer discovery with authentication natively. This also mitigates the chance of rogue VTEPs in a network and dealing with complicated controller redundancy, and scaling issues caused by controller.
- ARP Suppression
  Cumulus EVPN reduces broadcast traffic within a data center by allowing the local leaf switch to respond to a host's ARP requests instead of forwarding throughout the data center. Cumulus Linux enables ARP suppression by default.
- Scale and robustness
  EVPN uses the BGP routing protocol. BGP is very mature, scalable, flexible and robust. It is the primary routing protocol for the Internet and data centers. It supports routing policy and filtering, which provides granular control over traffic flow.
- Fast convergence and host mobility
  Cumulus EVPN supports the new BGP MAC mobility extended community, offering fast convergence and reducing discovery traffic after a MAC or VM move. MAC stickiness is also supported, preventing specific host mobility if desired.
- Support for VXLAN active-active mode
  Cumulus EVPN integrates with MLAG and multihoming, thereby providing host dual homing for redundancy.
- Multitenancy
  EVPN uses the RDs and RTs to separate tenants within a data center.

- VXLAN Routing
  Cumulus EVPN supports IP routing between VXLAN VNIs in overlay networks and is supported with Spectrum chipsets. VXLAN routing within a VRF is also supported.
- Interoperability between vendors
  The standardized multiprotocol BGP (MP-BGP) is used for the EVPN control plane. As long as vendor implementations maintain adherence to both the VXLAN and EVPN standards, interoperability is assured.

EVPN Route Types

Table 1 shows the different Route Types (RTs) used in EVPN. The minimum required RTs needed to operate an EVPN network are RT-2, RT-3, and RT-5. The rest are optional and dependent on the choices you make in building your network.

Route Type	What it carries	Primary use
Type 1	Ethernet Segment Auto Discovery	Used in the data center in support of multihomed endpoints.
Type 2	MAC, VNI, IP	Advertises reachability to a specific MAC address, and optionally its IP address.
Type 3	Inclusive Multicast Route	Required for Broadcast, Unknown Unicast and Multicast (BUM) traffic delivery across EVPN networks - provides information about P-tunnels that should be used to send the traffic.
Type 4	Ethernet Segment Route	Discovers VTEPs attached to the same Ethernet Segment and for Designated Forwarder Election used for BUM traffic.
Type 5	IP Prefix, L3 VNI	Advertises prefix (not /32 or /128), routes such as summarized routes in a virtual L3 network.
Type 6	Multicast group membership info	Information about interested multicast groups derived from IGMP.
Type 7	Multicast Membership Report Synch Route	IGMP synchronization mechanism that would allow all PE devices serving given ES to share their state - this route is used to coordinate the IGMP Membership Report.

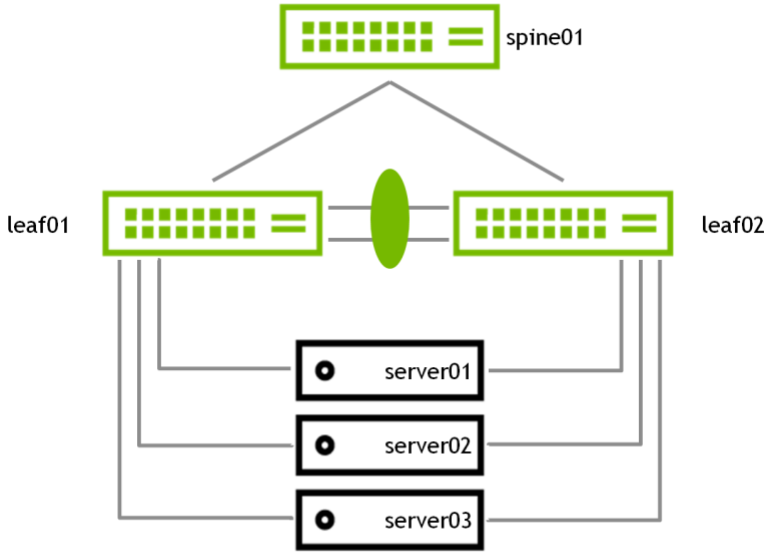
Type 8	Multicast Leave Synch Route	IGMP synchronization mechanism that would allow all PE devices serving given ES to share their state - this route is used to coordinate the IGMP Leave Group.
Type 9	Per-region I-PMSI Auto Discovery	Auto-Discovery routes used to announce the tunnels that instantiate an Inclusive PMSI - to all PEs in the same VPN.
Type 10	S-PMSI Auto Discovery	Auto-Discovery routes used to announce the tunnels that instantiate a Selective PMSI - to some of the PEs in the same VPN.
Type 11	Leaf Auto Discovery	Used for explicit leaf tracking purposes. Triggered by I/S-PMSI A-D routes and targeted at triggering route's (re-)advertiser.

Multi-Chassis Link Aggregation and Multihoming

Multi-Chassis Link Aggregation (MLAG)

MLAG enables a pair of switches to act redundantly in an active-active architecture and appear as a single, logical device from the perspective of the host. The two switches in an MLAG pair are connected by a link or bonded links called the *peer link*. In a basic MLAG configuration, as shown in Figure 5, leaf01 and leaf02 are MLAG peers. MLAG is on three bonds, each with a single port, a peer link that is a bond with two member ports, and three VLANs on each port.

FIGURE 5 - BASIC MLAG CONFIGURATION



VRR (Virtual Router-Redundancy) enables a pair of switches to act as a single gateway for High Availability (HA) and Active-Active server links. VRR enables hosts to communicate with any redundant switch without reconfiguration by running dynamic router protocols or router redundancy protocols. Redundant switches respond to ARP requests from hosts. The switches respond in an identical manner, but if one fails, the other redundant switches continue to respond.

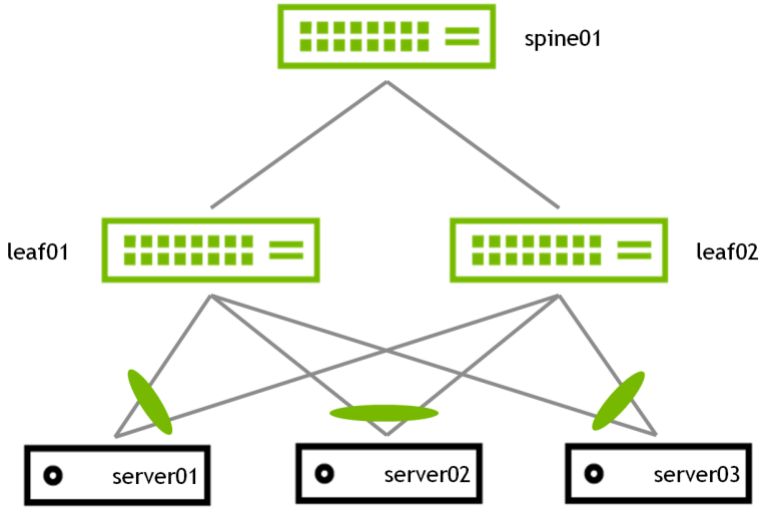
A device that connects to an MLAG bond believes there is a single device on the other end of the bond and only forwards one copy of the transit frames. With the virtual MAC active on both MLAG devices, either MLAG device handles the frame it receives. Cumulus Linux supports both VRR and VRRP. VRRP allows two or more network devices in an active or standby configuration to share a single virtual default gateway. However, VRRP cannot be used in an EVPN configuration.

Multihoming

EVPN Multihoming (EVPN-MH) is a standards-based replacement for the proprietary MLAG protocol in data center deployments. It provides an all-active server connectivity without the need for peer links between ToR switches. EVPN-MH enables multi-vendor interoperability with a single BGP-EVPN control plane. This protocol allows easier data center deployments without the need of understanding and using proprietary protocols.

EVPN-MH uses BGP-EVPN type-1, type-2 and type-4 routes to discover Ethernet segments (ES) and to forward traffic to them. The MAC and neighbor databases synchronize between the ES peers through these routes as well. An *ES* is a group of switch links that attach to the same server. As seen in Figure 6, EVPN-MH eliminates the need for peer links or inter-switch links between the top of rack switch.

FIGURE 6 - BASIC EVPN-MH CONFIGURATION



Design Considerations

Sometimes MLAG is still required in VXLAN environments for redundant host connectivity. EVPN-MH is an opportunity to move off proprietary MLAG solutions. Most MLAG systems allow dual homing across only two paths. In practice, MLAG systems are limited to dual core switches because it is extremely difficult to maintain a coherent state between more than two devices with sub microsecond refresh times. EVPN-MH, on the other hand, can scale beyond two leaf switches. VXLAN helps remove the need for back-to-back leaf-to-spine switch connections as required by MLAG. EVPN-MH goes one step further and eliminates any need for MLAG in server-to-leaf connectivity.

Multihoming uses EVPN messages to communicate host connectivity, and it dynamically builds L2 adjacency to servers using host connectivity information. Where MLAG requires LAG IDs, multihoming uses Ethernet segment IDs. Interfaces are mapped to segments that act like logical connections to the same end host. Additionally, moving to multihoming improves network vendor interoperability by using a protocol standard form of redundancy in the switch. Because multihoming is part of the EVPN address family, an open standard protocol, any vendor implementing multihoming through the RFC specification can be part of the Ethernet segment.

The switch selects a designated forwarder (DF) for each Ethernet segment. The DF forwards flooded traffic received through the VXLAN overlay to the locally attached Ethernet segment. You need to specify a preference on an Ethernet segment for the DF election, as this leads to predictable failure scenarios. The EVPN VTEP with the highest DF preference setting becomes the DF.

MLAG uses both uplinks at the same time. VRR enables both devices to act as gateways simultaneously for HA (high availability) and active-active mode (both are used at the same time).

The disadvantages of using MLAG are:

- More complicated (more moving parts)
- More configuration
- No interoperability between vendors
- ISL (inter-switch link) required
- MLAG can be formed with a maximum of 2x leaf switches

Where possible, EVPN-MH is recommended over MLAG.

The advantages of using EVPN MH:

- You don’t need ISL anymore
- You can use BGP everywhere
- It is a standards-based implementation that can be used in a multivendor environment
- It can be formed with more than 2x leaves and can create more than two multihomed server-to-leaf connectivity for active-active load balancing and resiliency

To configure EVPN-MH:

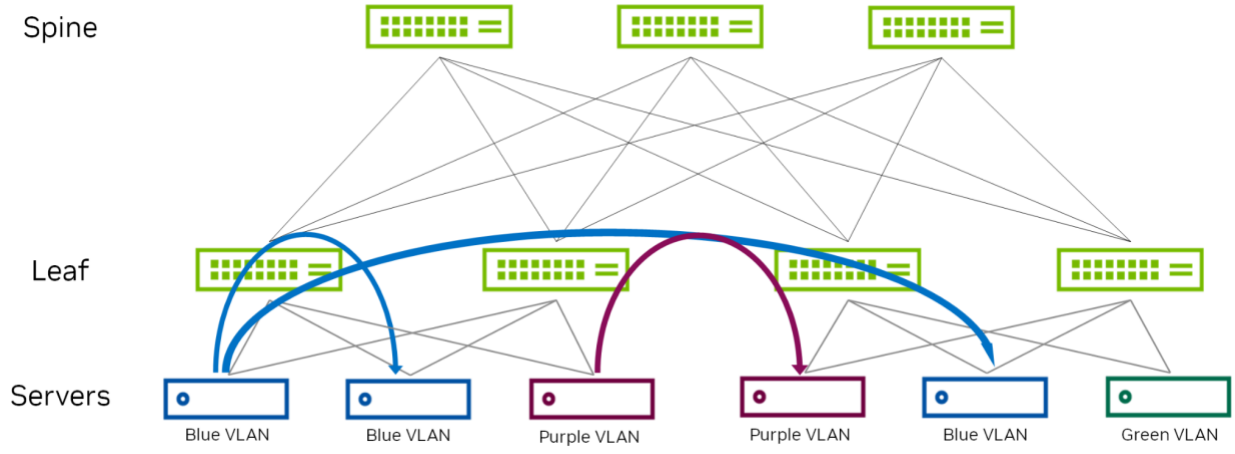
```
# Enable EVPN multihoming
nv set evpn multihoming enable on

# Set the Ethernet segment ID
nv set interface bond1 bond member swp1
nv set interface bond2 bond member swp2
nv set interface bond3 bond member swp3
nv set interface bond1 evpn multihoming segment local-id 1
nv set interface bond2 evpn multihoming segment local-id 2
nv set interface bond3 evpn multihoming segment local-id 3

# Set the Ethernet segment system MAC address
nv set interface bond1-3 evpn multihoming segment mac-address
44:38:39:BE:EF:AA
nv set interface bond1-3 evpn multihoming segment df-preference 50000

# Configure multihoming uplinks
nv set interface swp51-54 evpn multihoming uplink on
nv config apply
```

EVPN Deployment Scenarios

EVPN for L2 Deployments

L2 EVPN deployment uses a bridged overlay as seen in Figure 7. It provides Ethernet bridging in an EVPN network and extends VLANs between the leaf devices across VXLAN tunnels. These leaf-to-leaf VXLAN tunnels are useful in networks that require connectivity between leaf devices but do not need inter-VLAN routing. As a result, the *intelligence* is at the leaf layer. The spine layer simply provides connectivity between leaf devices. Leaf devices establish VTEPs to connect to other leaf devices. The tunnels enable communication between leaf devices and Ethernet-connected end systems in the data center.

FIGURE 7 - L2 EVPN EXAMPLE



STP is a L2 switching protocol, which ensures loop-free network topologies by always creating a single path tree structure through the network. If there is a link failure or the network is reconfigured, all network traffic is halted while the spanning tree algorithm recalculates the allowed loop-free paths. However, this is no longer necessary due to the inherent split-horizon functionality.

Points to consider:

- This is useful when L2 domains are divided by L3 fabrics and need to be stretched over them (such as legacy L2 applications, ESF, and so on.)
- Each ToR (leaf) is a VTEP and hosts the VLANs (mapped to VNIs) located on its rack.
- To have an extended L2 domain, the specific VNI has to be configured on the applicable VTEPs.
- Using this type of environment doesn't allow inter-VLAN connectivity. For routing between different VNIs, we need to look at L3 deployment models or an external gateway, outside the fabric in order to perform routing between VLANs.

Scenarios to consider using L2 EVPN:

- When you have the subnet across the racks in a data center.
- When the architecture uses a firewall as the gateway. For example, if your security policy defines that all inter-VLAN traffic must go through a firewall, the L3 gateway functionality is provided by design outside the fabric, so the bridged overlay architecture is a good fit.
- When you have an existing Ethernet-based data center network and want to introduce EVPN/VXLAN. Because the bridged overlay approach is so basic and simple, it’s a good option when you want to modernize your DC environment, but you want to take a phased or incremental approach.

EVPN for L3 Deployments

Traditionally, data centers have used L2 technologies such as STP and MLAG. As data centers evolve and expand, they tend to outgrow their limits; xSTP blocks ports, which locks out needed bandwidth, while MLAG may not provide enough redundancy. Additionally, a device outage is a significant event and larger modular-chassis devices use a lot of power.

Routing Models

You might have to communicate between L2 domains and between a VXLAN tunnel and the outside world, for which you can enable VXLAN routing in the network.
VXLAN routing can be performed with one of two architectures:

- Centralized routing performs all the VXLAN routing on one or two centralized routers (routing on the border leaves), which is a good option for data centers with a lot of north-south traffic. This can cause additional east-west traffic in the data center.
- Distributed routing provides the VXLAN routing closest to the hosts on the directly connected leaf switches (routing in the leaf layer), which is a good option for data centers with a lot of east-west traffic. This simplifies the traffic flow.

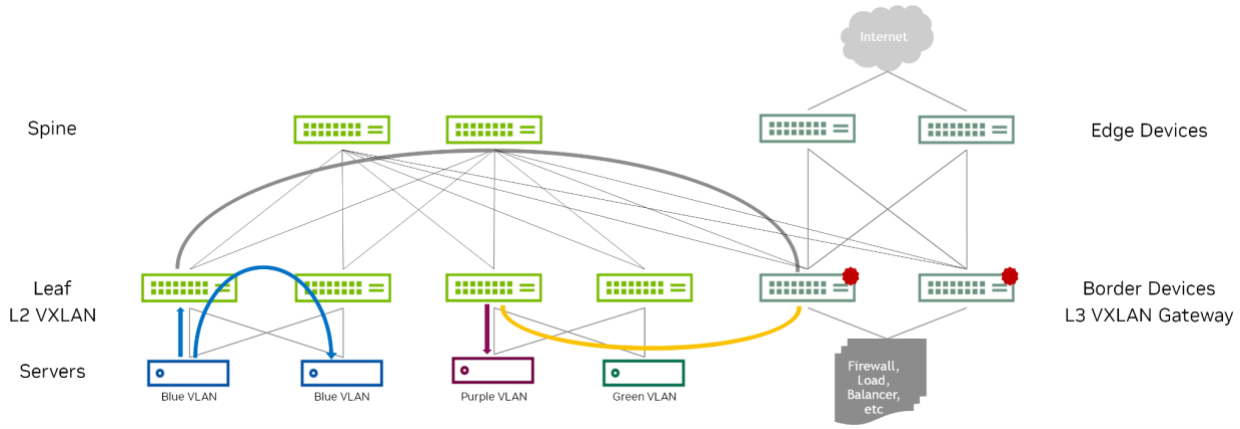
BGP EVPN is used to communicate the VXLAN L3 routing information to the leaves.

Centralized routing

The nature of a centrally routed bridging overlay is that routing occurs at a central gateway within the data center network (the border leaves, in this example) rather than at the VTEP device where end systems are connected (the leaf layer, in this example). You can use this overlay model when traffic needs to be routed through a centralized gateway or when edge VTEP devices lack the required routing capabilities.

Figure 8 shows a common way to deploy this model. Border devices are located at the edge, or border, of a data center fabric. These devices also act as the VTEP for north-south traffic entering and exiting the network fabric. The traffic that originates at the Ethernet-connected end systems is forwarded to the leaf VTEP devices over a trunk (multiple VLANs) or an access port (single VLAN). The VTEP device forwards the traffic to local end systems or to an end system at a remote VTEP device. An integrated routing and bridging (IRB) interface at the border devices routes traffic between the Ethernet virtual networks.

FIGURE 8 - CENTRALIZED IRB EXAMPLE



The main disadvantages of this approach are scalability and potentially non-optimal traffic flow.

Scenarios for using centralized IRB:

- The need for inter-VXLAN routing to happen within the fabric. This approach has the advantage of centralizing and consolidating the routing function (instead of distributing it at the leaf layer).
- The architecture is optimized for data centers running mostly north-south traffic.

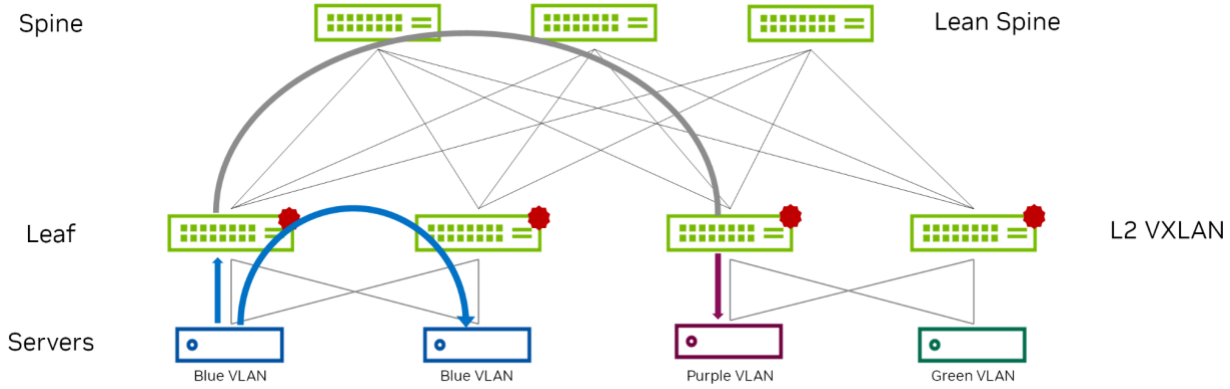
Distributed routing

This model enables faster server-to-server intra-data center traffic (also known as east-west traffic). Because the endpoints are connected to the same leaf device VTEP, routing occurs much closer to the end systems than with the centralized IRB model. It also allows for a simpler overall network. The spine devices are configured to handle IP traffic only, removing the need to extend bridging overlays to the spine devices.
Using the distributed architecture, the Internet Engineering Task Force (IETF) defines two models to accomplish intersubnet routing with EVPN: asymmetric (IRB) and symmetric IRB. With Cumulus Linux, depending on the requirement, you can choose either of the methods.

Symmetric IRB

This is the default EVPN routing model.
The symmetric model routes and bridges on both the ingress and the egress leaves. This results in bidirectional traffic being able to travel on the same VNI, hence the symmetric name. However, a new specialty transit VNI is used for all routed VXLAN traffic, called the L3VNI. All traffic that must be routed is routed onto the L3VNI, tunneled across the L3 infrastructure, routed off the L3VNI to the appropriate VLAN, and ultimately bridged to the destination. Figure 9 shows bridging and routing in a sample symmetric configuration.

FIGURE 9 - SYMMETRIC IRB EXAMPLE



Points to consider:

- The leaf switches only need to host the VLANs and the corresponding VNIs that are located on its rack, as well as the L3VNI and its associated VLAN. The ingress leaf switch doesn't need to know the destination VNI.
    - The ability to host only the local VNIs (plus one extra) helps with scale.
- The configuration is more complex as an extra VXLAN tunnel and VLAN in your network are required.
- Multitenancy requires one L3VNI per VRF, and all switches participating in that VRF must be configured with the same L3VNI. The L3VNI is used by the egress leaf to identify the VRF in which to route the packet.

Scenarios for using Symmetric IRB:

- All use cases of EVPN fabric within the data center except where there is need for a centralized gateway.
- Deployments where the network fabric has non-EVPN routes, such as default routes, static routes, or dynamic routes.
- Large-scale EVPN deployments.
    - Widely dispersed VLANs, subnets, or VNIs.

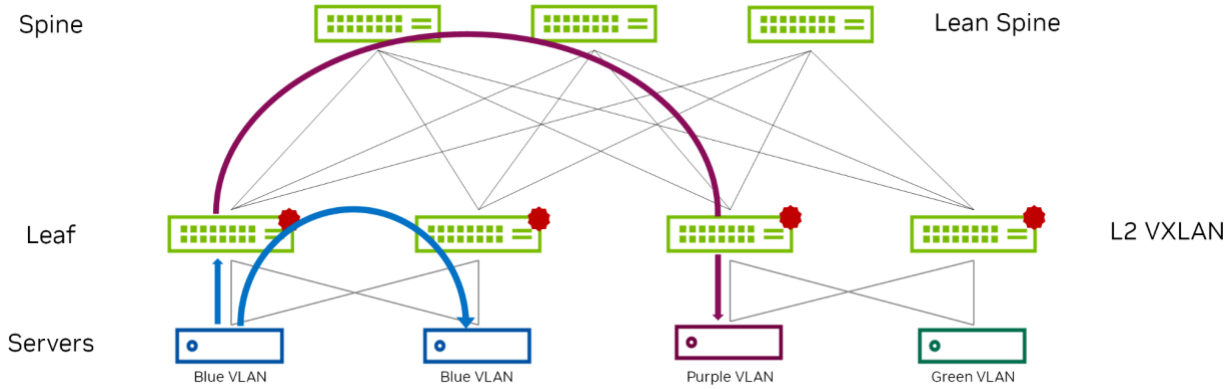
Symmetric VXLAN routing is configured directly on the ToR, using EVPN for both VLAN and VXLAN bridging as well as VXLAN and external routing. Each server is configured on a VLAN and MLAG or MH bond set up between servers and the leaves. Each leaf is configured with an anycast gateway and the servers' default gateways are pointing towards the corresponding leaf switch IP gateway address. Tenant VNIs (corresponding to the number of VLANs/VXLANs) are bridged to corresponding VLANs. The benefits of this approach include:

- The L2 domain is reduced to the pair of ToRs.
- The aggregation layer is all L3.
- Route scaling and flexibility.
- High availability.
- Overlay flows pass through the same VNI (transit VNI), providing a symmetrical overlay path, making it easy to monitor the flows.

Asymmetric IRB

The asymmetric model enables routing and bridging on the VXLAN tunnel ingress, but only bridging on the egress. This results in bidirectional VXLAN traffic traveling on different VNIs in each direction (always the destination VNI) across the routed infrastructure. Figure 10 shows bridging and routing in a sample asymmetric configuration. Even though this is supported by Cumulus Linux, Symmetric IRB is the recommended deployment model.

FIGURE 10 - ASYMMETRIC IRB EXAMPLE



Scenarios for using Asymmetric IRB:

- Preferred model for centralized gateway deployment.
- Networks that have switches with legacy ASICs that do not support L3 VXLAN and, therefore, must use centralized gateways.
- Small and medium scale data center deployments.
    - Simpler configurations such as all VLANs, subnets, or VNIs configured on all leaves. It's simpler to configure and doesn't require extra VNIs to troubleshoot.

Multi-tenancy and VRF

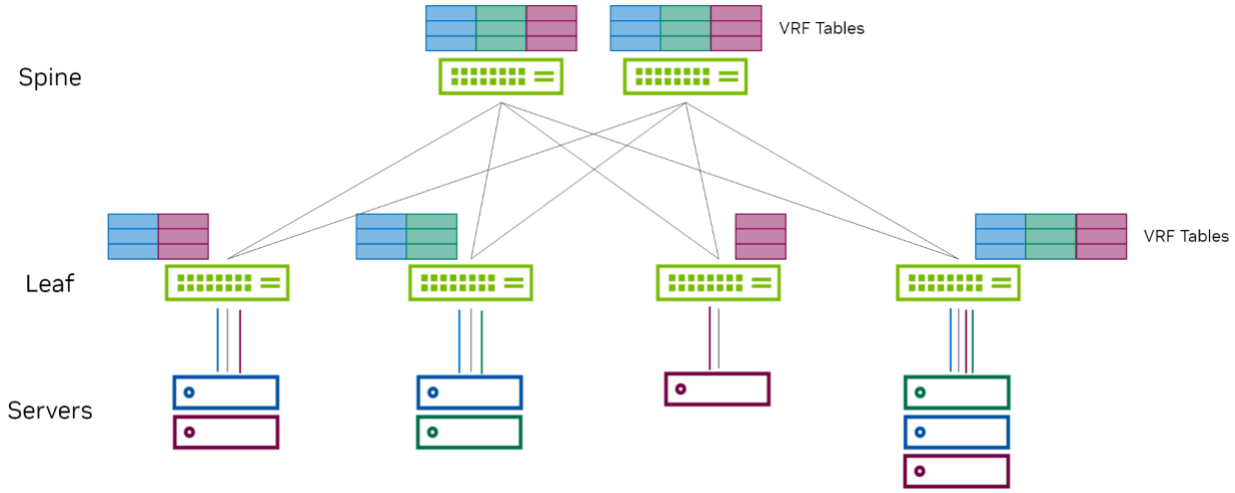
VRF segmentation is used to organize users and devices in groups on a shared network while separating and isolating the different groups. The routing devices on the network create and maintain separate virtual routing and forwarding (VRF) tables for each group. You can use VRFs in the data center to carry multiple isolated traffic streams for multi-tenant environments. Because the multiple routing instances are independent of each other and can use different outgoing interfaces, overlapping IP addresses do not cause any conflict, thus enabling multi-tenancy.

In EVPN, routing is assumed to occur within the context of a VRF. This is true regardless of whether the model is symmetric or asymmetric. The underlay routing table is assumed to be in the default or global routing table, while the overlay routing table is assumed to be in a VRF-specific routing table. It is possible to have asymmetric routing work without the use of VRFs. But VRFs are necessary if the endpoints have to communicate to the external world, because RT-5 advertisements are involved. RT-5 advertisements always occur in the context of a VRF:

the L3 VNI signaled in the advertisement. Therefore, to preserve a uniform routing model, always using VRFs with EVPN routing is recommended.

As shown in Figure 11, the servers in a group are placed in one VRF segment and can communicate with each other, but they cannot communicate with users in another VRF segment. If you want to send and receive traffic from one VRF segment to another VRF segment, you must configure route leaking or rely on an external gateway.

FIGURE 11 - MULTI-TENANCY USING VRF



Points to consider:

- You must configure the overlay (tenants) in a specific VRF and separate from the underlay, which resides in the default VRF. Cumulus Linux does not support L3 VNI mapping for the default VRF.
- A VRF table can have an IP address, which is a loopback interface for the VRF.
- Cumulus Linux adds the associated rules automatically.
- You can add a default route to avoid skipping across tables when the kernel forwards a packet.
- VRF table names can be a maximum of 15 characters. However, you cannot use the name mgmt; Cumulus Linux uses this name for the management VRF.
- Cumulus Linux supports up to 255 VRFs on the Spectrum 1 switch.

To configure a VRF *BLUE* and assign a table ID automatically:

```
nv set vrf BLUE table auto
nv set interface swp1 ip vrf BLUE
nv config apply
```

BGP Community Lists

You can use community lists to define a BGP community to tag one or more routes. You can then use the communities to apply a route policy on either egress or ingress. The BGP community list can be either standard or expanded. The standard BGP community list is a pair of values (such as *100:100*) that you can tag on a specific prefix and advertise to other

neighbors or you can apply them on route ingress. An expanded BGP community list takes a regular expression of communities and matches the listed communities.

When the neighbor receives the prefix, it examines the community value and takes action accordingly, such as permitting or denying the community member in the routing policy.BGP EVPN routes can have a set of EVPN extended communities carried in the BGP update message path attribute, and as such, you can use these extended communities for filtering BGP EVPN routes. The EVPN specific information available in extended communities includes, for example, encapsulation type, MAC-mobility information, EVPN split-horizon label information, EVPN ESI split-horizon label, ESI mode, E-tree leaf label, and more.
Here is an example of a standard community list filter:

```
nv set router policy community-list COMMUNITY1 rule 10 action permit
nv set router policy community-list COMMUNITY1 rule 10 community
100:100
nv config apply
```

Summarized Route Announcements

In EVPN symmetric routing configurations with VXLAN active-active (MLAG), all EVPN routes advertise with the anycast IP address as the next hop IP address and the anycast MAC address as the router MAC address. In a failure scenario, the switch might forward traffic to a leaf switch that does not have the destination routes. To prevent dropped traffic in this failure scenario, Cumulus Linux enables the Advertise Primary IP address feature by default so that the switch handles the next hop IP address of the VTEP conditionally depending on the route type: host type-2 (MAC and IP advertisement) or type-5 (IP prefix route).

- For host type-2 routes, the anycast IP address is the next hop IP address and the anycast MAC address is the router MAC address.
- For type-5 routes, the system IP address (the unique primary loopback IP address of the VTEP) is the next hop IP address and the unique router MAC address of the VTEP is the router MAC address.

Prefix-based Routing

The EVPN type-2 (MAC and IP) advertisement does not support advertising summarized or prefix routes such as /16 or /24 routes. This affects the scalability of the solution.
If you consider a network with edge devices, the edge devices commonly advertise only the default route to border devices. In just about every deployment, the spines and the leaves do not carry the routing table of the external world. They just carry the default route which gets them to the border devices and from there to the edge devices. The default route is 0.0.0.0/0, which has a non-/32 prefix (IPv6 has ::/0 as the default route).

A new route type, type-5 (RT-5) was introduced to support this use case. EVPN in Cumulus Linux supports prefix-based routing using EVPN type-5 (prefix) routes. Type-5 routes (or prefix routes) primarily route to destinations outside of the data center fabric. EVPN prefix routes carry

the L3 VNI and router MAC address and follow the symmetric routing model to route to the destination prefix.

Points to consider:

- When connecting to a WAN edge router to reach destinations outside the data center, deploy specific border or exit leaf switches to originate the type-5 routes.
- On switches with Spectrum ASICs, centralized routing, symmetric routing, and prefix-based routing only work with Spectrum-A1 and later.
- Configure a per-tenant VXLAN interface that specifies the L3 VNI for the tenant. This VXLAN interface is part of the bridge; router MAC addresses of remote VTEPs install over this interface.
- Configure an SVI (L3 interface) corresponding to the per-tenant VXLAN interface. This attaches to the VRF of the tenant. The remote prefix routes install over this SVI.
- Specify the mapping of the VRF to L3 VNI. This configuration is for the BGP control plane.

Scenarios for using prefix-based routing:

- Route to destinations outside of the data center fabric.
- To subdivide the data center into multiple pods with full host mobility within a pod but only do prefix-based routing across pods. You can achieve this by only exchanging EVPN type-5 routes across pods.
  The following example commands configure EVPN to advertise type-5 routes on the leaf:

```
nv set router policy route-map map1 rule 10 match type ipv4
nv set router policy route-map map1 rule 10 match evpn-route-
type ip-prefix
nv set router policy route-map map1 rule 10 action permit
nv set vrf default router bgp address-family ipv4-unicast
route-export to-evpn route-map map1
nv config apply
```

- To only exchange EVPN routes carrying a particular VXLAN ID. For example, if data centers or pods within a data center only share certain tenants, you can use a route map to control the EVPN routes exchanged based on the VNI.
  The following example configures a route map that only advertises EVPN routes from VNI 1000:

```
nv set router policy route-map map1 rule 10 match evpn-vni 1000
nv set router policy route-map map1 rule 10 action permit
nv config apply
```

- Cumulus Linux supports originating EVPN default type-5 routes. The default type-5 route originates from a border (exit) leaf and advertises to all the other leafs within the pod. Any leaf within the pod follows the default route towards the border leaf for all external traffic (towards the Internet or a different pod).

```
nv set vrf RED router bgp address-family ipv4-unicast route-
```

```
export to-evpn default-route-origination on
nv set vrf RED router bgp address-family ipv6-unicast route-
export to-evpn default-route-origination on
nv config apply
```

Route Leaking

VRFs are typically used when you want multiple independent routing and forwarding tables. To reach destinations in one VRF from another VRF, Cumulus Linux supports dynamic VRF route leaking. With route leaking, a destination VRF wants to know the routes of a source VRF. As routes come and go in the source VRF, they dynamically leak to the destination VRF through BGP. If BGP learns the routes in the source VRF, you do not need to perform any additional configuration. If OSPF learns the routes in the source VRF, if you configure the routes statically, or you need to reach directly connected networks, you need to redistribute the routes first into BGP (in the source VRF).

You can also use route leaking to reach remote destinations as well as directly connected destinations in another VRF. Multiple VRFs can import routes from a single source VRF and a VRF can import routes from multiple source VRFs. You can use this method when a single VRF provides connectivity to external networks or common services such as DHCP or DNS that are often delivered to multiple tenant VRFs. You can control the routes leaked dynamically across VRFs with a route map.

Because route leaking happens through BGP, the underlying mechanism relies on the BGP constructs of the Route Distinguisher (RD) and Route Targets (RTs). However, you do not need to configure these parameters; Cumulus Linux derives them automatically when you enable route leaking between a pair of VRFs.

Points to consider:

- You can assign an interface to only one VRF; Cumulus Linux routes any packets arriving on that interface using the associated VRF routing table.
- You cannot route leak overlapping addresses.
- You can use VRF route leaking with EVPN in a symmetric routing configuration only.
- You cannot use VRF route leaking between the tenant VRF and the default VRF with onlink next hops (BGP unnumbered).
- You cannot reach the loopback address of a VRF (the address assigned to the VRF device) from another VRF.
- You must use the redistribute command in BGP to leak non-BGP routes (connected or static routes); you cannot use the network command.
- Cumulus Linux does not leak routes in the management VRF with the next hop as eth0 or the management interface.
- You can leak routes in a VRF that iBGP or multi-hop eBGP learns even if their next hops become unreachable. NVIDIA recommends route leaking for routes that BGP learns through single-hop eBGP.

- You cannot configure VRF instances of BGP in multiple autonomous systems (AS) or an AS that is not the same as the global AS.
- Do not use the default VRF as a shared service VRF. Create another VRF for shared services.
- Run common services in a separate VRF (service VRF) instead of the default VRF to simplify configuration and avoid using route maps for filtering.
- To exclude certain prefixes from the import process, configure the prefixes in a route map.

An EVPN symmetric routing configuration has certain limitations when leaking routes between the default VRF and non-default VRFs. The default VRF has routes to VTEP addresses that you cannot leak to any tenant VRFs. If you need to leak routes between the default VRF and a non-default VRF, you must filter out routes to the VTEP addresses to prevent leaking these routes.

Downstream VNI enables you to assign a VNI from a downstream remote VTEP through EVPN routes instead of configuring L3 VNIs globally across the network. To configure a downstream VNI, you configure tenant VRFs as usual; however, to configure the desired route leaking, you define a route target import and, or export statement.
Scenarios for using route leaking:

- To make a service, such as a firewall available to multiple VRFs.
- To enable routing to external networks or the Internet for multiple VRFs, where the external network itself is reachable through a specific VRF.

In the following example commands, routes in the BGP routing table of VRF *BLUE* dynamically leak into VRF *RED*.

```
nv set vrf RED router bgp address-family ipv4-unicast route-import
from-vrf list BLUE
nv config apply
```

The following example configures a route map to match the source protocol BGP and imports the routes from VRF *BLUE* to VRF *RED*. For the imported routes, the community is *11:11* in VRF *RED*.

```
nv set vrf RED router bgp address-family ipv4-unicast route-import
from-vrf list BLUE
nv set router policy route-map BLUEtoRED rule 10 match type ipv4
nv set router policy route-map BLUEtoRED rule 10 match source-
protocol bgp
nv set router policy route-map BLUEtoRED rule 10 action permit
nv set router policy route-map BLUEtoRED rule 10 set community 11:11
nv set vrf RED router bgp address-family ipv4-unicast route-import
from-vrf route-map BLUEtoRED
nv config apply
```

ARP Suppression

ARP suppression with EVPN allows a VTEP to suppress ARP flooding over VXLAN tunnels as much as possible. It helps reduce broadcast traffic by using EVPN to proxy responses to ARP requests directly to clients from the ToR VTEP. A local proxy handles ARP requests from locally attached hosts for remote hosts. ARP suppression is for IPv4; ND suppression is for IPv6. Cumulus Linux enables ARP and ND suppression by default on all VNIs to reduce ARP and ND packet flooding over VXLAN tunnels; however, you must configure L3 interfaces (SVIs) for ARP and ND suppression to work with EVPN.

Without ARP suppression, all ARP requests are broadcast throughout the entire VXLAN fabric, sent to every VTEP that has a VNI for the network. With ARP suppression enabled, MAC addresses learned over EVPN are passed down to the ARP control plane.
The leaf switch, which acts as the VTEP, responds directly back to the ARP requester through a proxy ARP reply.

Because the IP-to-MAC mappings are already communicated through the VXLAN control plane using EVPN type-2 messages, implementing ARP suppression enables optimization for faster resolution of the overlay control plane. It also reduces the amount of broadcast traffic in the fabric, as ARP suppression reduces the need for flooding ARP requests to every VTEP in the VXLAN infrastructure.

Points to consider:

- You can only use ND suppression on Spectrum_A1 and above.
- Cumulus Linux enables ARP suppression by default. However, in a VXLAN active-active (MLAG) configuration, if the switch does not suppress ARPs, the control plane does not synchronize neighbor entries between the two switches operating in active-active mode. You do not see any impact on forwarding.
- In an EVPN centralized routing configuration, where the L2 network extends beyond VTEPs, (for example, a host with bridges), the gateway MAC address does not refresh in the network when ARP suppression exists on the gateway. To work around this issue, disable ARP suppression on the centralized gateway.

EVPN for BUM Traffic

The common terminology to refer to flooded packets is Broadcast, unknown Unicast, and unknown Multicast, or BUM, packets. EVPN provides two choices for packet forwarding of BUM packets: ingress replication and L3 underlay multicast.

Ingress replication is called head-end-replication which performs unicast delivery of VXLAN encapsulated packets across remote VTEPs. In unicast replication, the source VTEP delivers the same data to every other remote VTEP. Whereas in multicast replication, all the VTEPs join at a rendezvous point (preferred is PIM-SM RP) to receive VXLAN encapsulated data. This enables multicast to have a lower overhead and faster delivery compared to unicast; however, multicast is less secure.

Ingress Replication/Head-end-replication

In ingress replication, the ingress NVE sends multiple copies of a packet, one for each egress NVE interested in the virtual network.

Benefits of this model include:

- It keeps the underlay simple. The underlay needs to provide only IP unicast routing to support network virtualization.
- It is easy to configure - there is no additional configuration required. The replication list is automatically built from the BGP EVPN RT-3 (carrying the Virtual Network Identifiers [VNIs] of interest to a VTEP) messages without any further intervention from the user.
- This makes the solution more robust, because the chances of human error are reduced significantly.

Disadvantages of this model include:

- The replication bandwidth required from the underlay can be high, especially if there are lots of BUM packets.

If the number of NVEs to replicate to is not large and the amount of BUM traffic is low, this approach works quite well. Even if the number of NVEs to replicate is higher but the amount of traffic is low, this method works quite well.
Cumulus Linux uses Head End Replication by default with EVPN multihoming.

Multicast Replication

By using multicast, the ingress NVE does not have to send a separate copy for each egress NVE. The most commonly used multicast routing protocol is called Protocol Independent Multicast (PIM). PIM-SM (PIM Sparse Mode) is used for optimizing flooded traffic in a network with EVPN-MH.

Benefits of this model include:

- It is possible to handle a large volume of BUM packets or even well-known multicast packets efficiently.

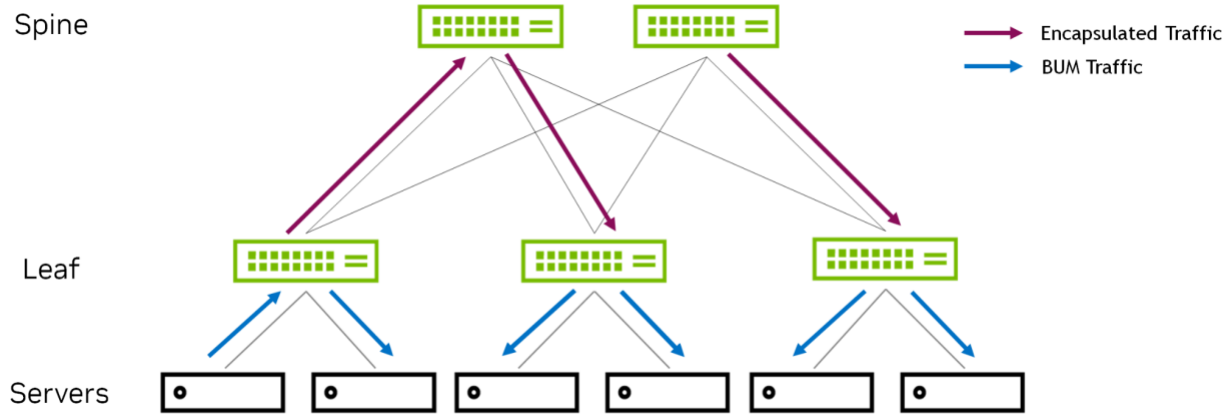
Disadvantages of this model include:

- This might become difficult to manage. In this model, in addition to providing unicast routing support, the underlay must also provide multicast routing support.
- PIM-SM requires additional protocols, such as Session Discovery Protocol (SDP), for a reliable deployment.
- To ensure that for every virtual network only the associated NVEs get the packet, each virtual network must be in its own multicast group. But this results in far too many multicast groups to scale well. So, you must now manually map all of the virtual networks into a smaller number of multicast groups. This in turn leads to some NVEs receiving BUM packets for virtual networks in which they have no interest.

- Mapping a virtual network to a multicast group also adds significant configuration complexity. You must configure the mapping of a virtual network to a multicast group on every single NVE. There is no simple way to ensure that this configuration is consistent and correct across all the NVEs.

Figure 12 shows an EVPN-PIM configuration, where underlay multicast distributes BUM traffic.

FIGURE 12 – EVPN-PIM FOR BUM TRAFFIC



By default, the VTEP floods all BUM packets (such as ARP, NS, or DHCP) it receives to all interfaces (except for the incoming interface) and to all VXLAN tunnel interfaces in the same broadcast domain. When the switch receives such packets on a VXLAN tunnel interface, it floods the packets to all interfaces in the packet's broadcast domain. For PIM-SM, type-3 routes do not result in any forwarding entries. Cumulus Linux does **not** advertise type-3 routes for an L2 VNI when BUM mode for that VNI is PIM-SM.

Dropping BUM packets

BUM packets are considered by many network administrators to be a cheap way to launch a Distributed Denial-of-Service (DDoS) attack on the network. By sending packets to addresses that might never be seen by the network, less network bandwidth is available for legitimate traffic. Such packets can deluge end hosts in that virtual network and cause them to fail because the system is very busy coping with BUM packets.
Dropping BUM packets implies that after we hear from an endpoint, its MAC address is communicated to all the other nodes via BGP. Therefore, there is really no need to flood these packets.

You can disable BUM flooding over VXLAN tunnels so that EVPN does not advertise type-3 routes for each local VNI and stops taking action on received type-3 routes.
Disabling BUM flooding is useful in a deployment with a controller or orchestrator, where the switch is pre-provisioned and there is no need to flood any ARP, NS, or DHCP packets.

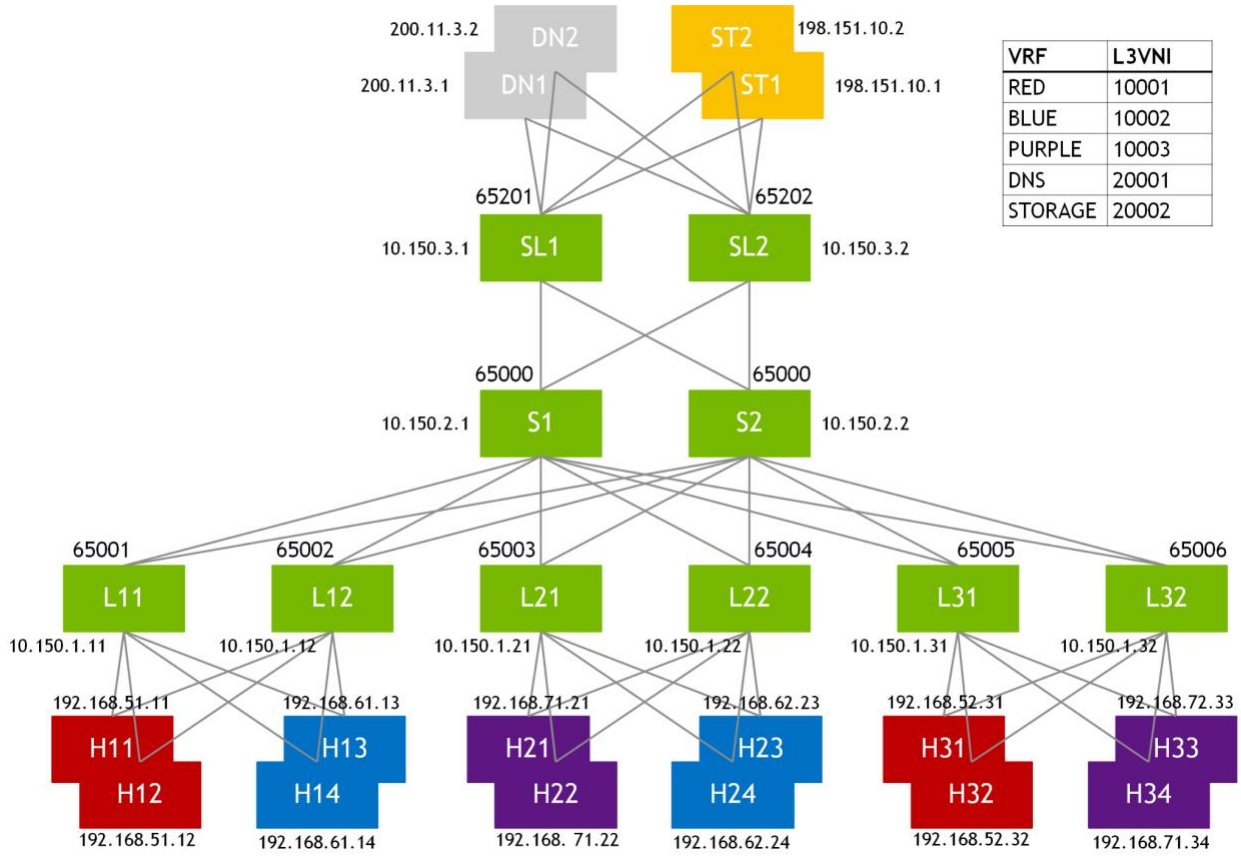
```
nv set nve vxlan flooding enable off
nv config apply
```

The main disadvantage of this approach is possible communication breakdown with traffic meant for silent servers. Silent servers are increasingly rare so this may not be a problem for your network. Dropping BUM packets is a local configuration setting, not advertised to other neighbors.

Sample Configurations

Access to common services and Internet connectivity

Consider a scenario with two tenants in different VRFs - one needs access to the Internet and the other needs access to common services like DHCP.
There is a traditional method of route leaking into the VRF that can be leveraged. However, due to its scaling implications, the Downstream VNI (D-VNI) model can be used to access common services like DHCP, DNS, and so on.
Consider the network topology below:



VRF	L3VNI
RED	10001
BLUE	10002
PURPLE	10003
DNS	20001
STORAGE	20002

The topology shows three different tenants represented by VRFs RED, BLUE and PURPLE with corresponding hosts connected over an EVPN network. The network also hosts DNS (DN1 and DN2) and Storage (ST1 and ST2) servers in separate VRFs, and these services must be offered to the 3 tenants. The service leaf switches (SL1 and SL2) only have the shared services

VRFs configured while the server leaf switches (L11, L12, L21, L22, L31 and L32) only have the tenant VRFs configured.

With D-VNI support, access to the shared services works as follows:

1. The VRFs (RED, BLUE and PURPLE) on the server leaf switches (L11 and so on) are configured to import the RTs with which the service leaf switches (SL1 and SL2) export routes from their shared services VRFs. For example, if they export with auto-derived RTs, the server leaf switches are configured to import RTs *:20001 and *:20002; the '*' refers to the ASN of SL1 and SL2 (65201 and 65202). You can specify the ASNs instead of specifying a wildcard.
2. The VRFs (DNS and STORAGE) on SL1 and SL2 are configured to import RTs announced by the server leaf switches for the tenant VRFs. For example, *:10001, *:10002 and so on.
3. When routes to the DNS servers or the storage servers are received by the server leaf switches and installed into their VRF routing tables, they are set up to use the D-VNI; 20001 and 20002 respectively.
4. Similar behavior occurs on the service leaf switches for the routes they import.
5. When a server like H11 (192.168.51.11) tries to communicate with DNS server DN1 (200.11.3.1), the corresponding leaf switch (L11 or L12) encapsulates the packet with VNI 20001 and tunnel over to SL1 (10.150.3.1) or SL2 (10.150.3.2). Based on the received VNI (20001), SL1 and SL2 know to route the packet in the DNS VRF after VXLAN decapsulation. The reverse traffic from DN1 to H11 is encapsulated by SL1 or SL2 with VNI 10001 and tunneled over to L11 or L12, where routing occurs in the RED VRF.

Below is the snippet to show the import of multiple wildcard RTs.

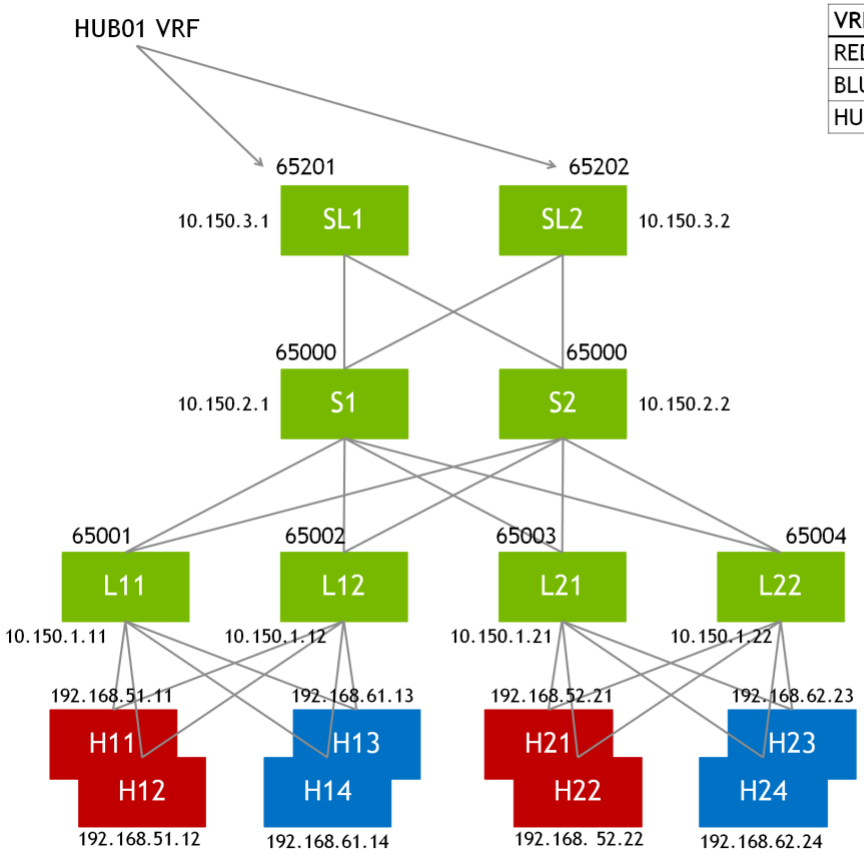
```
nv set vrf RED router bgp autonomous-system 65001
nv set vrf RED router bgp address-family ipv4-unicast redistribute
connected enable on
nv set vrf RED router bgp route-import from-evpn route-target *:20001
nv set vrf RED router bgp route-import from-evpn route-target *:20002
```

Communication between tenants in different VRFs

Consider a scenario with two tenants in different VRFs that need access to each other and access to common services like DHCP.

Route leaking can be leveraged, where for example, VRF BLUE can be imported inside VRF RED. However, the better approach is through the HUB and SPOKE method using D-VNI as explained below.

Consider the network topology below:



VRF	L3VNI
RED	10001
BLUE	10002
HUB01	20001

Communication between the tenant represented by VRF RED and the one represented by VRF BLUE is expected to happen through the service leaf switches, SL1 and SL2. This works with D-VNI as follows:

1. A VRF HUB01 is provisioned on the service leaf switches SL1 and SL2 and is configured to import routes originated in the tenant VRFs RED and BLUE by the server leaf switches. For example, VRF HUB01 is configured to import RTs *:10001 and *:10002.
2. Additionally, the service leaf switches are configured to aggregate the VRF RED and VRF BLUE routes that they import into VRF HUB01 and then to originate the aggregate for VRF RED routes with the export RT 65201:10002 (or 65202:10002); likewise, they also originate the aggregate for VRF BLUE routes with RT 65201:10001 (or 65201:10001).
3. The server leaf switches L11, L12, L21 and L22 use their auto-derived RTs for route export and import. This means that they import the aggregate routes for inter-vrf routing through the service leaf switches.
4. When a server in VRF RED like H11 (192.168.51.11) tries to communicate with a server in VRF BLUE like H24 (192.168.62.24), the corresponding leaf switch (L11 or L12) route using the aggregate route for 192.168.62.0/24 from SL1 or SL2; the packet is encapsulated with VNI 20001 and tunneled over to SL1 or SL2. SL1 or SL2 decapsulate the packet and route in VRF HUB01, which now use the host route for 192.168.62.24/32

that originated from L21 and L22; the packet is encapsulated with VNI 10002 and tunneled back over VXLAN to L21 or L22, where the packet will be routed in the BLUE VRF. Note that a similar forwarding behavior occurs even if the communication is between servers H11 and H13, which are connected to the same server leaf switches because these hosts are in different VRFs.

VRF Configuration on Border Leafs

Configuring all tenant VRFs is not required on border leafs, you can create a different VRF on the border leaf (such as shared) and import the RT of *shared* on the tenant VRFs.

Internet route distribution into the fabric

You can use route maps to distribute any route into the EVPN fabric or you can enable the default originate option on the border leaf.
The following commands set up and add a route map filter to IPv4 EVPN type-5 route advertisement:

```
nv set router policy prefix-list ext-routes-to-vrf1 rule 10 match
81.1.1.0/24
nv set router policy prefix-list ext-routes-to-vrf1 rule 10 action
permit

nv set router policy prefix-list ext-routes-to-vrf2 rule 10 match
81.1.2.0/24
nv set router policy prefix-list ext-routes-to-vrf2 rule 10 action
permit

nv set router policy prefix-list ext-routes-to-all-vrfs rule 10 match
120.0.0.1/32
nv set router policy prefix-list ext-routes-to-all-vrfs rule 10
action permit

nv set router policy route-map IPV4-TO-EXT rule 10 action permit
nv set router policy route-map IPV4-TO-EXT rule 10 match type ipv4
nv set router policy route-map IPV4-TO-EXT rule 10 match ip-prefix-
list IPV4-TO-EXT

nv set router policy route-map ext-routes-to-vrf rule 10 match type
ipv4
nv set router policy route-map ext-routes-to-vrf rule 10 match ip-
prefix-list ext-routes-to-vrf1
nv set router policy route-map ext-routes-to-vrf rule 10 set
extcommunity rt 65050:104001
nv set router policy route-map ext-routes-to-vrf rule 10 action
permit

nv set router policy route-map ext-routes-to-vrf rule 20 match type
```

```
ipv4
nv set router policy route-map ext-routes-to-vrf rule 20 match type
ipv4
nv set router policy route-map ext-routes-to-vrf rule 20 match ip-
prefix-list ext-routes-to-vrf2
nv set router policy route-map ext-routes-to-vrf rule 20 set
extcommunity rt 65050:104002
nv set router policy route-map ext-routes-to-vrf rule 20 action
permit

nv set router policy route-map ext-routes-to-vrf rule 30 match type
ipv4
nv set router policy route-map ext-routes-to-vrf rule 30 match ip-
prefix-list ext-routes-to-all-vrfs
nv set router policy route-map ext-routes-to-vrf rule 30 set
extcommunity rt 65050:104001
nv set router policy route-map ext-routes-to-vrf rule 30 set
extcommunity rt 65050:104002
nv set router policy route-map ext-routes-to-vrf rule 30 set
extcommunity rt 65050:104003
nv set router policy route-map ext-routes-to-vrf rule 30 action
permit

nv set vrf shared router bgp router-id 144.1.1.2
nv set vrf shared router bgp autonomous-system 65201
nv set vrf shared router bgp neighbor 144.1.1.1 remote-as external
nv set vrf shared router bgp address-family ipv4-unicast redistribute
connected enable on
nv set vrf shared router bgp address-family ipv4-unicast route-export
to-evpn route-map ext-routes-to-vrf
```

To originate a default type-5 route in EVPN:

```
nv set vrf shared router bgp router-id 144.1.1.2
nv set vrf shared router bgp autonomous-system 65201
nv set vrf shared router bgp neighbor 144.1.1.1 remote-as external
nv set vrf shared router bgp address-family ipv4-unicast redistribute
connected enable on
nv set vrf shared router bgp address-family ipv4-unicast route-export
to-evpn default-route-origination on
nv set vrf shared router bgp address-family ipv6-unicast route-export
to-evpn default-route-origination on
```

Additional Information

RDMA over Converged Ethernet (RoCE)

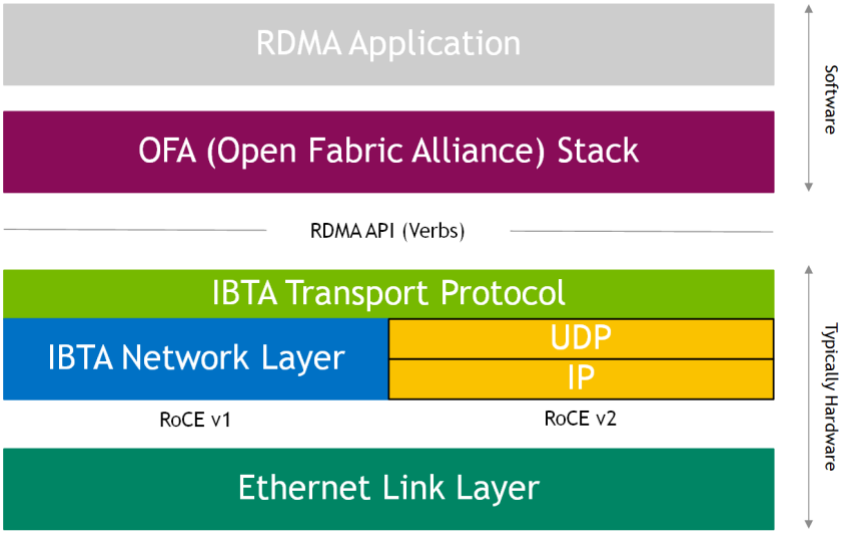
Remote Direct Memory Access (RDMA) over Converged Ethernet (RoCE) is a network protocol that leverages RDMA capabilities to accelerate communications between applications hosted on clusters of servers and storage arrays. RoCE is a remote memory management capability that allows server-to-server data movement directly between application memory without any CPU involvement. Both the transport processing and the memory translation and placement are performed by hardware resulting in lower latency, higher throughput, and better performance compared to software-based protocols.

With advances in data center convergence over reliable Ethernet, the ConnectX® Ethernet adapter cards family with its hardware offload support takes advantage of this efficient RDMA transport service over Ethernet to deliver ultra-low latency for performance-critical and transaction-intensive applications such as financial, database, storage, and content delivery networks.

As originally implemented and standardized by the InfiniBand Trade Association (IBTA), RoCE was envisioned as an L2 protocol. Effectively the IBTA L1 and L2 fields are replaced by the corresponding Ethernet fields. Specifically at L2, the local routing header (LRH) is replaced by an Ethernet MAC header and frame check sequence. The EtherType field indicates the payload encapsulates the RoCE protocol which implements the IBTA protocol above L2. In addition, the IBTA network management (subnet manager) is replaced by standard Ethernet L2 management protocols.

This approach has the advantages that it is simple to implement, strictly layered, and preserves the application-level API verbs which sit above the channel interface. The disadvantage is the scalability limitations of an L2 Ethernet deployment caused by broadcast domains and the complexity of IP allocation constraints across a flat subnet. In addition, certain switches might forward RoCE packets on a slower exception path compared to the more common IP packets. These limitations have driven the demand for RoCE to operate in L3 (routable) environments. A straightforward extension of the RoCE framework allows it to be readily transported across L3 networks. An L3 capable RoCE protocol simply continues up the stack and replaces the optional L3 global routing header (GRH) with the standard IP networking header and adds a UDP header as a stateless encapsulation of the L4 payload. This is a very natural extension of RoCE as the L3 header is already based on an IP address, therefore, this substitution is straightforward. In addition, the UDP encapsulation is a standard type of L4 packet and is forwarded efficiently by routers as a mainstream data path operation.

FIGURE 13 - RoCE



L2 Considerations

At the link level, lossless Ethernet L2 network can be achieved by using flow control. Flow control is achieved by either enabling global pause across the network, or by the use of priority flow control (PFC). PFC is a link level protocol that allows a receiver to assert flow control telling the transmitter to pause sending traffic for a specified priority. PFC supports flow control on individual priorities as specified in the class of service field of the 802.1Q VLAN tag. Therefore, it is possible for a single link to carry both lossless traffic to support RoCE and other best effort traffic on a lower priority class of service.

In a converged environment, lossy traffic shares the same physical link with lossless RoCE traffic. Typically, separate dedicated buffering and queue resources are allocated within switches and routers for the lossless and best-effort traffic classes that effectively isolate these flows from one another. Although global pause configuration is easier and might work nicely in a lab condition, PFC is recommended in an operational network to be able to differentiate between different flows. Otherwise, in case of congestion, important lossy traffic, such as control protocols, might be affected. You should run RoCE on a VLAN with priority enabled with PFC, and the control protocols (lossy) without flow control enabled on a different priority.

L3 Considerations

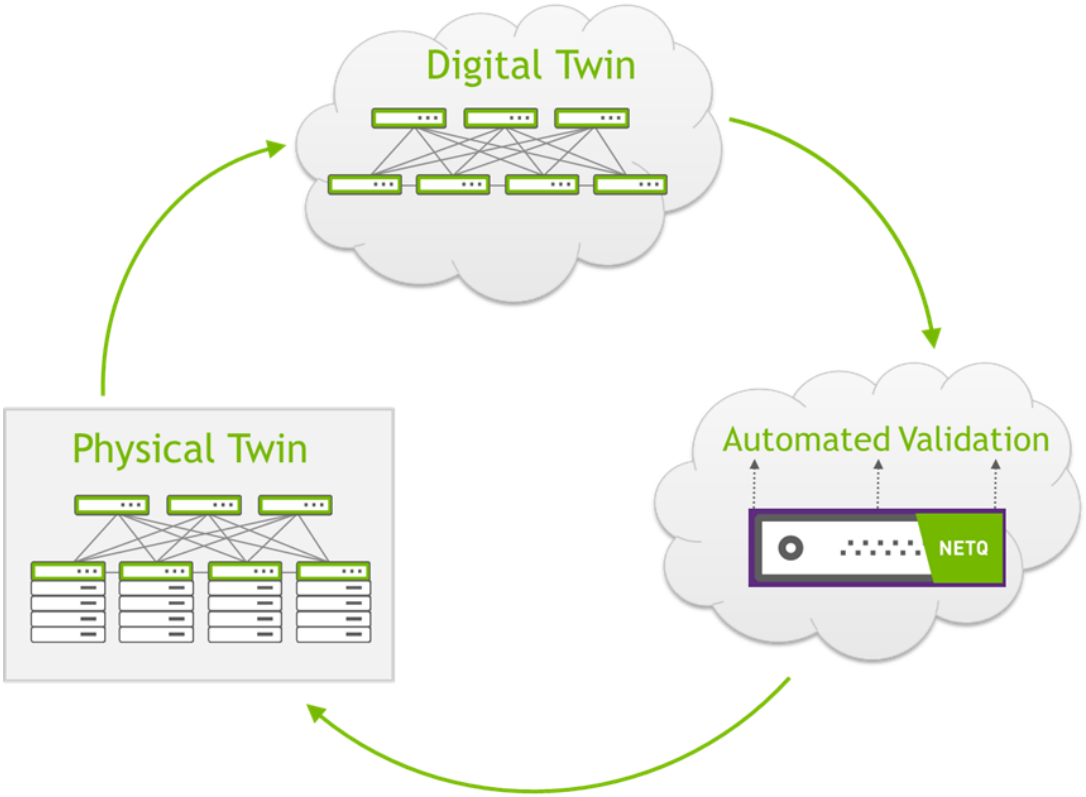
Operating RoCE at L3 requires that the lossless characteristics of the network are preserved across L3 routers that connect L2 subnets. The intervening L3 routers should be configured to transport L2 priority flow control (PFC) lossless priorities across the L3 router between Ethernet interfaces on the respective subnets. This can typically be accomplished through standard router configuration mechanisms mapping the received L2 priority settings to the corresponding L3 Differentiated Serviced Code Point (DSCP) QoS setting. The peer host should mark the RDMA packet with DSCP and L2 priority bits (802.1p or COS bits) (PCP). There are two ways for the router to extract the priority from the packet, either from the DSCP (in this case the packet could be untagged) or through PCP (in this case the packet must carry a VLAN (as the

PCP is part of the VLAN tag). The router should keep the DSCP bits unchanged, and make sure the L2 PCP bits (if VLAN exists) are copied to the next network.

Instead of being constrained by L2 link-breaking protocols such as STP, L3 networks can implement forwarding algorithms that take much better advantage of available network connectivity. Advanced data center networks can utilize multi-path routing mechanisms for load balancing and improved utilization. One commonly used protocol to achieve these goals is Equal Cost Multiple Path (ECMP). When using Reliable Connection RDMA (RC) the Source UDP port is scrambled per QP. This helps for the ECMP hash function to span different RDMA flows on different spines in a large L3 network.

Data Center Digital Twin (NVIDIA Air)

The NVIDIA approach to NetDevOps and CI/CD for networks starts with a Digital Twin of a complete data center network that can be hosted on the NVIDIA AIR platform (air.nvidia.com) or on your own private or public cloud.



This Digital Twin has logical instances of every switch and cable, as well as many servers, which enables it to be used for validating security policy compliance, automation process, monitoring tools, interoperability, and upgrade procedures. No other vendor can match this functionality. Our SE's can create a clone of your existing environment of 100+ switches in minutes.

When the physical equipment gets installed, cabled, and powered on, the entire data center network can be up and running in minutes - reducing the time to production by 95%. But now this environment is software-defined and hardware-accelerated in a way that is ready for the CI/CD operational model where the network infrastructure is treated as code because that Digital Twin doesn't go away after the production network is running.

Instead, any change requests get implemented in the Digital Twin first, where it can be verified that the security, automation, monitoring, and upgrade processes are still in compliance. Only after the functionality has been verified do the changes get pushed into the production environment. This verification process improves security and speeds up change requests, and has been shown to decrease unplanned data center downtime by 64%, according to a 2021 IDC report.

The demo marketplace on NVIDIA Air has some fully configured pre-built labs that demonstrate best-practice configuration.

- EVPN L2 Extension
- EVPN Centralized Routing
- EVPN Symmetric Routing
- EVPN Multihoming

Automation

Cumulus switches are the only ones that provide a true choice in automation - you can automate with the CLI or API, or use Linux methods (by creating flat files) with any tool of your choice, by natively treating Cumulus Linux like any other Linux server. Every other implementation uses Expect scripts or "Ansible Agents" that provide a subset of the full power of a native Ansible solution.

Production Ready Automation (PRA)

The Production Ready Automation package from NVIDIA provides several examples of a fully operationalized, automated data center and includes:

- A standard reference topology for all examples.
- A variety of golden standard EVPN-VXLAN architecture reference configurations.
- A full Vagrant and libvirt simulation of the NVIDIA reference topology (cldemo2) that provides the foundational physical infrastructure and bootstrap configuration to support and demonstrate Cumulus Linux features and technologies.
- Best practice Ansible automation and infrastructure as code (IaC).
- Working examples of Continuous Integration and Continuous Deployment (CI/CD) using GitLab.
- CI/CD testing powered by NetQ Cloud.

You can use this Production Ready Automation package as a learning resource and as a starting template to implement these features, technologies, and operational workflows in your

Cumulus Linux network environments. NVIDIA currently provides three officially supported demo solutions to overlay and provision the reference topology. These demos are EVPN-VXLAN environments and each performs tenant routing in a different style.

The golden standard demos and the underlying base reference topology are officially hosted on GitLab in the Golden Turtle folder of the Cumulus Consulting GitLab group.

- EVPN L2 Only is an EVPN-VXLAN environment with only an L2 extension.
- EVPN Centralized Routing is an EVPN-VXLAN environment with an L2 extension between tenants with inter-tenant routing on a centralized (fw) device.
- EVPN Symmetric Mode is an EVPN-VXLAN environment with an L2 extension, L3 VXLAN routing, and VRFs for multi-tenancy.

For more detailed information about IP addressing and included features, refer to the README page of the demo.

NetDevOps (CI/CD)

NVIDIA Spectrum offers the cloud-scale operational models and actionable visibility needed to enable Accelerated Ethernet. The Cumulus Linux network operating system is built on a standardized Linux stack, which integrates natively with automation and monitoring toolsets. With Cumulus Linux, you can tap into increased operational efficiency by reducing time-to-production by up to 95 percent, and spending 36 percent less IT time “keeping the lights on”. In addition, NVIDIA Air enables you to build data center digital twins and simulate upgrades, automation, and policies first to deploy with confidence. NVIDIA NetQ provides unparalleled visibility into the inner workings of the network, including the switch and DPU, providing cloud-scale insight and validation to reduce mean time to innocence and keep your network operations on track.

Learn more

To learn more about NVIDIA Cumulus Linux, visit:
nvidia.com/en-us/networking/ethernet-switching/cumulus-linux/