Configuring Layer 4 - Layer 7 Network Services Integration

This chapter contains these sections:

About VXLAN Layer 4 - Layer 7 Services

This chapter covers insertion of Layer 4 – Layer 7 network services (firewall, load balancer, and so on) in a VXLAN fabric.

As opposed to traditional 3-tier network topologies, in which L4-L7 services are connected to the switches hosting the default gateway (aggregation/distribution), L4-L7 services in VXLAN fabrics are typically connected to the leaf or border switches, often referred to as **services leafs**.

You can attach a L4-L7 services device to a VXLAN fabric in various ways. This chapter addresses the considerations you must take depending on how the L4-L7 services device is attached and the requirements of the device and the network.

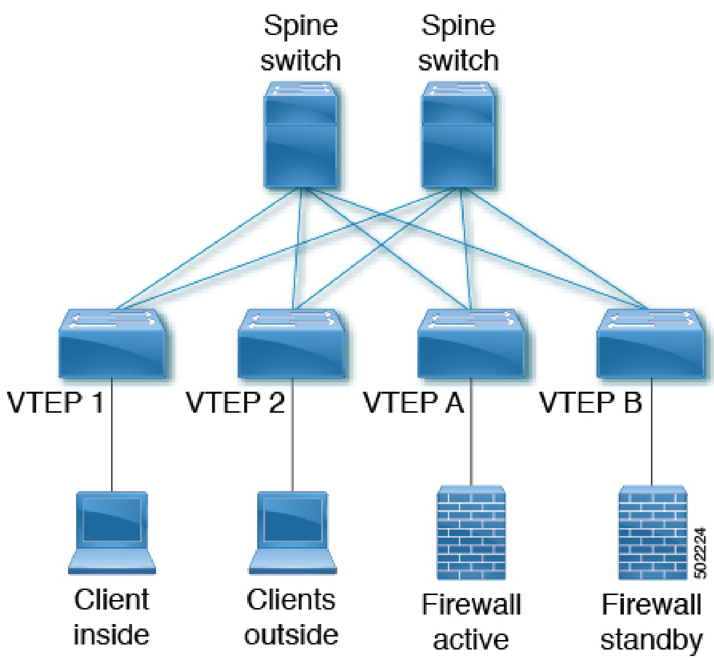
Integrating Layer 3 Firewalls in VXLAN Fabrics

This section provides details on how to integrate a firewall within a VXLAN EVPN fabric. A Layer-3 firewall involves separating different security zones.

When integrating a Layer-3 firewall in a VXLAN EVPN fabric with a distributed Anycast Gateway, each of these zones must correspond to a VRF/tenant on the fabric. The traffic within a tenant is routed by the fabric. Traffic between the tenants is routed by the firewall. This scenario often refers to an inter-tenant or tenant edge firewall.

Consider two zones: an inside zone and an outside zone. This scenario requires a VRF definition on the fabric. You can call the VRFs the inside VRF and the outside VRF. Traffic between subnets within the same VRF is routed on the VXLAN fabric using the distributed gateway. Traffic between VRFs is routed by the firewall where the rules are applied.

Figure 1: Topology Overview with Firewall Attachment



Single-Attached Firewall with Static Routing

If the firewall does not support running a routing protocol, you must have static routes on each VTEP pointing to the firewall as the next hop. The firewall also has static routes pointing to the Anycast Gateway IP as the next hop. The challenge with a static route is that the VTEP with an active firewall must be the one advertising the routes to the fabric. One way to accomplish this is to track the active firewall reachability via HMM and use this tracking to advertise routes into the fabric. When the active firewall is connected to VTEP A, VTEP A has a static route that tracks where the route is advertised if the firewall IP is learned as the HMM route. When the firewall fails and the standby firewall takes over, VTEP A learns the firewall IP using BGP, and VTEP B learns the firewall IP using HMM. VTEP A withdraws the route, and VTEP B advertises the route into the fabric. See the following example.

VTEP A and VTEP B

VTEP A and VTEP B:

```
vlan 10
 name inside
 vn-segment 10010

vlan 20
 name outside
 vn-segment 10020


interface vlan 10
 description inside_vlan
 vrf member INSIDE
 ip address 10.1.1.254/24
 fabric forwarding mode anycast-gateway
```

```
interface vlan 20
 description outside_vlan
 vrf member OUTSIDE
 ip address 20.1.1.254/24
 fabric forwarding mode anycast-gateway

interface nve1
 no shutdown
 host-reachability protocol bgp
 source-interface loopback1
 member vni 10010
  mcastgroup 239.1.1.1
 member vni 10020
  mcastgroup 239.1.1.1
 member vni 1001000 associate-vrf
 member vni 1002000 associate-vrf


track 10 ip route 10.1.1.1/32 reachability hmm
  vrf member INSIDE
!
vrf context INSIDE
 vni 1001000
 ip route 20.1.1.0/24 10.1.1.1 track 10


track 20 ip route 20.1.1.1/32 reachability hmm
  vrf member OUTSIDE

!
vrf context OUTSIDE
 vni 1001000
 ip route 10.1.1.0/24 20.1.1.1 track 20


vtepa# show track 10 track 10
ip route 20.1.1.1/32 reachability reachability is up

vtepa# show ip route 20.1.1.0/24 vrf inside
ip route table for vrf "INSIDE"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes vrf <string>

20.1.1.0/24, ubest/mbest: 1/0
  *via 10.1.1.1 [1/0], 00:00:08, static

Firewall failure on VTEP A caused the track to go down causing VTEP A to withdraw the static
 route.

vtepa# show track 20 track 20
ip route 20.1.1.1/32 reachability reachability is down

vtepa# show ip route 20.1.1.0/24 vrf INSIDE
ip route table for vrf "RED"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

route not found
```

Recursive Static Routes Distributed to the Rest of the Fabric

With this approach, the static routes are configured wherever the inside or outside VRF exists. As the next-hop is reachable through host routes (EVPN Route-Type2), the change of the active firewall to standby and vice versa is only seen locally and doesn't introduce any churn to the broader VXLAN fabric. This approach can help to better scale and improve convergence.

Any VTEP:

```
vrf context OUTSIDE
 vni 1002000
 ip route 10.1.1.0/24 20.1.1.1
! static route on VTEP pointing to firewall next hop
! firewall VIP 20.1.1.1

vrf context INSIDE
 vni 1001000
 ip route 20.1.1.0/24 10.1.1.1
! static route on VTEP pointing to firewall next hop
! firewall VIP 10.1.1.1
```

Redistribute Static Routes into BGP and Advertise to the Rest of the Fabric

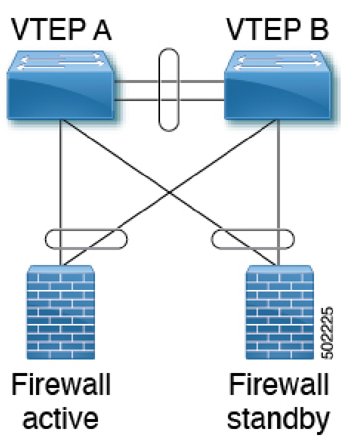
Through redistribution, we make the route toward the active firewall shown to the VTEP where it resides. The route is seen as a prefix route (EVPN Route-Type5), and as such, only the route toward the VTEP with the active firewall is seen. In the case of a firewall active/standby change, the tracking needs to detect the change and inform all of the remote VTEPs of this change. This behavior is equal to a route "delete" followed by an "add." This approach needs to notify all VTEPs with the VRF, and hence a wider churn can be seen.

VTEP A and VTEP B:

```
router bgp 65000
 vrf OUTSIDE
  address-family ipv4 unicast
   redistribute static route-map static-to-bgp
```

Dual-Attached Firewall with Static Routing

Figure 2: Dual-Attached Firewall with Static Routing



VTEP A and VTEP B:

```
vlan 10
 name inside
 vn-segment 10010

vlan 20
 name outside
 vn-segment 10020

interface nve1
 no shutdown
 host-reachability protocol bgp
 source-interface loopback1
 member vni 10010
  mcastgroup 239.1.1.1
 member vni 10020
  mcastgroup 239.1.1.1
 member vni 1001000 associate-vrf
 member vni 1002000 associate-vrf


interface vlan 10
 description inside_vlan
 vrf member INSIDE
 ip address 10.1.1.254/24
 fabric forwarding mode anycast-gateway

interface vlan 20
 description outside_vlan
 vrf member OUTSIDE
 ip address 20.1.1.254/24
 fabric forwarding mode anycast-gateway

vrf context INSIDE
 vni 1001000
 ip route 20.1.1.0/24 10.1.1.1
! static route on VTEP pointing to firewall next hop
! firewall VIP 10.1.1.1
vrf context OUTSIDE
 vni 1002000
 ip route 10.1.1.0/24 20.1.1.1
! static route on VTEP pointing to firewall next hop
! firewall VIP 20.1.1.1


router bgp 65000
 vrf INSIDE
  address-family ipv4 unicast
   redistribute static route-map inside-to-bgp
 vrf outside
  address-family ipv4 unicast
   redistribute static route-map outside-to-bgp
```

Single-Attached Firewall with eBGP Routing

If the firewall supports BGP, one option is to use BGP as a protocol between the firewall and the service VTEP. Peering using the anycast IP is not supported. The recommended design is to use dedicated loopback IPs on each VTEP and peer using the loopback. As long as the loopback interfaces are not advertised via EVPN, the same IP address could be used on all of the belonging VTEPs. We recommend using individual IP addresses on a per-VTEP basis.

Reachability to the loopback from the firewall can be configured using a static route on the firewall, pointing to the Anycast Gateway IP on the VTEPs.

In the following example, an eBGP peering is established from the VTEPs, which are in AS 65000, and the firewall in AS 65002. The BGP peering with iBGP is not supported.

Note When having eBGP peering to active/standby firewalls connected to different VTEPs, **export-gateway-ip** must be enabled. For more information, see Configuring Proportional Multipath for VNF.

Do not use Anycast Gateway for BGP peerings.

VTEP A:

```
vlan 10
 name inside
 vn-segment 10010

vlan 20
 name outside
 vn-segment 10020

interface vlan 10
 description inside_vlan
 vrf member INSIDE
ip address 10.1.1.254/24
fabric forwarding mode anycast-gateway

interface loopback100
 vrf member INSIDE
 ip address 172.16.1.253/32

 interface vlan 20
 description outside_vlan
 vrf member OUTSIDE
 ip address 20.1.1.254/24
 fabric forwarding mode anycast-gateway

 interface loopback101
 vrf member OUTSIDE
 ip address 172.18.1.253/32

router bgp 65000
 vrf INSIDE
 ! peer with firewall inside
 neighbor 10.1.1.0/24 remote-as 65123
 update-source loopback100
 ebgp-multihop 5
 address-family ipv4 unicast
  local-as 65051 no-prepend replace-as

 vrf OUTSIDE
 ! peer with firewall outside
 neighbor 20.1.1.0/24 remote-as 65123
 update-source loopback101
 ebgp-multihop 5
 address-family ipv4 unicast
  local-as 65052 no-prepend replace-as
```

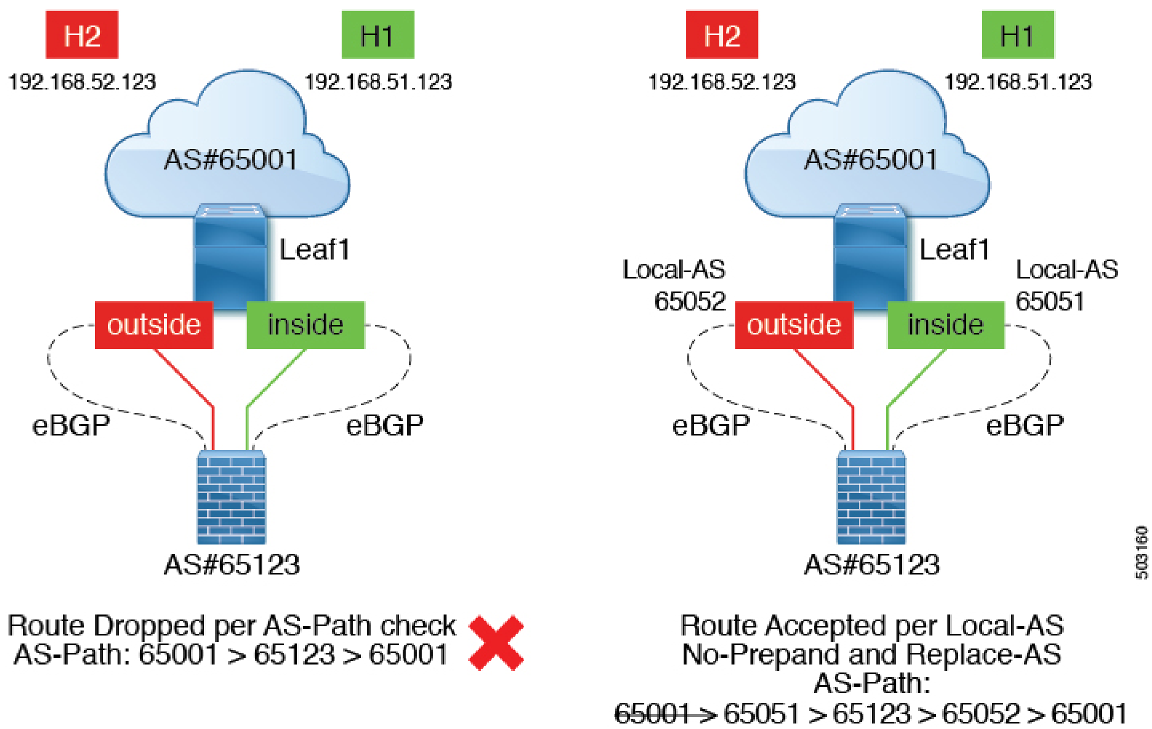
VTEP B:

Configuring Layer 4 - Layer 7 Network Services Integration

```
vlan 10
 name inside
 vn-segment 10010

vlan 20
 name outside
 vn-segment 10020
interface vlan 10
 description inside_vlan
 vrf member inside
ip address 10.1.1.254/24
fabric forwarding mode anycast-gateway

interface loopback100
 vrf member INSIDE
 ip address 172.16.1.254/32

 interface vlan 20
 description outside_vlan
 vrf member OUTSIDE
 ip address 20.1.1.254/24
 fabric forwarding mode anycast-gateway

 interface loopback101
 vrf member OUTSIDE
 ip address 172.18.1.254/32


router bgp 65000
 vrf INSIDE
 ! peer with firewall inside
 neighbor 10.1.1.0/24 remote-as 65123
 update-source loopback100
 ebgp-multihop 5
 address-family ipv4 unicast
  local-as 65051 no-prepend replace-as

 vrf OUTSIDE
 ! peer with firewall outside
 neighbor 20.1.1.0/24 remote-as 65123
 update-source loopback101
 ebgp-multihop 5
 address-family ipv4 unicast
  local-as 65052 no-prepend replace-as
```

With the VXLAN fabric generally being in a single BGP Autonomous System (AS), the AS of the inside VRF and the outside VRF is the same. BGP does not install routes that are received from its own AS. Therefore, we need to adjust the AS-path to override this rule. Various approaches exist, including disabling the rule that BGP drops routes from its own AS, which has further implications to the network. To keep all of the BGP protection mechanics in place, the "local-as" approach allows you to mimic routes being originated from a different AS. We recommend inserting the "local-as #ASN# no-prepend replace-as" on each firewall peering with different "local-as" per VRF.

Figure 3: eBGP AS-Path Check



Dual-Attached Firewall with eBGP Routing

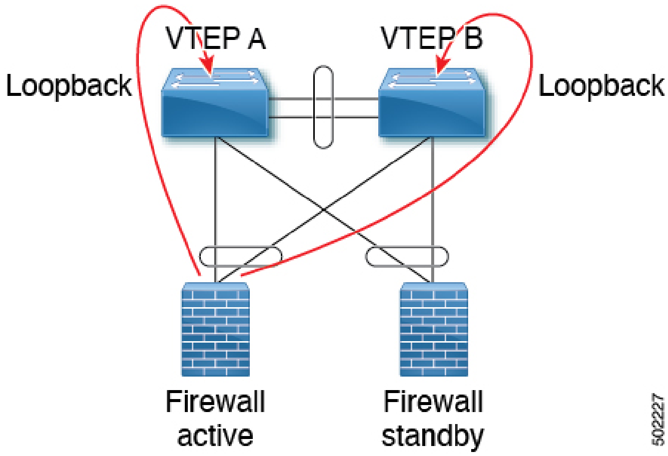
If the firewall supports BGP, one option is to use BGP as a protocol between the firewall and the service VTEP. Peering using the anycast IP is not supported. The recommended design is to use dedicated loopback IPs on each VTEP and peer using the loopback. As long as the loopback interfaces are not advertised via EVPN, the same IP address could be used on all of the belonging VTEPs. We recommend using individual IP addresses on a per-VTEP basis. For vPC environments, it is required.

Reachability to the loopback from the firewall can be configured using a static route on the firewall, pointing to the Anycast Gateway IP on the VTEPs.

In vPC deployments, you must have a per-VRF peering via a vPC peer-link. In addition to the per-VRF peering, you can enable the advertisement of prefix routes (EVPN Route-Type 5) using the **advertise-pip** command. For vPC with fabric peering, the per-VRF peering is not necessary, and the advertisement of prefix routes (EVPN Route-Type5) is required.

In the following example, an eBGP peering is established from the VTEPs, which are in AS 65000, and the firewall in AS 65002. The BGP peering with iBGP is not supported.

Figure 4: Dual-Attached Firewall with eBGP



Note When having eBGP peering to active/standby firewalls connected to different VTEPs, **export-gateway-ip** must be enabled. For more information, see Configuring Proportional Multipath for VNF.

Do not use Anycast Gateway for BGP peerings.

VTEP A:

```
vlan 10
 name inside
 vn-segment 10010

vlan 20
 name outside
 vn-segment 10020

interface vlan 10
 description inside_vlan
 vrf member INSIDE
ip address 10.1.1.254/24
fabric forwarding mode anycast-gateway

interface loopback100
 vrf member INSIDE
 ip address 172.16.1.253/32

 interface vlan 20
 description outside_vlan
 vrf member OUTSIDE
 ip address 20.1.1.254/24
 fabric forwarding mode anycast-gateway

 interface loopback101
 vrf member OUTSIDE
 ip address 172.18.1.253/32

router bgp 65000
vrf INSIDE
 ! peer with firewall inside
 neighbor 10.1.1.0/24 remote-as 65123
 update-source loopback100
```

```
 ebgp-multihop 5
 address-family ipv4 unicast
  local-as 65051 no-prepend replace-as

 vrf OUTSIDE
 ! peer with firewall outside
 neighbor 20.1.1.0/24 remote-as 65123
 update-source loopback101
 ebgp-multihop 5
 address-family ipv4 unicast
  local-as 65052 no-prepend replace-as
```

VTEP B:

```
vlan 10
 name inside
 vn-segment 10010

vlan 20
 name outside
 vn-segment 10020


interface vlan 10
 description inside_vlan
 vrf member INSIDE
ip address 10.1.1.254/24
fabric forwarding mode anycast-gateway

interface loopback100
 vrf member INSIDE
 ip address 172.16.1.254/32

 interface vlan 20
 description outside_vlan
 vrf member OUTSIDE
 ip address 20.1.1.254/24
 fabric forwarding mode anycast-gateway

 interface loopback101
 vrf member OUTSIDE
 ip address 172.18.1.254/32


router bgp 65000
 vrf INSIDE
 ! peer with firewall inside
 neighbor 10.1.1.0/24 remote-as 65123
 update-source loopback100
 ebgp-multihop 5
 address-family ipv4 unicast
  local-as 65051 no-prepend replace-as

 vrf OUTSIDE
 ! peer with firewall outside
 neighbor 20.1.1.0/24 remote-as 65123
 update-source loopback101
 ebgp-multihop 5
 address-family ipv4 unicast
  local-as 65052 no-prepend replace-as
```

Per-VRF Peering via vPC Peer-Link

This reference describes the configuration and function of per-VRF peering via vPC peer-link.

VTEP A and VTEP B:

```
vlan 3966
! vlan use for peering between the vPC VTEPS

vlan 3967
! vlan use for peering between the vPC VTEPS

system nve infra-vlans 3966,3967

interface vlan 3966
vrf member INSIDE
 ip address 100.1.1.1/31

interface vlan 3967
 vrf member OUTSIDE
 ip address 100.1.2.1/31

router bgp 65000
 vrf INSIDE
 neighbor 100.1.1.0 remote-as 65000
 update-source vlan 3966
 next-hop self
 address-family ipv4 unicast

 vrf OUTSIDE
 neighbor 100.1.2.0 remote-as 65000
 update-source vlan 3967
 next-hop self
 address-family ipv4 unicast
```

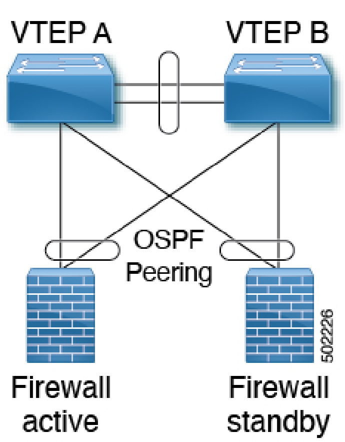
The routes learned in each VRF are advertised to the rest of the fabric via BGP EVPN updates.

Dual-Attached Firewall with OSPF

This section describes how to configure a dual-attached firewall with OSPF peering over vPC on Cisco NX-OS.

Cisco NX-OS supports dynamic OSPF peering over vPC using Layer 3, which enables firewall connectivity using vPC and establishes OSPF peering over this link. The VLAN used to establish peering between the Cisco Nexus 9000 switches and the firewall must be a non-VXLAN-enabled VLAN.

Figure 5: Dual-Attached Firewall with OSPF



Note Do not use Anycast Gateway for OSPF adjacencies.

VTEP A:

```
vlan 10
 name inside

vlan 20
 name outside

interface vlan 10
 description inside_vlan
 vrf member INSIDE
 ip address 10.1.1.253/24
 ip router ospf 1 area 0

interface vlan 20
 description outside_vlan
 vrf member OUTSIDE
 ip address 20.1.1.253/24
 ip router ospf 1 area 0

vpc domain 100
 layer3 peer-router
 peer-gateway
 peer-switch
 peer-keepalive destination x.x.x.x source x.x.x.x peer-gateway
 ipv6 nd synchronize
 ip arp synchronize

router ospf 1
 router-id 192.168.1.1
  vrf INSIDE
  vrf OUTSIDE
```

VTEP B:

```
vlan 10
 name inside

vlan 20
 name outside

interface vlan 10
 description inside_vlan
 vrf member INSIDE
 ip address 10.1.1.254/24
 ip router ospf 1 area 0

interface vlan 20
 description outside_vlan
 vrf member OUTSIDE
 ip address 20.1.1.254/24
 ip router ospf 1 area 0

vpc domain 100
 layer3 peer-router
 peer-gateway
 peer-switch
 peer-keepalive destination x.x.x.x source x.x.x.x peer-gateway
 ipv6 nd synchronize
```

```
 ip arp synchronize

router ospf 1
 router-id 192.168.2.1
  vrf INSIDE
  vrf OUTSIDE

vtepa# show ip route ospf-1 vrf outside
ip route table for vrf "OUTSIDE"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes vrf <string>

10.1.1.0/24, ubest/mbest: 1/0
  *via 20.1.1.1 vlan20, [110/41], 1w5d, ospf-1, intra

vtepa# show ip route ospf-1 vrf inside
ip route table for vrf "INSIDE"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes vrf <string>

20.1.1.0/24, ubest/mbest: 1/0
  *via 10.1.1.1 vlan10, [110/41], 1w5d, ospf-1, intra
```

Redistribute OSPF Routes into BGP and Advertise to the Rest of the Fabric

This section describes how to redistribute OSPF routes into BGP and advertise them to the rest of the fabric.

VTEP A and VTEP B:

```
router bgp 65000
 vrf OUTSIDE
  address-family ipv4 unicast
   redistribute ospf 1 route-map outsideospf-to-bgp
 vrf INSIDE
  address-family ipv4 unicast
   redistribute ospf 1 route-map insideospf-to-bgp
```

Firewall as Default Gateway

This section describes the deployment model where the firewall acts as the default gateway in a Layer 2 VXLAN fabric.

In this deployment model, the VXLAN fabric is a Layer 2 fabric, and the default gateway resides on the firewall.

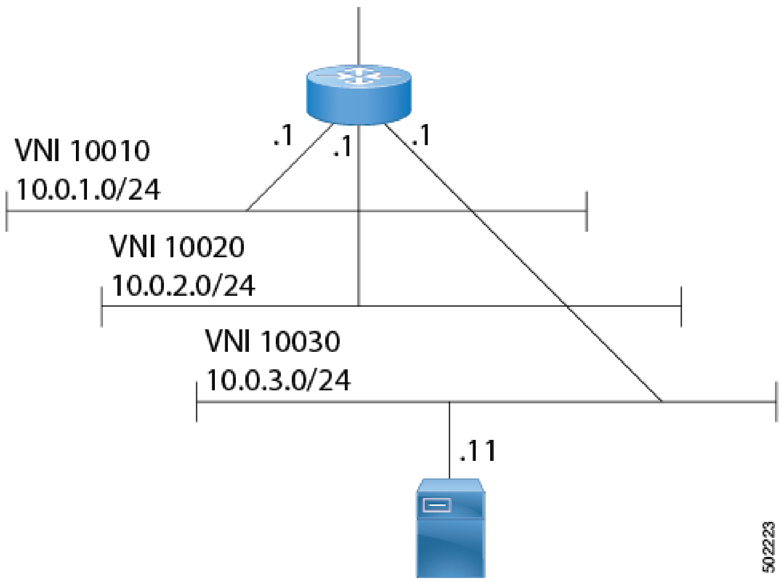
For example:

```
vlan 10
  name web
  vn-segment 10010
vlan 20
  name application
  vn-segment 10020
vlan 30
  name database
  vn-segment 10030
```

```
interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback1
  member vni 10010
   mcastgroup 239.1.1.1
  member vni 10020
   mcastgroup 239.1.1.1
  member vni 10030
   mcastgroup 239.1.1.1
```

The firewall has a logical interface in each VNI and is the default gateway for all endpoints. Every inter-VNI communication flows through the firewall. Take special care with the sizing of the firewall so that it does not become a bottleneck. Therefore, use this design in environments with low-bandwidth requirements.

Figure 6: Firewall as Default Gateway with a Layer-2 VXLAN Fabric



Transparent Firewall Insertion

Transparent firewalls or Layer 2 firewalls (including IPS/IDS) typically bridge between an inside VLAN and outside VLAN and inspect traffic as it traverses through them. VLAN stitching is done by placing the default gateway for the service on the inside VLAN. The Layer 2 reachability to this gateway is done on the outside VLAN.

Overview of EVPN with Transparent Firewall Insertion

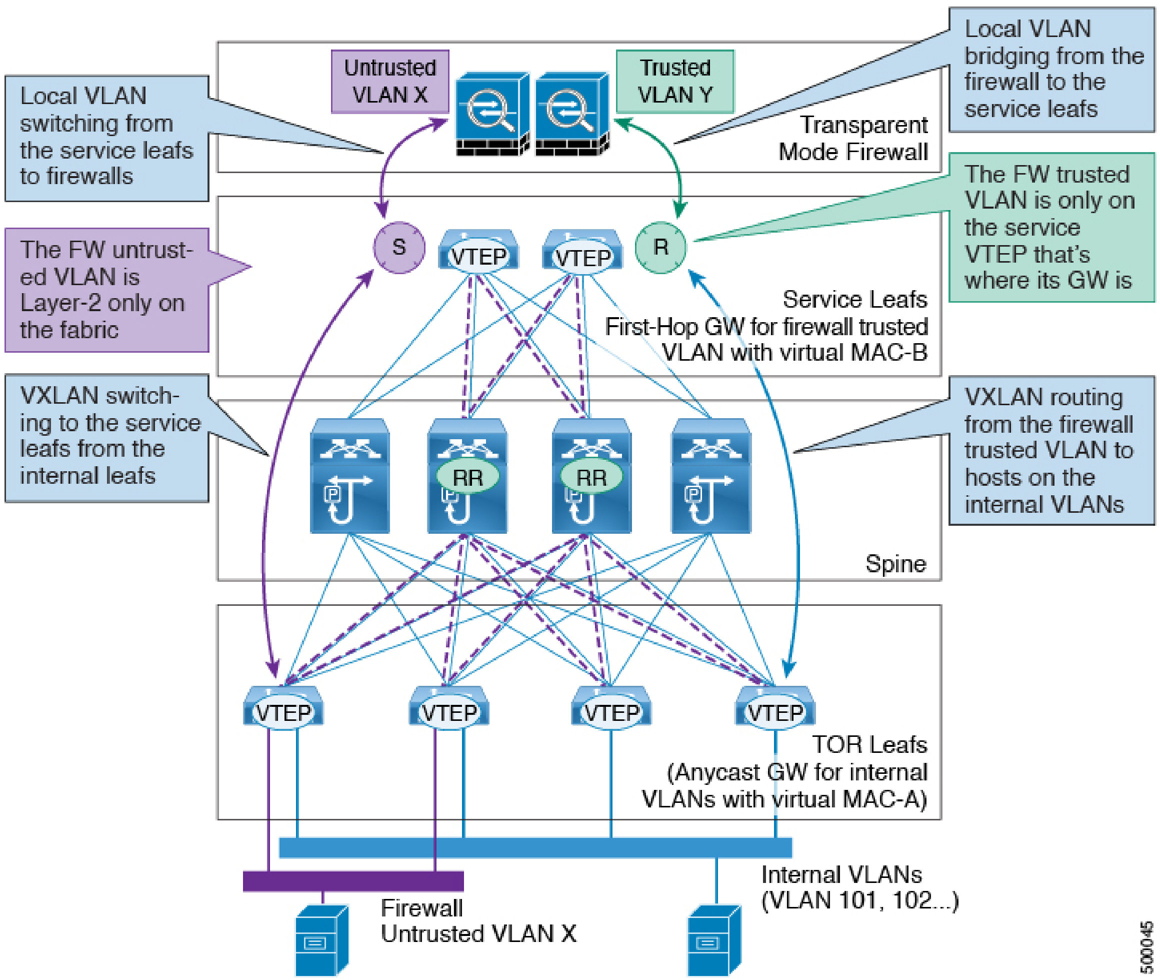
The topology contains the following types of VLANs:

- Internal VLAN (a regular VXLAN on ToR leafs with Anycast Gateway)
- Firewall untrusted VLAN X
- Firewall trusted VLAN Y

Overview of EVPN with Transparent Firewall Insertion

In this topology, the traffic that goes from VLAN X to other VLANs must go through a transparent Layer 2 firewall that is attached to the service leafs. This topology utilizes an approach of an untrusted VLAN X and a trusted VLAN Y. All ToR leafs have a Layer 2 VNI VLAN X. There is no SVI for VLAN X. The service leafs that are connected to the firewall have Layer 2 VNI VLAN X, non-VXLAN VLAN Y, and SVI Y with an HSRP gateway.

Note For VXLAN EVPN, we recommend using the distributed Anycast Gateway with transparent firewall insertion. Doing so allows all VLANs to be VXLAN enabled. When using an HSRP/VRRP-based First-Hop Gateway, the VLAN for the SVI can't be VXLAN enabled and should reside on a vPC pair for redundancy.



EVPN with Transparent Firewall Insertion Example

This section provides an example configuration for EVPN with transparent firewall insertion, including device roles and sample configurations.

This example includes the following devices and roles:

- Host in VLAN X: 10.1.94.101
- ToR leaf: N9372-1
- Service leaf in vPC: N9332-1 and N9332-2
- Border leaf: N9332-5

The following configuration examples illustrate the setup for each device in the topology.

- **ToR Leaf Configuration**

```
vlan 94
vn-segment 100094

interface nve1
  member vni 100094
   mcastgroup 239.1.1.1

router bgp 64500
  routerid 1.1.2.1
  neighbor 1.1.1.1 remote-as 64500
  address-family l2vpn evpn
    send-community extended
  neighbor 1.1.1.2 remote-as 64500
  address-family l2vpn evpn
    send-community extended
  vrf ten1
   address-family ipv4 unicast
     advertise l2vpn evpn

evpn
vni 100094 l2
  rd auto
  route-target import auto
  route-target export auto
```

- **Service Leaf 1 Configuration Using HSRP**

```
vlan 94
description untrusted_vlan
  vn-segment 100094

vlan 95
  description trusted_vlan

vpc domain 10
  peer-switch
  peer-keepalive destination 10.1.59.160
  peer-gateway
  auto-recovery
  ip arp synchronize

interface vlan2
description vpc_backup_svi_for_overlay
  no shutdown
  no ip redirects
  ip address 10.10.60.17/30
  no ipv6 redirects
  ip router ospf 100 area 0.0.0.0
  ip ospf bfd
  ip pim sparsemode

interface vlan95
```

Configuring Layer 4 - Layer 7 Network Services Integration

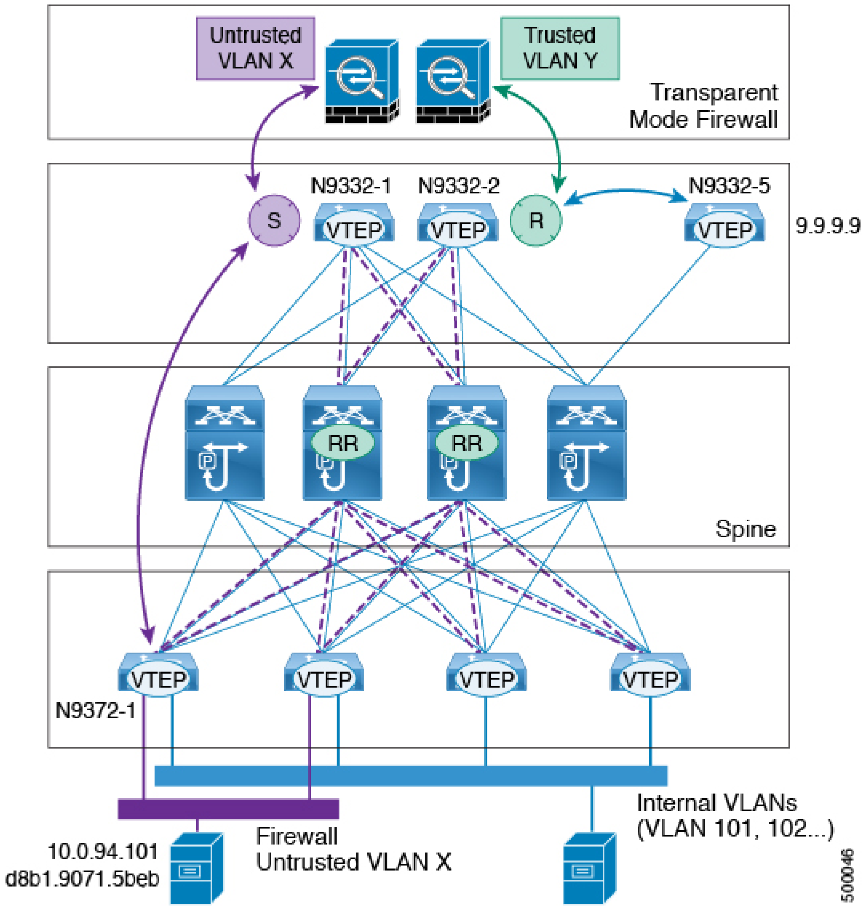
```
description svi_for_trusted_vlan
  no shutdown
  mtu 9216
  vrf member ten-1
  no ip redirects
  ip address 10.0.94.2/24
  hsrp 0
   preempt priority 255
   ip 10.0.94.1

interface nve1
  member vni 100094
   mcast-group 239.1.1.1

router bgp 64500
  routerid 1.1.2.1
  neighbor 1.1.1.1 remote-as 64500
  address-family l2vpn evpn
   send-community extended
  neighbor 1.1.1.2 remote-as 64500
   address-family l2vpn evpn
    send-community extended
  vrf ten-1
   address-family ipv4 unicast
    network 10.0.94.0/24 /*advertise /24 for svi 95 subnet; it is not vxlan anymore*/
    advertise l2vpn evpn

evpn
vni 100094 l2
 rd auto
 route-target import auto
 route-target export auto
```

- **Service Leaf 2 Configuration Using HSRP**

```
vlan 94
  description untrusted_vlan
  vnsegment 100094

vlan 95
  description trusted_vlan

vpc domain 10
  peer-switch
  peer-keepalive destination 10.1.59.159
  peer-gateway
  auto-recovery
  ip arp synchronize

interface vlan2
description vpc_backup_svi_for_overlay
  no shutdown
  no ip redirects
  ip address 10.10.60.18/30
  no ipv6 redirects
  ip router ospf 100 area 0.0.0.0
  ip pim sparsemode

interface vlan95
  description svi_for_trusted_vlan
  no shutdown
  mtu 9216
  vrf member ten-1
  no ip redirects
  ip address 10.0.94.3/24
```

```
  hsrp 0
   preempt priority 255
   ip 10.0.94.1

interface nve1
  member vni 100094
   mcastgroup 239.1.1.1

router bgp 64500
  router-id 1.1.2.1
  neighbor 1.1.1.1 remote-as 64500
  address-family l2vpn evpn
    send-community extended
  neighbor 1.1.1.2 remote-as 64500
  address-family l2vpn evpn
   send-community extended
  vrf ten-1
    address-family ipv4 unicast
     network 10.0.94.0/24 /*advertise /24 for svi 95 subnet; it is not vxlan anymore*/

     advertise l2vpn evpn

evpn
vni 100094 l2
  rd auto
  route-target import auto
  route-target export auto
```

Figure 7: EVPN with Transparent Firewall Insertion Topology



Show Command Examples

This section provides examples of show commands to display MAC address, ARP, and route information on the switch.

Display information about the ingress leaf learned local MAC from host:

```
switch# sh mac add vl 94 | i 5b|MAC
* primary entry, G - Gateway MAC, (R) Routed - MAC, O - Overlay MAC
VLAN MAC Address Type age Secure NTFY Ports
* 94 d8b1.9071.5beb dynamic 0 F F Eth1/1
```

Display information about the service leaf found MAC of host:

Note In VLAN 94, the service leaf learned the host MAC from the remote peer by BGP.

```
switch# sh mac add vl 94 | i VLAN|eb

VLAN MAC Address Type age Secure NTFY Ports
* 94 d8b1.9071.5beb dynamic 0 F F nve1(1.1.2.1)
```

```
switch#  sh mac add vl 94 | i VLAN|eb

VLAN MAC Address Type age Secure NTFY Ports
* 94 d8b1.9071.5beb dynamic 0 F F nve1(1.1.2.1)

switch#  sh mac add vl 95 | i VLAN|eb

VLAN MAC Address Type age Secure NTFY Ports
+ 95 d8b1.9071.5beb dynamic 0 F F Po300

switch#  sh mac add vl 95 | i VLAN|eb

VLAN MAC Address Type age Secure NTFY Ports
+ 95 d8b1.9071.5beb dynamic 0 F F Po300
```

Display information about service leaf learned ARP for host on VLAN 95:

```
switch# sh ip arp vrf ten-1
Address      Age       MAC Address      Interface
10.0.94.101 00:00:26 d8b1.9071.5beb  Vlan95
```

Service Leaf learns 9.9.9.9 from EVPN.

```
switch# sh ip route vrf ten-1 9.9.9.9
IP Route Table for VRF "Ten-1"
'*' denotes best ucast nexthop
'**' denotes best mcast nexthop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

9.9.9.9/32, ubest/mbest: 1/0
  *via 1.1.2.7%default, [200/0], 02:57:27, bgp64500,internal, tag 65000 (evpn) segid: 10011
tunnelid: 0x1
010207 encap: VXLAN
```

Display information about the border leaf learned host routes by BGP:

```
switch# sh ip route 10.0.94.101

IP Route Table for VRF "default"
'*' denotes best ucast nexthop
'**' denotes best mcast nexthop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.0.94.0/24, ubest/mbest: 1/0
  *via 10.100.5.0, [20/0], 03:14:27, bgp65000,external, tag 6450
```

Service Redirection in VXLAN EVPN Fabrics

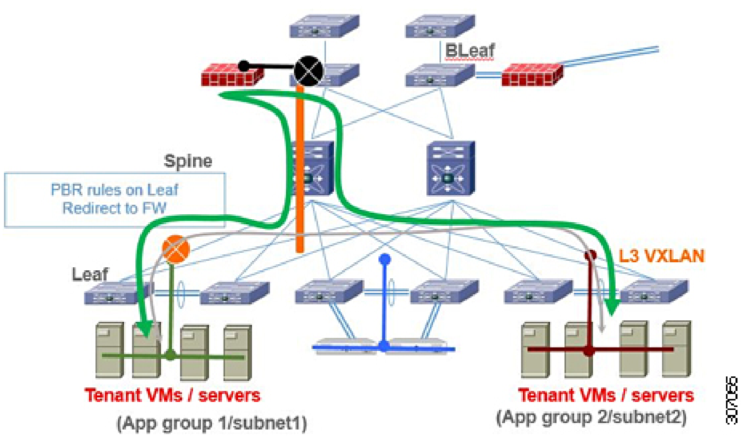
Today, insertion of service appliances (also referred to as service nodes or service endpoints) such as firewalls, load-balancers, etc are needed to secure and optimize applications within a data center. This section describes the Layer 4-Layer 7 service insertion and redirection features offered on VXLAN EVPN fabrics that provides sophisticated mechanisms to onboard and selectively redirect traffic to these services.

Use of Policy-Based Redirect for Services Insertion

Policy-based redirect (PBR) provides a mechanism to bypass a routing table lookup and redirect traffic to a next-hop IP reachable over VXLAN. The feature enables service redirection to Layer 4-Layer 7 devices such as firewalls and load balancers.

- PBR involves configuring a route-map with rules that dictate where traffic must be forwarded. The route map is applied on the tenant SVI to influence traffic coming from the host-facing interfaces to a next hop reachable via the fabric.
- In scenarios where traffic is coming to a VTEP from the overlay and needs to be redirected to another next hop, the PBR policy must be applied on the fabric facing Layer-3 VNI Interface.
- In the previous figure, communication between App group 1 and App group 2 takes place via inter-VLAN/VNI routing in the tenant VRF by default. If there is a requirement where traffic from App group 1 to App group 2 must go through a firewall, a PBR policy can be used to redirect traffic. The example in section "Configuration Example for Policy-Based Redirect" provides the necessary configuration that redirects the traffic flow.
- This VXLAN PBR functionality is very basic and lacks many of the required functionality for proper insertion of services in VXLAN fabric. Hence the recommendation is to instead look at ePBR for all the reasons explained in Enhanced-Policy Based Redirect (ePBR), on page 26 section.

Figure 8:



Guidelines and Limitations for Policy-Based Redirect

The following guidelines and limitations apply to PBR over VXLAN.

- The following platforms support PBR over VXLAN:
    - Cisco Nexus 9332C and 9364C switches
    - Cisco Nexus 9300-EX switches
    - Cisco Nexus 9300-FX/FX2/FX3 switches
    - Cisco Nexus 9300-GX switches
    - Cisco Nexus 9504 and 9508 switches with -EX/FX line cards

- PBR over VXLAN doesn't support the following features:VTEP ECMP, and the **load-share** keyword in the **set {ip | ipv6} next-hop** *ip-address* command.
- When you configure **bestpath as-path multipath-relax**, BGP installs all the multi-paths for IPv4 as best-path in URIB with least metric available among the paths.
- When you configure **bestpath as-path multipath-relax**, BGP doesn't install all the multi-paths for IPv6 as best-path in U6RIB. It will still have the individual metric available for those paths.

Enable the Policy-Based Redirect Feature

Before you begin

Enable the policy-based redirect feature before you can configure a route policy.

To configure basic PBR, in cases where the advanced (and recommended) ePBR functions are not deployed, see the following sections:

- Enable the Policy-Based Redirect Feature
- Configuring a Route Policy
- Verifying the Policy-Based Redirect Configuration
- Configuration Example for Policy-Based Redirect

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
```

Step 2 Use the **[no] feature pbr** command to enable the policy-based routing feature.

Example:

```
switch(config)# feature pbr
```

Step 3 (Optional) Use the **show feature** command to display enabled and disabled features.

Example:

```
switch(config)# show feature
```

Step 4 (Optional) Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config)# copy running-config startup-config
```

Configuring a Route Policy

Before you begin

Configure the RACL TCAM region (using TCAM carving) before you apply the policy-based routing policy. For instructions, see the “Configuring ACL TCAM Region Sizes” section in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide, Release 9.2(x).

You can use route maps in policy-based routing to assign routing policies to the inbound interface. Cisco NX-OS routes the packets when it finds a next hop and an interface.

Note The switch has a RACL TCAM region by default for IPv4 traffic.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
```

Step 2 Use the **interface** *type slot/port* command to enter interface configuration mode.

Example:

```
switch(config)# interface ethernet 1/2
```

Step 3 Use the **{ip | ipv6} policy route-map** *map-name* command to assign a route map for IPv4 or IPv6 policy-based routing to the interface.

Example:

```
switch(config-inf)# ip policy route-map Testmap
```

Step 4 Use the **route-map** *map-name* **[permit | deny]** [*seq*] command to create a route map or enter route-map configuration mode for an existing route map.

Example:

```
switch(config-inf)# route-map Testmap
```

Step 5 Use the **match {ip | ipv6} address access-list-name** *name* [*name*...] command to match an IPv4 or IPv6 address against one or more IPv4 or IPv6 access control lists (ACLs).

Example:

```
switch(config-route-map)# match ip address access-list-name ACL1
```

Step 6 Use the **set ip next-hop** *address1* command to set the IPv4 next-hop address for policy-based routing.

Example:

```
switch(config-route-map)# set ip next-hop 192.0.2.1
```

Step 7 Use the **set ipv6 next-hop** *address1* command to set the IPv6 next-hop address for policy-based routing.

Example:

```
switch(config-route-map)# set ipv6 next-hop 2001:0DB8::1
```

Step 8 Use the **set interface null0** command to set the interface that is used for routing.

Example:

```
switch(config-route-map)# set interface null0
```

Step 9 Use the **copy running-config startup-config** command to save this configuration change.

Example:

```
switch(config-route-map)# copy running-config startup-config
```

Verifying the Policy-Based Redirect Configuration

This section describes how to verify the policy-based redirect configuration.

To display the policy-based redirect configuration information, perform one of the following tasks:

- Use the **route-map** *map-name* **pbr-statistics** command to enable policy statistics.
- Use the **clear route-map** *map-name* **pbr-statistics** command to clear these policy statistics.

Table 1: Policy-Based Redirect Verification Commands

Command	Purpose
show [ip | ipv6] policy [*name*]	Displays information about an IPv4 or IPv6 policy.
show route-map [*name*] **pbr-statistics**	Displays policy statistics.

Configuration Example for Policy-Based Redirect

This section provides a configuration example for implementing Policy-Based Redirect (PBR) on tenant and service VTEPs.

Perform the following configuration on all tenant VTEPs, excluding the service VTEP.

```
feature pbr

ipv6 access-list ipv6_app_group_1
10 permit ipv6 any 2001:10:1:1::0/64

ip access-list ipv4_app_group_1
10 permit ip any 10.1.1.0/24

ipv6 access-list ipv6_app_group_2
10 permit ipv6 any 2001:20:1:1::0/64

ip access-list ipv4_app_group_2
10 permit ip any 20.1.1.0/24

route-map ipv6_pbr_appgroup1 permit 10
  match ipv6 address ipv6_app_group_2
  set ipv6 next-hop 2001:100:1:1::20  (next hop is that of the firewall)

route-map ipv4_ pbr_appgroup1 permit 10
  match ip address ipv4_app_group_2
  set ip next-hop 10.100.1.20 (next hop is that of the firewall)

route-map ipv6_pbr_appgroup2 permit 10
  match ipv6 address ipv6_app_group1
  set ipv6 next-hop 2001:100:1:1::20  (next hop is that of the firewall)
```

```
route-map ipv4_ pbr_appgroup2 permit 10
  match ip address ipv4_app_group_1
  set ip next-hop 10.100.1.20 (next hop is that of the firewall)


interface vlan10
! tenant SVI appgroup 1
vrf member appgroup
 ip address 10.1.1.1/24
 no ip redirect
 ipv6 address 2001:10:1:1::1/64
 no ipv6 redirects
 fabric forwarding mode anycast-gateway
ip policy route-map ipv4_ pbr_appgroup1
ipv6 policy route-map ipv6_pbr_appgroup1
interface vlan20
! tenant SVI appgroup 2
vrf member appgroup
 ip address 20.1.1.1/24
 no ip redirect
 ipv6 address 2001:20:1:1::1/64
 no ipv6 redirects
 fabric forwarding mode anycast-gateway
ip policy route-map ipv4_ pbr_appgroup2
ipv6 policy route-map ipv6_pbr_appgroup2

on the service vtep, the pbr policy is applied on the tenant vrf svi. this ensures the
traffic post decapsulation will be redirected to firewall.
feature pbr

ipv6 access-list ipv6_app_group_1
10 permit ipv6 any 2001:10:1:1::0/64

ip access-list ipv4_app_group_1
10 permit ip any 10.1.1.0/24

ipv6 access-list ipv6_app_group_2
10 permit ipv6 any 2001:20:1:1::0/64

ip access-list ipv4_app_group_2
10 permit ip any 20.1.1.0/24

route-map ipv6_pbr_appgroup1 permit 10
  match ipv6 address ipv6_app_group_2
  set ipv6 next-hop 2001:100:1:1::20  (next hop is that of the firewall)

route-map ipv6_pbr_appgroup permit 20
  match ipv6 address ipv6_app_group1
  set ipv6 next-hop 2001:100:1:1::20  (next hop is that of the firewall)

route-map ipv4_ pbr_appgroup permit 10
  match ip address ipv4_app_group_2
  set ip next-hop 10.100.1.20 (next hop is that of the firewall)

route-map ipv4_ pbr_appgroup permit 20
  match ip address ipv4_app_group_1
  set ip next-hop 10.100.1.20 (next hop is that of the firewall)


interface vlan1000
!L3VNI SVI for tenant VRF
vrf member appgroup
ip forward
ipv6 forward
```

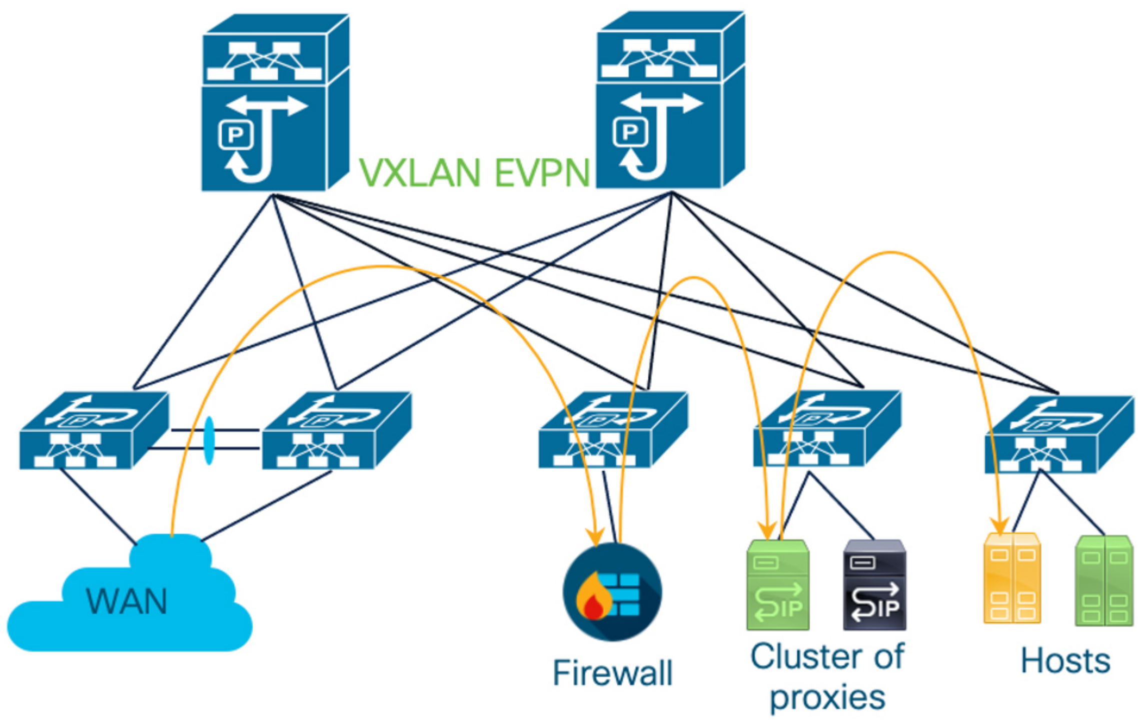
```
ipv6 ipv6 address use-link-local-only
ip policy route-map ipv4_ pbr_appgroup
ipv6 policy route-map ipv6_pbr_appgroup
```

Enhanced-Policy Based Redirect (ePBR)

VXLAN PBR as a solution to selectively redirect traffic can only cater to simple traffic redirection requirements. For more complex use cases like service chaining, symmetric load-balancing, or tracking health of service appliances, usage of PBR becomes difficult. The challenge with service chaining using PBR is that it requires the user to create unique policies per node and manage the redirection rules manually across all the nodes in the chain. Also, given the stateful nature of the service nodes, the PBR rules must ensure symmetry for the reverse traffic, and this adds additional complexity to the configuration and management of the PBR policies.

Enhanced Policy-Based Redirect (ePBR) provides a comprehensive solution to insert service nodes, selectively redirect and load-balance traffic. ePBR provides a simplified workflow to create traffic chains and load-balancing rules along with providing options for probing/monitoring the health of service appliances and taking corrective action in the event of failure. ePBR is supported in both single and multi-site VXLAN EVPN deployments.

In this Figure, selective traffic originating from WAN is chained to a firewall and then the traffic is load-balanced across a cluster of proxies before forwarding toward the destination hosts. ePBR ensures symmetry is maintained for a given flow by making sure that traffic in both forward and reverse direction is redirected to the same service endpoint in the cluster of TCP proxies.



For more detailed information, guidelines and configuration examples on ePBR, see Cisco Nexus 9000 Series NX-OS ePBR Configuration Guide and Layer 4 to Layer 7 Service Redirection with Enhanced Policy-Based Redirect White Paper.