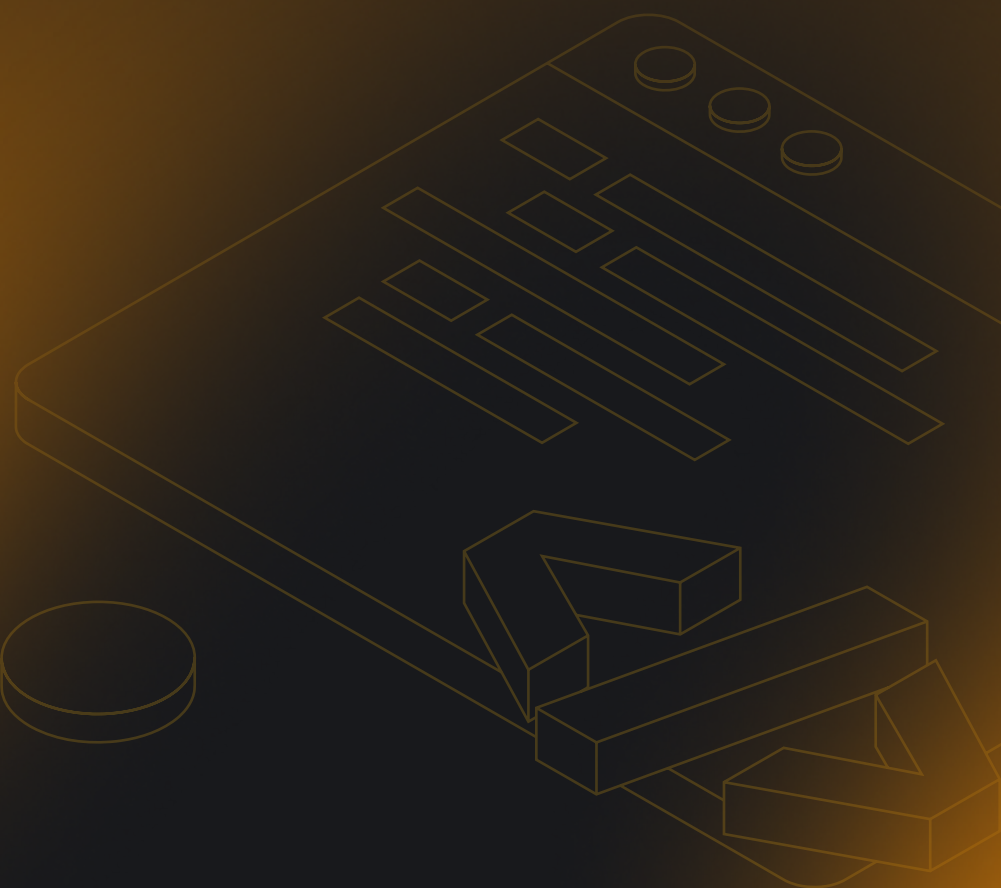




Deployment Guide | Technical Doc

VYOS ANYCAST GATEWAY IN EVPN-VXLAN NETWORK

Index:

Introduction	3
Scenario	3
Topology	4
Configuration and deployment	5
Prerequisites	5
Underlay configuration	6
Overlay configuration	8
Anycast gateway IP and MAC configuration	10
Validation and troubleshooting	11
Underlay	11
Overlay	12
Test anycast gateway and symmetric IRB	14
Conclusion	18



Introduction

In an EVPN-VXLAN network, the anycast gateway feature enables multiple VTEPs (VXLAN Tunnel Endpoints) to advertise and provide the same default gateway IP address for hosts within a Layer 2 segment. This architecture allows workloads and services to be migrated seamlessly across different locations without requiring gateway reconfiguration.

By sharing both the gateway IP address and MAC address across all participating VTEPs, hosts can continue using the same default gateway regardless of their physical location. The consistent gateway MAC address also eliminates the need for ARP relearning during workload migration, reducing traffic disruption and improving convergence times.

To implement the anycast gateway, VyOS uses a pseudo-Ethernet interface that hosts the shared gateway IP and MAC addresses. This approach prevents Duplicate Address Detection (DAD) issues that arise when the same gateway address is configured on multiple devices and ensures consistent gateway behavior across the EVPN fabric.

Scenario

The following scenario illustrates a typical EVPN-VXLAN deployment in a Leaf–Spine topology. In this example, PC1 needs to communicate with PC6. Because the two hosts belong to different Layer 2 segments, VLAN 100 for PC1 and VLAN 200 for PC6, the traffic must be routed through the default gateway. Therefore, PC1 first needs to learn the MAC address associated with its gateway IP address (192.168.1.1).

One of the key advantages of EVPN is seamless host mobility. If PC1 moves from a location connected to LEAF-1 to a location connected to LEAF-2, it can retain the same IP address and default gateway configuration. Additionally, PC1 does not need to relearn a new default gateway MAC address, as the gateway is presented consistently across the fabric.

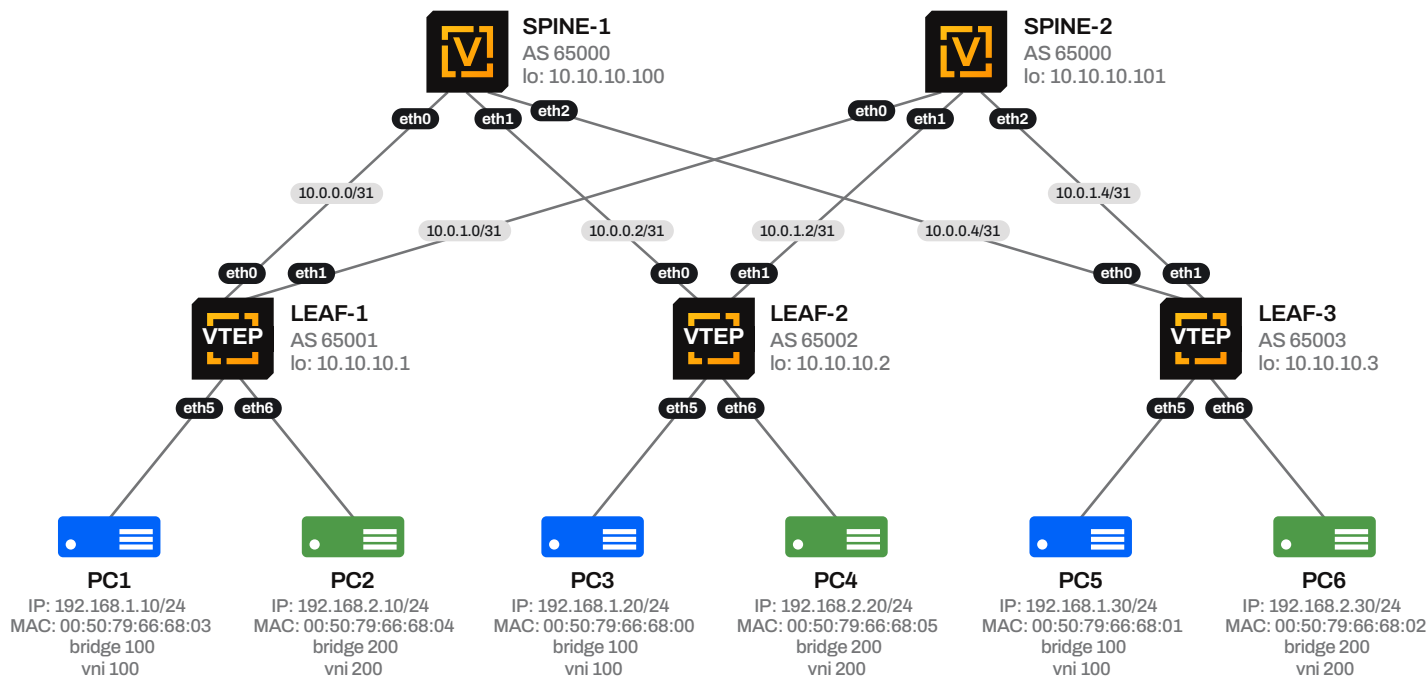
In this topology, PC1 and PC3 are connected to LEAF-1 and LEAF-2, respectively, while remaining in the same VLAN. They are used to represent the movement of PC1 between leaf switches. PC6 is connected to LEAF-3 and resides in VLAN 200.



Topology

The following diagram illustrates the topology used in this scenario.

IPv4 fabric eBGP underlay, EVPN VXLAN overlay, Anycast Gateway



Configuration and deployment

Prerequisites

This anycast gateway guide was developed using VyOS 2026.06.24-0045-rolling and has been tested and verified.

Networks	Addresses
LEAF-SPINE	10.0.0.0/16
Loopback	10.10.10.0/24
VLAN 100	192.168.1.0/24
VLAN 200	192.168.2.0/24

IP addresses of PCs:

PCs	Addresses
PC1	192.168.1.10/24
PC2	192.168.2.10/24
PC3	192.168.1.20/24
PC4	192.168.2.20/24
PC5	192.168.1.30/24
PC6	192.168.2.30/24

Loopback addressing:

Loopbacks	Addresses
LEAF-1	10.10.10.1
LEAF-2	10.10.10.2
LEAF-3	10.10.10.3
SPINE-1	10.10.10.100
SPINE-2	10.10.10.101



Underlay configuration

In this scenario, eBGP is used as the underlay routing protocol, providing ECMP support for efficient load balancing and path redundancy. All SPINE switches belong to the same AS, while each LEAF switch operates within its own dedicated AS.

1. Configure interfaces in the EVPN fabric:

LEAF-1

```
set interfaces ethernet eth0 address '10.0.0.1/31'
set interfaces ethernet eth0 description 'to SPINE-1'
set interfaces ethernet eth0 mtu '9200'
set interfaces ethernet eth1 address '10.0.1.1/31'
set interfaces ethernet eth1 description 'to SPINE-2'
set interfaces ethernet eth1 mtu '9200'
```

LEAF-2

```
set interfaces ethernet eth0 address '10.0.0.3/31'
set interfaces ethernet eth0 description 'to SPINE-1'
set interfaces ethernet eth0 mtu '9200'
set interfaces ethernet eth1 address '10.0.1.3/31'
set interfaces ethernet eth1 description 'to SPINE-2'
set interfaces ethernet eth1 mtu '9200'
```

LEAF-3

```
set interfaces ethernet eth0 address '10.0.0.5/31'
set interfaces ethernet eth0 description 'to SPINE-1'
set interfaces ethernet eth0 mtu '9200'
set interfaces ethernet eth1 address '10.0.1.5/31'
set interfaces ethernet eth1 description 'to SPINE-2'
set interfaces ethernet eth1 mtu '9200'
```

SPINE-1

```
set interfaces ethernet eth0 address '10.0.0.0/31'
set interfaces ethernet eth0 description 'toLEAF-1'
set interfaces ethernet eth0 mtu '9200'
set interfaces ethernet eth1 address '10.0.0.2/31'
set interfaces ethernet eth1 description 'toLEAF-2'
set interfaces ethernet eth1 mtu '9200'
set interfaces ethernet eth2 address '10.0.0.4/31'
set interfaces ethernet eth2 description 'toLEAF-3'
set interfaces ethernet eth2 mtu '9200'
```

SPINE-2

```
set interfaces ethernet eth0 address '10.0.1.0/31'
set interfaces ethernet eth0 description 'toLEAF-1'
```



```
set interfaces ethernet eth0 mtu '9200'  
set interfaces ethernet eth1 address '10.0.1.2/31'  
set interfaces ethernet eth1 description 'toLEAF-2'  
set interfaces ethernet eth1 mtu '9200'  
set interfaces ethernet eth2 address '10.0.1.4/31'  
set interfaces ethernet eth2 description 'toLEAF-3'  
set interfaces ethernet eth2 mtu '9200'
```

2. Configure Loopback interfaces:

LEAF-1

```
set interfaces loopback lo address '10.10.10.1/32'
```

LEAF-2

```
set interfaces loopback lo address '10.10.10.2/32'
```

LEAF-3

```
set interfaces loopback lo address '10.10.10.3/32'
```

SPINE-1

```
set interfaces loopback lo address '10.10.10.100/32'
```

SPINE-2

```
set interfaces loopback lo address '10.10.10.101/32'
```

3. Configuring the eBGP protocol for underlay connectivity:

Shared LEAF configuration

```
set policy prefix-list LOOPBACKS rule 5 action 'permit'  
set policy prefix-list LOOPBACKS rule 5 ge '32'  
set policy prefix-list LOOPBACKS rule 5 prefix '10.10.10.0/24'  
set policy route-map UNDERLAY-CONNECTED description 'Redistribute connected routes'  
set policy route-map UNDERLAY-CONNECTED rule 5 action 'permit'  
set policy route-map UNDERLAY-CONNECTED rule 5 match ip address prefix-list 'LOOPBACKS'  
set protocols bgp address-family ipv4-unicast maximum-paths ebgp '2'  
set protocols bgp address-family ipv4-unicast redistribute connected route-map 'UNDERLAY-CONNECTED'  
set protocols bgp peer-group SPINES-UNDERLAY address-family ipv4-unicast  
set protocols bgp peer-group SPINES-UNDERLAY remote-as '65000'  
set protocols bgp system-as '65001'
```

LEAF-1

```
set protocols bgp neighbor 10.0.0.0 peer-group 'SPINES-UNDERLAY'  
set protocols bgp neighbor 10.0.1.0 peer-group 'SPINES-UNDERLAY'
```

LEAF-2

```
set protocols bgp neighbor 10.0.0.2 peer-group 'SPINES-UNDERLAY'  
set protocols bgp neighbor 10.0.1.2 peer-group 'SPINES-UNDERLAY'
```

LEAF-3

```
set protocols bgp neighbor 10.0.0.4 peer-group 'SPINES-UNDERLAY'  
set protocols bgp neighbor 10.0.1.4 peer-group 'SPINES-UNDERLAY'
```

Shared SPINE configuration

```
set protocols bgp address-family ipv4-unicast maximum-paths ebgp '128'  
set protocols bgp address-family ipv4-unicast redistribute connected route-map 'UNDERLAY-CONNECTED'  
set protocols bgp peer-group LEAFS-UNDERLAY address-family ipv4-unicast  
set protocols bgp peer-group LEAFS-UNDERLAY remote-as 'external'  
set protocols bgp system-as '65000'
```

SPINE -1

```
set protocols bgp listen range 10.0.0.0/24 peer-group 'LEAFS-UNDERLAY'
```

SPINE -2

```
set protocols bgp listen range 10.0.1.0/24 peer-group 'LEAFS-UNDERLAY'
```

Overlay configuration

This scenario uses symmetric IRB with the single VXLAN device approach.

1. Create a VRF and set VNI to it:

```
set vrf name RED table '111'  
set vrf name RED vni '1000'
```

2. Create a VXLAN interface:

```
set interfaces vxlan vxlan0 mtu '9000'  
set interfaces vxlan vxlan0 parameters neighbor-suppress  
set interfaces vxlan vxlan0 parameters nolearning  
set interfaces vxlan vxlan0 parameters vni-filter
```



```
set interfaces vxlan vxlan0 source-address '10.10.10.1'
set interfaces vxlan vxlan0 source-interface 'lo'
```

3. Create a bridge interface with sub-interfaces.

Add the PC-facing Ethernet interfaces (each with its native VLAN) and the VXLAN interface to the bridge, assign a unique MAC address to the bridge, and map each sub-interface to the VRF.

```
set interfaces bridge br0 enable-vlan
set interfaces bridge br0 mac '00:bb:bb:bb:bb:ba'
set interfaces bridge br0 member interface eth5 native-vlan '100'
set interfaces bridge br0 member interface eth6 native-vlan '200'
set interfaces bridge br0 member interface vxlan0 disable-learning
set interfaces bridge br0 mtu '9000'
set interfaces bridge br0 vif 100 mtu '9000'
set interfaces bridge br0 vif 100 vrf 'RED'
set interfaces bridge br0 vif 200 mtu '9000'
set interfaces bridge br0 vif 200 vrf 'RED'
set interfaces bridge br0 vif 1000 mtu '9000'
set interfaces bridge br0 vif 1000 vrf 'RED'
```

4. Map VLAN to VNI in the VXLAN configuration:

```
set interfaces vxlan vxlan0 parameters external
set interfaces vxlan vxlan0 vlan-to-vni 100 vni '100'
set interfaces vxlan vxlan0 vlan-to-vni 200 vni '200'
set interfaces vxlan vxlan0 vlan-to-vni 1000 vni '1000'
```

5. Configure EVPN:

Shared LEAF configuration

```
set protocols bgp address-family l2vpn-evpn advertise-all-vni
set protocols bgp neighbor 10.10.10.100 peer-group 'SPINES-OVERLAY'
set protocols bgp neighbor 10.10.10.101 peer-group 'SPINES-OVERLAY'
set protocols bgp peer-group SPINES-OVERLAY address-family l2vpn-evpn
set protocols bgp peer-group SPINES-OVERLAY ebgp-multihop '3'
set protocols bgp peer-group SPINES-OVERLAY remote-as '65000'
set protocols bgp peer-group SPINES-OVERLAY update-source 'lo'
```

Shared SPINE configuration

```
set protocols bgp address-family l2vpn-evpn advertise-all-vni
set protocols bgp listen range 10.10.10.0/24 peer-group 'LEAFS-OVERLAY'
set protocols bgp peer-group LEAFS-OVERLAY address-family l2vpn-evpn attribute-unchanged next-hop
set protocols bgp peer-group LEAFS-OVERLAY ebgp-multihop '3'
set protocols bgp peer-group LEAFS-OVERLAY remote-as 'external'
set protocols bgp peer-group LEAFS-OVERLAY update-source 'lo'
```

6. Configure Route Distinguishers (RD) and Route Targets (RT) on LEAFs:

LEAF-1



```
set protocols bgp address-family l2vpn-evpn vni 100 rd '65001:100'
set protocols bgp address-family l2vpn-evpn vni 100 route-target both '1000:100'
set protocols bgp address-family l2vpn-evpn vni 200 rd '65001:200'
set protocols bgp address-family l2vpn-evpn vni 200 route-target both '1000:200'
set vrf name RED protocols bgp address-family ipv4-unicast redistribute connected
set vrf name RED protocols bgp address-family l2vpn-evpn advertise ipv4 unicast
set vrf name RED protocols bgp address-family l2vpn-evpn rd '65001:1000'
set vrf name RED protocols bgp address-family l2vpn-evpn route-target both '1000:1000'
set vrf name RED protocols bgp system-as '65001'
```

LEAF-2

```
set protocols bgp address-family l2vpn-evpn vni 100 rd '65002:100'
set protocols bgp address-family l2vpn-evpn vni 100 route-target both '1000:100'
set protocols bgp address-family l2vpn-evpn vni 200 rd '65002:200'
set protocols bgp address-family l2vpn-evpn vni 200 route-target both '1000:200'
set vrf name RED protocols bgp address-family ipv4-unicast redistribute connected
set vrf name RED protocols bgp address-family l2vpn-evpn advertise ipv4 unicast
set vrf name RED protocols bgp address-family l2vpn-evpn rd '65002:1000'
set vrf name RED protocols bgp address-family l2vpn-evpn route-target both '1000:1000'
set vrf name RED protocols bgp system-as '65002'
```

LEAF-3

```
set protocols bgp address-family l2vpn-evpn vni 100 rd '65003:100'
set protocols bgp address-family l2vpn-evpn vni 100 route-target both '1000:100'
set protocols bgp address-family l2vpn-evpn vni 200 rd '65003:200'
set protocols bgp address-family l2vpn-evpn vni 200 route-target both '1000:200'
set vrf name RED protocols bgp address-family ipv4-unicast redistribute connected
set vrf name RED protocols bgp address-family l2vpn-evpn advertise ipv4 unicast
set vrf name RED protocols bgp address-family l2vpn-evpn rd '65003:1000'
set vrf name RED protocols bgp address-family l2vpn-evpn route-target both '1000:1000'
set vrf name RED protocols bgp system-as '65003'
```

Anycast gateway IP and MAC configuration

Configure the anycast gateway IP and MAC addresses for each VLAN on all LEAFs where the VLAN is present.

```
set interfaces pseudo-ethernet peth100 address '192.168.1.1/24'
set interfaces pseudo-ethernet peth100 anycast-gateway
set interfaces pseudo-ethernet peth100 ip enable-arp-accept
set interfaces pseudo-ethernet peth100 mac '00:aa:aa:aa:aa:aa'
set interfaces pseudo-ethernet peth100 mtu '9000'
set interfaces pseudo-ethernet peth100 source-interface 'br0.100'
set interfaces pseudo-ethernet peth100 vrf 'RED'
set interfaces pseudo-ethernet peth200 address '192.168.2.1/24'
set interfaces pseudo-ethernet peth200 anycast-gateway
set interfaces pseudo-ethernet peth200 ip enable-arp-accept
set interfaces pseudo-ethernet peth200 mac '00:aa:aa:aa:aa:aa'
set interfaces pseudo-ethernet peth200 mtu '9000'
set interfaces pseudo-ethernet peth200 source-interface 'br0.200'
set interfaces pseudo-ethernet peth200 vrf 'RED'
```

Validation and troubleshooting

Underlay

The result of the underlay configuration should be the distribution of loopback IP addresses. They will be used as IP addresses for overlay BGP EVPN sessions.

1. Verify that the LEAF has established BGP sessions to all SPINES:

```
vyos@LEAF-1:~$ show ip bgp summary

IPv4 Unicast Summary:
BGP router identifier 10.10.10.1, local AS number 65001 VRF default vrf-id 0
BGP table version 5
RIB entries 9, using 1440 bytes of memory
Peers 2, using 58 KiB of memory
Peer groups 2, using 144 bytes of memory
```

Neighbor Desc	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd	PfxSnt
10.0.0.0	4	65000	87	87	5	0	0	01:20:42	3	5 N/A
10.0.1.0	4	65000	87	87	5	0	0	01:20:42	3	5 N/A

```
Total number of neighbors 2
```

2. Verify that the SPINE has established BGP sessions to all LEAFs:

```
vyos@SPINE-1:~$ show ip bgp summary

IPv4 Unicast Summary:
BGP router identifier 10.10.10.100, local AS number 65000 VRF default vrf-id 0
BGP table version 4
RIB entries 7, using 1120 bytes of memory
Peers 3, using 87 KiB of memory
Peer groups 2, using 144 bytes of memory
```

Neighbor Desc	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd	PfxSnt
*10.0.0.1	4	65001	92	92	4	0	0	01:25:17	1	4 N/A
*10.0.0.3	4	65002	92	92	4	0	0	01:25:14	1	4 N/A
*10.0.0.5	4	65003	92	92	4	0	0	01:25:12	1	4 N/A

```
Total number of neighbors 3
* - dynamic neighbor
3 dynamic neighbor(s), limit 100
```

3. Verify that the LEAF receives the loopback IP addresses of all nodes via BGP:

```
vyos@LEAF-1:~$ show ip route
Codes: K - kernel route, C - connected, L - local, S - static,
R - RIP, O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
f - OpenFabric, t - Table-Direct,
```



> - selected route, * - FIB route, q - queued, r - rejected, b - backup
t - trapped, o - offload failure

IPv4 unicast VRF default:

```
C>* 10.0.0.0/31 is directly connected, eth0, weight 1, 01:27:12
L>* 10.0.0.1/32 is directly connected, eth0, weight 1, 01:27:12
C>* 10.0.1.0/31 is directly connected, eth1, weight 1, 01:27:08
L>* 10.0.1.1/32 is directly connected, eth1, weight 1, 01:27:08
L * 10.10.10.1/32 is directly connected, lo, weight 1, 01:27:02
C>* 10.10.10.1/32 is directly connected, lo, weight 1, 01:27:02
B>* 10.10.10.2/32 [20/0] via 10.0.0.0, eth0, weight 1, 01:26:48
    *                               via 10.0.1.0, eth1, weight 1, 01:26:48
B>* 10.10.10.3/32 [20/0] via 10.0.0.0, eth0, weight 1, 01:26:46
    *                               via 10.0.1.0, eth1, weight 1, 01:26:46
B>* 10.10.10.100/32 [20/0] via 10.0.0.0, eth0, weight 1, 01:26:50
B>* 10.10.10.101/32 [20/0] via 10.0.1.0, eth1, weight 1, 01:26:50
```

← Received via BGP loopback IPs

4. Verify that the SPINE receives the loopback IP addresses of all LEAFs:

```
vyos@SPINE-1:~$ show ip route
```

Codes: K - kernel route, C - connected, L - local, S - static,
R - RIP, O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
f - OpenFabric, t - Table-Direct,
> - selected route, * - FIB route, q - queued, r - rejected, b - backup
t - trapped, o - offload failure

IPv4 unicast VRF default:

```
C>* 10.0.0.0/31 is directly connected, eth0, weight 1, 01:29:19
L>* 10.0.0.0/32 is directly connected, eth0, weight 1, 01:29:19
C>* 10.0.0.2/31 is directly connected, eth1, weight 1, 01:29:22
L>* 10.0.0.2/32 is directly connected, eth1, weight 1, 01:29:22
C>* 10.0.0.4/31 is directly connected, eth2, weight 1, 01:29:18
L>* 10.0.0.4/32 is directly connected, eth2, weight 1, 01:29:18
B>* 10.10.10.1/32 [20/0] via 10.0.0.1, eth0, weight 1, 01:28:53
B>* 10.10.10.2/32 [20/0] via 10.0.0.3, eth1, weight 1, 01:28:51
B>* 10.10.10.3/32 [20/0] via 10.0.0.5, eth2, weight 1, 01:28:49
L * 10.10.10.100/32 is directly connected, lo, weight 1, 01:29:23
C>* 10.10.10.100/32 is directly connected, lo, weight 1, 01:29:23
```

← Received via BGP loopback IPs

Overlay

1. Verify that the LEAF has established overlay BGPEVPN sessions with all SPINES using loopback IP addresses:

```
vyos@LEAF-1:~$ show bgp l2vpn evpn summary
```

BGP router identifier 10.10.10.1, local AS number 65001 VRF default vrf-id 0
BGP table version 0
RIB entries 21, using 3360 bytes of memory
Peers 2, using 58 KiB of memory
Peer groups 2, using 144 bytes of memory

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd	PfxSnt
Desc										



```

10.10.10.100  4      65000      173      174      4      0      0 02:41:03      8      12 N/A
10.10.10.101  4      65000      173      174      4      0      0 02:41:03      8      12 N/A

```

Total number of neighbors 2

2. Verify in LEAFs that all VTEPs are distributed via EVPN route type 3:

```

vyos@LEAF-1:~$ show bgp l2vpn evpn route type 3
BGP table version is 1, local router ID is 10.10.10.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN type-1 prefix: [1]:[EthTag]:[ESI]:[IPlen]:[VTEP-IP]:[Frag-id]
EVPN type-2 prefix: [2]:[EthTag]:[MAClen]:[MAC]:[IPlen]:[IP]
EVPN type-3 prefix: [3]:[EthTag]:[IPlen]:[OrigIP]
EVPN type-4 prefix: [4]:[ESI]:[IPlen]:[OrigIP]
EVPN type-5 prefix: [5]:[EthTag]:[IPlen]:[IP]

```

Network	Next Hop	Metric	LocPrf	Weight	Path
Route Distinguisher: 65001:100					
*> [3]:[0]:[32]:[10.10.10.1]	RD 65001:100				
10.10.10.1				32768	i
ET:8	RT:1000:100				
Route Distinguisher: 65001:200					
*> [3]:[0]:[32]:[10.10.10.1]	RD 65001:200				
10.10.10.1				32768	i
ET:8	RT:1000:200				
Route Distinguisher: 65002:100					
*> [3]:[0]:[32]:[10.10.10.2]	RD 65002:100				
10.10.10.2				0	65000 65002 i
RT:1000:100	ET:8				
*= [3]:[0]:[32]:[10.10.10.2]	RD 65002:100				
10.10.10.2				0	65000 65002 i
RT:1000:100	ET:8				
Route Distinguisher: 65002:200					
*> [3]:[0]:[32]:[10.10.10.2]	RD 65002:200				
10.10.10.2				0	65000 65002 i
RT:1000:200	ET:8				
*= [3]:[0]:[32]:[10.10.10.2]	RD 65002:200				
10.10.10.2				0	65000 65002 i
RT:1000:200	ET:8				
Route Distinguisher: 65003:100					
*> [3]:[0]:[32]:[10.10.10.3]	RD 65003:100				
10.10.10.3				0	65000 65003 i
RT:1000:100	ET:8				
*= [3]:[0]:[32]:[10.10.10.3]	RD 65003:100				
10.10.10.3				0	65000 65003 i
RT:1000:100	ET:8				
Route Distinguisher: 65003:200					
*> [3]:[0]:[32]:[10.10.10.3]	RD 65003:200				
10.10.10.3				0	65000 65003 i
RT:1000:200	ET:8				
*= [3]:[0]:[32]:[10.10.10.3]	RD 65003:200				
10.10.10.3				0	65000 65003 i
RT:1000:200	ET:8				

Displayed 6 prefixes (10 paths) (of requested type)

Test Anycast gateway and symmetric IRB

Run the ping command on PC1 to test connectivity with PC6. Since PC1 and PC6 are in different networks, symmetric routing should be used. This means that packets should be marked with a VRF VNI. In this topology, the VNI is 1000.

Run the same test on PC3, then check the ARP tables on PC1 and PC3. IP and MAC addresses of the default gateway on both hosts should be the same.

1. Run the ping command from PC1 to PC6, then from PC3 to PC6, and inspect the ARP table on both hosts.

PC1

```
PC1> ping 192.168.2.30
84 bytes from 192.168.2.30 icmp_seq=1 ttl=63 time=132.504 ms
84 bytes from 192.168.2.30 icmp_seq=2 ttl=62 time=2.870 ms
84 bytes from 192.168.2.30 icmp_seq=3 ttl=62 time=2.752 ms
84 bytes from 192.168.2.30 icmp_seq=4 ttl=62 time=2.629 ms
84 bytes from 192.168.2.30 icmp_seq=5 ttl=62 time=2.604 ms

PC1> show arp
00:aa:aa:aa:aa:aa 192.168.1.1 expires in 117 seconds
```

PC3

```
PC3> ping 192.168.2.30
84 bytes from 192.168.2.30 icmp_seq=1 ttl=62 time=3.145 ms
84 bytes from 192.168.2.30 icmp_seq=2 ttl=62 time=2.687 ms
84 bytes from 192.168.2.30 icmp_seq=3 ttl=62 time=2.570 ms
84 bytes from 192.168.2.30 icmp_seq=4 ttl=62 time=2.698 ms
84 bytes from 192.168.2.30 icmp_seq=5 ttl=62 time=2.517 ms

PC3> show arp
00:aa:aa:aa:aa:aa 192.168.1.1 expires in 118 seconds
```

2. Verify the EVPN Type-2 (MAC/IP) routes on LEAF-1. The MAC and IP addresses of both PC1 and PC6 should be present in the BGPEVPN table.

```
vyos@LEAF-1:~$ show bgp l2vpn evpn route type 2

BGP table version is 7, local router ID is 10.10.10.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN type-1 prefix: [1]:[EthTag]:[ESI]:[IPlen]:[VTEP-IP]:[Frag-id]
EVPN type-2 prefix: [2]:[EthTag]:[MAClen]:[MAC]:[IPlen]:[IP]
EVPN type-3 prefix: [3]:[EthTag]:[IPlen]:[OrigIP]
EVPN type-4 prefix: [4]:[ESI]:[IPlen]:[OrigIP]
EVPN type-5 prefix: [5]:[EthTag]:[IPlen]:[IP]

Network          Next Hop          Metric LocPrf Weight Path
Extended Community
```



```

Route Distinguisher: 65001:100
*> [2]:[0]:[48]:[00:50:79:66:68:00] RD 65001:100
    10.10.10.1 32768 i
    ET:8 RT:1000:100
*> [2]:[0]:[48]:[00:50:79:66:68:00]:[32]:[192.168.1.10] RD 65001:100
    10.10.10.1 32768 i
    ET:8 RT:1000:100 RT:1000:1000 Rmac:00:bb:bb:bb:bb:ba
Route Distinguisher: 65003:200
*> [2]:[0]:[48]:[00:50:79:66:68:04] RD 65003:200
    10.10.10.3 0 65000 65003 i
    RT:1000:200 ET:8
*= [2]:[0]:[48]:[00:50:79:66:68:04] RD 65003:200
    10.10.10.3 0 65000 65003 i
    RT:1000:200 ET:8
*> [2]:[0]:[48]:[00:50:79:66:68:04]:[32]:[192.168.2.30] RD 65003:200
    10.10.10.3 0 65000 65003 i
    RT:1000:200 RT:1000:1000 ET:8 Rmac:00:bb:bb:bb:bb:bc
*= [2]:[0]:[48]:[00:50:79:66:68:04]:[32]:[192.168.2.30] RD 65003:200
    10.10.10.3 0 65000 65003 i
    RT:1000:200 RT:1000:1000 ET:8 Rmac:00:bb:bb:bb:bb:bc

Displayed 4 prefixes (6 paths) (of requested type)

```

PC1 IP/MAC route

PC6 IP/MAC routes

3. Verify the EVPN ARP cache on LEAF-1. It shows IP/MAC and VTEP mapping.

```

vyos@LEAF-1:~$ show evpn arp-cache vni all
VNI 200 #ARP (IPv4 and IPv6, local and remote) 1

Flags: I=local-inactive, P=peer-active, X=peer-proxy
Neighbor      Type  Flags State  MAC              Remote ES/VTEP      Seq
# 's
192.168.2.30  remote      active  00:50:79:66:68:04 10.10.10.3        0/0

VNI 100 #ARP (IPv4 and IPv6, local and remote) 1

Flags: I=local-inactive, P=peer-active, X=peer-proxy
Neighbor      Type  Flags State  MAC              Remote ES/VTEP      Seq
# 's
192.168.1.10  local      active  00:50:79:66:68:00

```

4. Verify the ARP table on LEAF-1. ARPs received via VXLAN will be in the NOARP state.

```

vyos@LEAF-1:~$ show arp
Address      Interface  Link layer address  State
-----
10.10.10.3   br0.1000  00:bb:bb:bb:bb:bc   NOARP
10.0.0.0     eth0      0c:e6:b7:a2:00:00   STALE
10.0.1.0     eth1      0c:ea:c7:bd:00:00   REACHABLE
192.168.2.30 br0.200   00:50:79:66:68:04   NOARP
192.168.2.30 peth200   00:50:79:66:68:04   NOARP
10.10.10.3   vxlan0    00:bb:bb:bb:bb:bc   NOARP
192.168.1.10 peth100   00:50:79:66:68:00   REACHABLE

```

PC6 ARP entries

5. Repeat steps 2-4 on LEAF-3:

```
vyos@LEAF-3:~$ show bgp l2vpn evpn route type 2
BGP table version is 7, local router ID is 10.10.10.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN type-1 prefix: [1]:[EthTag]:[ESI]:[IPlen]:[VTEP-IP]:[Frag-id]
EVPN type-2 prefix: [2]:[EthTag]:[MAClen]:[MAC]:[IPlen]:[IP]
EVPN type-3 prefix: [3]:[EthTag]:[IPlen]:[OrigIP]
EVPN type-4 prefix: [4]:[ESI]:[IPlen]:[OrigIP]
EVPN type-5 prefix: [5]:[EthTag]:[IPlen]:[IP]
```

```

Network          Next Hop          Metric LocPrf Weight Path
Extended Community
Route Distinguisher: 65001:100
*> [2]:[0]:[48]:[00:50:79:66:68:00] RD 65001:100
    10.10.10.1          0 65000 65001 i
    RT:1000:100 ET:8
*= [2]:[0]:[48]:[00:50:79:66:68:00] RD 65001:100
    10.10.10.1          0 65000 65001 i
    RT:1000:100 ET:8
*> [2]:[0]:[48]:[00:50:79:66:68:00]:[32]:[192.168.1.10] RD 65001:100
    10.10.10.1          0 65000 65001 i
    RT:1000:100 RT:1000:1000 ET:8 Rmac:00:bb:bb:bb:bb:ba
*= [2]:[0]:[48]:[00:50:79:66:68:00]:[32]:[192.168.1.10] RD 65001:100
    10.10.10.1          0 65000 65001 i
    RT:1000:100 RT:1000:1000 ET:8 Rmac:00:bb:bb:bb:bb:ba

Route Distinguisher: 65003:200
*> [2]:[0]:[48]:[00:50:79:66:68:04] RD 65003:200
    10.10.10.3          32768 i
    ET:8 RT:1000:200
*> [2]:[0]:[48]:[00:50:79:66:68:04]:[32]:[192.168.2.30] RD 65003:200
    10.10.10.3          32768 i
    ET:8 RT:1000:200 RT:1000:1000 Rmac:00:bb:bb:bb:bb:bc

```

PC1 IP/MAC routes

PC6 IP/MAC route

Displayed 4 prefixes (6 paths) (of requested type)

```
vyos@LEAF-3:~$ show evpn arp-cache vni all
VNI 200 #ARP (IPv4 and IPv6, local and remote) 1
```

Flags: I=local-inactive, P=peer-active, X=peer-proxy

Neighbor	Type	Flags	State	MAC	Remote ES/VTEP	Seq
#'s						
192.168.2.30	local		active	00:50:79:66:68:04		0/0

```
VNI 100 #ARP (IPv4 and IPv6, local and remote) 1
```

Flags: I=local-inactive, P=peer-active, X=peer-proxy

Neighbor	Type	Flags	State	MAC	Remote ES/VTEP	Seq
#'s						
192.168.1.10	remote		active	00:50:79:66:68:00	10.10.10.1	0/0

```
vyos@LEAF-3:~$ show arp
```

Address	Interface	Link layer address	State
10.0.1.4	eth1	0c:ea:c7:bd:00:02	REACHABLE
10.0.0.4	eth0	0c:e6:b7:a2:00:02	REACHABLE



```

10.10.10.1   br0.1000   00:bb:bb:bb:bb:ba   NOARP
192.168.1.10 br0.100   00:50:79:66:68:00   NOARP
192.168.1.10 peth100  00:50:79:66:68:00   NOARP
192.168.2.30 peth200  00:50:79:66:68:04   REACHABLE
10.10.10.1   vxlan0    00:bb:bb:bb:bb:ba   NOARP
  
```

PC1 ARP entries

As BGP ECMP is used, ICMP requests and replies use different links. Requests use the path via SPINE-2. Replies use the path via SPINE-1. All ICMP packets are encapsulated in a VXLAN header with VRF VNI 1000.

No.	Time	Source	Protocol	Destination	Length	Text item	Info
7	13:09:47,588608	192.168.1.10	ICMP	192.168.2.30	148	✓	Echo (ping) request id=0x9bf9
8	13:09:48,591965	192.168.1.10	ICMP	192.168.2.30	148	✓	Echo (ping) request id=0x9cf9
9	13:09:49,595313	192.168.1.10	ICMP	192.168.2.30	148	✓	Echo (ping) request id=0x9df9
10	13:09:50,598588	192.168.1.10	ICMP	192.168.2.30	148	✓	Echo (ping) request id=0x9ef9
11	13:09:51,602080	192.168.1.10	ICMP	192.168.2.30	148	✓	Echo (ping) request id=0x9ff9
36	13:14:05,513749	192.168.1.10	ICMP	192.168.2.30	148	✓	Echo (ping) request id=0x9dfa
37	13:14:06,519282	192.168.1.10	ICMP	192.168.2.30	148	✓	Echo (ping) request id=0x9efa
38	13:14:07,522597	192.168.1.10	ICMP	192.168.2.30	148	✓	Echo (ping) request id=0x9ffa
39	13:14:08,525947	192.168.1.10	ICMP	192.168.2.30	148	✓	Echo (ping) request id=0xa0fa
40	13:14:09,529325	192.168.1.10	ICMP	192.168.2.30	148	✓	Echo (ping) request id=0xa1fa
41	13:14:10,532540	192.168.1.10	ICMP	192.168.2.30	148	✓	Echo (ping) request id=0xa2fa
44	13:14:11,535863	192.168.1.10	ICMP	192.168.2.30	148	✓	Echo (ping) request id=0xa3fa
45	13:14:12,539231	192.168.1.10	ICMP	192.168.2.30	148	✓	Echo (ping) request id=0xa4fa

>	Frame 7: Packet, 148 bytes on wire (1184 bits), 148 bytes captured (1184 bits) on interface -, id 0	
>	Ethernet II, Src: 0c:77:52:8f:00:01 (0c:77:52:8f:00:01), Dst: 0c:ea:c7:bd:00:00 (0c:ea:c7:bd:00:00)	0000 0c ea c7
>	Internet Protocol Version 4, Src: 10.10.10.1, Dst: 10.10.10.3	0010 00 86 33
>	User Datagram Protocol, Src Port: 58739, Dst Port: 4789	0020 0a 03 e5
>	Virtual eXtensible Local Area Network	0030 e8 00 00
>	Flags: 0x0800, VXLAN Network ID (VNI)	0040 45 00 00
>	Group Policy ID: 0	0050 c0 a8 02
>	VXLAN Network Identifier (VNI): 1000	0060 0c 0d 0e
>	Reserved: 0	0070 1c 1d 1e
>	Ethernet II, Src: 00:bb:bb:bb:bb:ba (00:bb:bb:bb:bb:ba), Dst: 00:bb:bb:bb:bb:bc (00:bb:bb:bb:bb:bc)	0080 2c 2d 2e
>	Internet Protocol Version 4, Src: 192.168.1.10, Dst: 192.168.2.30	0090 3c 3d 3e
>	Internet Control Message Protocol	

VRF VNI

No.	Time	Source	Protocol	Destination	Length	Text item	Info
8	13:09:47,590389	192.168.2.30	ICMP	192.168.1.10	148	✓	Echo (ping) reply id=0x9bf9
9	13:09:48,593713	192.168.2.30	ICMP	192.168.1.10	148	✓	Echo (ping) reply id=0x9cf9
10	13:09:49,597091	192.168.2.30	ICMP	192.168.1.10	148	✓	Echo (ping) reply id=0x9df9
11	13:09:50,600315	192.168.2.30	ICMP	192.168.1.10	148	✓	Echo (ping) reply id=0x9ef9
12	13:09:51,603789	192.168.2.30	ICMP	192.168.1.10	148	✓	Echo (ping) reply id=0x9ff9
41	13:14:05,517281	192.168.2.30	ICMP	192.168.1.10	148	✓	Echo (ping) reply id=0x9dfa
42	13:14:06,521017	192.168.2.30	ICMP	192.168.1.10	148	✓	Echo (ping) reply id=0x9efa
43	13:14:07,524527	192.168.2.30	ICMP	192.168.1.10	148	✓	Echo (ping) reply id=0x9ffa
44	13:14:08,527999	192.168.2.30	ICMP	192.168.1.10	148	✓	Echo (ping) reply id=0xa0fa

>	Frame 8: Packet, 148 bytes on wire (1184 bits), 148 bytes captured (1184 bits) on interface -, id 0	
>	Ethernet II, Src: 0c:e6:b7:a2:00:00 (0c:e6:b7:a2:00:00), Dst: 0c:77:52:8f:00:00 (0c:77:52:8f:00:00)	0000 0c 77 52
>	Internet Protocol Version 4, Src: 10.10.10.3, Dst: 10.10.10.1	0010 00 86 ad
>	User Datagram Protocol, Src Port: 59130, Dst Port: 4789	0020 0a 01 e6
>	Virtual eXtensible Local Area Network	0030 e8 00 00
>	Flags: 0x0800, VXLAN Network ID (VNI)	0040 45 00 00
>	Group Policy ID: 0	0050 c0 a8 01
>	VXLAN Network Identifier (VNI): 1000	0060 0c 0d 0e
>	Reserved: 0	0070 1c 1d 1e
>	Ethernet II, Src: 00:bb:bb:bb:bb:bc (00:bb:bb:bb:bb:bc), Dst: 00:bb:bb:bb:bb:ba (00:bb:bb:bb:bb:ba)	0080 2c 2d 2e
>	Internet Protocol Version 4, Src: 192.168.2.30, Dst: 192.168.1.10	0090 3c 3d 3e
>	Internet Control Message Protocol	

VRF VNI

Conclusion

The anycast gateway is a cornerstone of modern EVPN-based datacenter architectures, enabling seamless virtual machine mobility across racks, pods, or even geographically distributed datacenters without requiring any reconfiguration on the host VM.

This walkthrough demonstrated how to deploy and validate an anycast gateway on VyOS using the symmetric IRB model, which distributes routing responsibilities evenly across leaf nodes and ensures both high availability and low-latency forwarding. Beyond the initial configuration, it also covered how to verify proper operation through VyOS's built-in show commands and how to identify common misconfigurations before they impact production traffic.

Together, these building blocks form a solid foundation for any scalable, resilient EVPN fabric built on VyOS.

