

# Implement BGP EVPN Routing Policy on Catalyst 9000 Series Switches

## Contents

[Introduction](#)  
[Prerequisites](#)  
[Requirements](#)  
[Components Used](#)  
[Background Information](#)  
[Document Details](#)  
[Supported Route Types](#)  
[Terminology](#)  
[Configure](#)  
[Network Diagram](#)  
[Configurations](#)  
[Leaf-01 \(Base EVPN Config\)](#)  
[Leaf-01 \(Route Filtering\)](#)  
[CGW \(Base Config\)](#)  
[Verify](#)  
[Network Diagram](#)  
[Route-Type 3 Policy \(Leaf-01\)](#)  
[Local Routing \(CGW\)](#)  
[Troubleshoot](#)  
[Debug Route-Map Policy \(Inbound\)](#)  
[Verify Route-Map Policy \(Outbound\)](#)  
[Related Information](#)

## Introduction

This document describes how to implement routing policy in BGP EVPN VXLAN on Catalyst 9000 Series Switches.

## Prerequisites

### Requirements

Cisco recommends that you have knowledge of these topics:

- BGP configuration and validation
- Route-maps & basic BGP routing policy
- EVPN VxLAN

### Components Used

The information in this document is based on these software and hardware versions:

- Catalyst 9300
- Catalyst 9400

- Catalyst 9500
- Catalyst 9600
- Cisco IOS® XE 17.11.1 and later versions

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

## Background Information

EVPN VxLAN routing policy uses route-maps to control the traffic flow of hosts, and what routes VTEPs learn and process:

- This is a key aspect of the EVPN **protected segment** type design (outside the scope of this document) where ARP traffic is only sent to the Central Gateway (CGW) so the CGW can proxy reply with its own MAC address even for local segment traffic.
- Protected segment gives the ability to control traffic flows in such a way that all traffic can be passed through a Firewall's traffic inspection and traffic policies.

Cisco also implemented the Default Gateway (DEF GW) extended community attribute in order to indicate which MAC/IP prefix is owned by the CGW.

---

**Note:** This is required for EVPN protected segments, which is outside the scope of this document.

---

**Tip:** To learn more about Auto RT & Auto RD CLI simplification features see this document [Configure BGP VRF Auto RD Auto RT for EVPN on Catalyst 9000 Series Switches](#)

---

## Document Details

Key details about this document:

- Replication mode for BUM traffic is ingress (unicast BUM tunnels)
  - Vlans used are 201 & 202
  - VNIs used are 20101 & 20201
  - All routed traffic is sent to the CGW (VTEPs do not have their own SVI)
  - IMET route-type 3 BGP prefixes are hub and spoke (Leafs only accept IMET routes from CGW, not from each other)
  - Route-type 2 & 5 are full mesh
- 

**Note:** The design decision to allow full mesh MAC/IP prefixes are for mobility purposes. If the Leafs cannot see each other's RT2s, and a host moves from one Leaf to another it assumes it is a new prefix, rather than incrementing the sequence number of the existing RT2.

---

## Supported Route Types

- Route-type 3 (Inclusive Multicast Ethernet Tag route)
- Route-type 7 (IGMP Join Synch Routes)
- Route-type 8 (IGMP Leave Synch Routes)

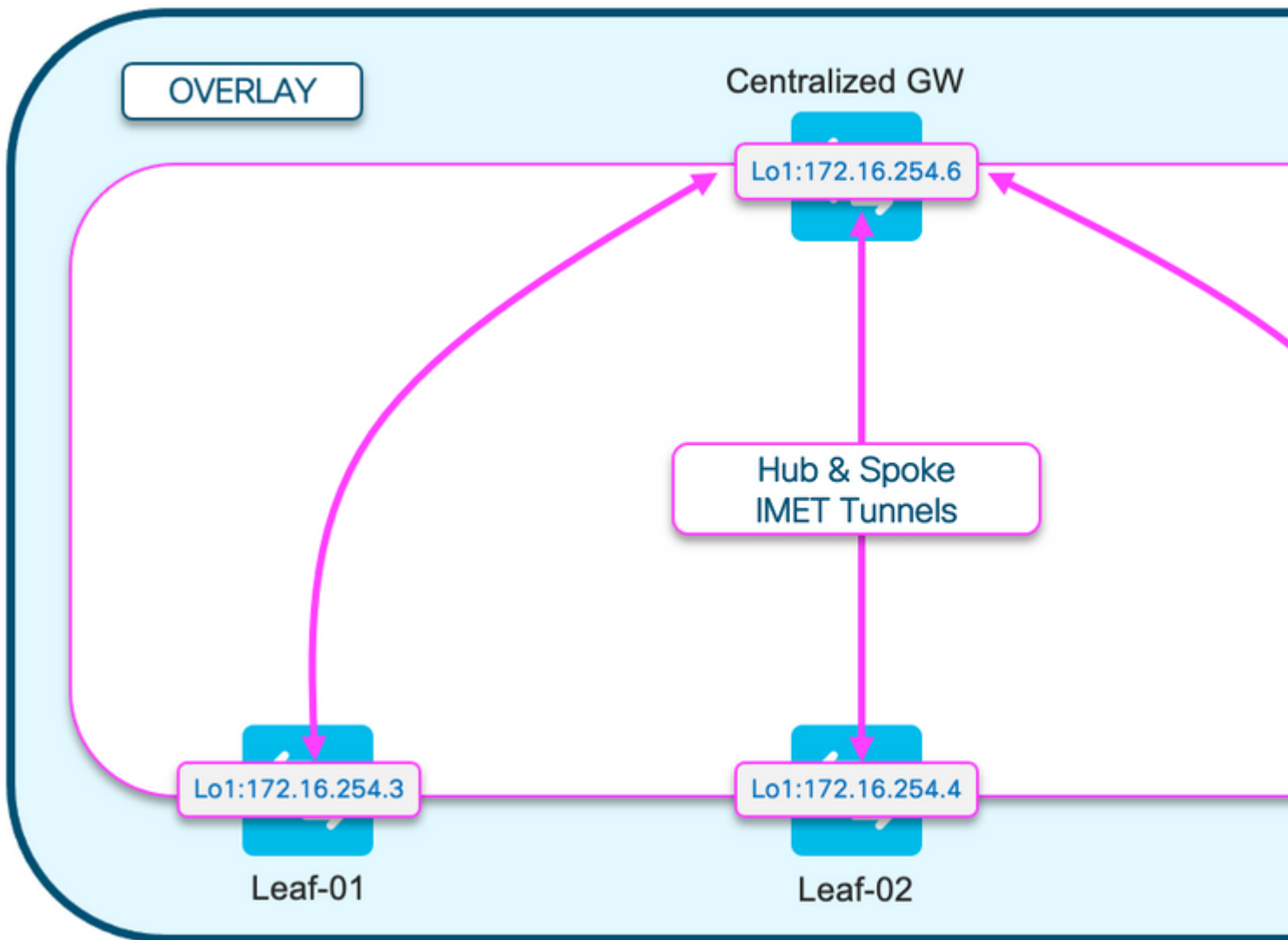
**Note: As part of this routing policy enhancement route-type 7 & route-type 8 are also supported.** This document only demonstrates how to match and filter route-type 3 prefixes, but the verification and methodology applies to all 3 types

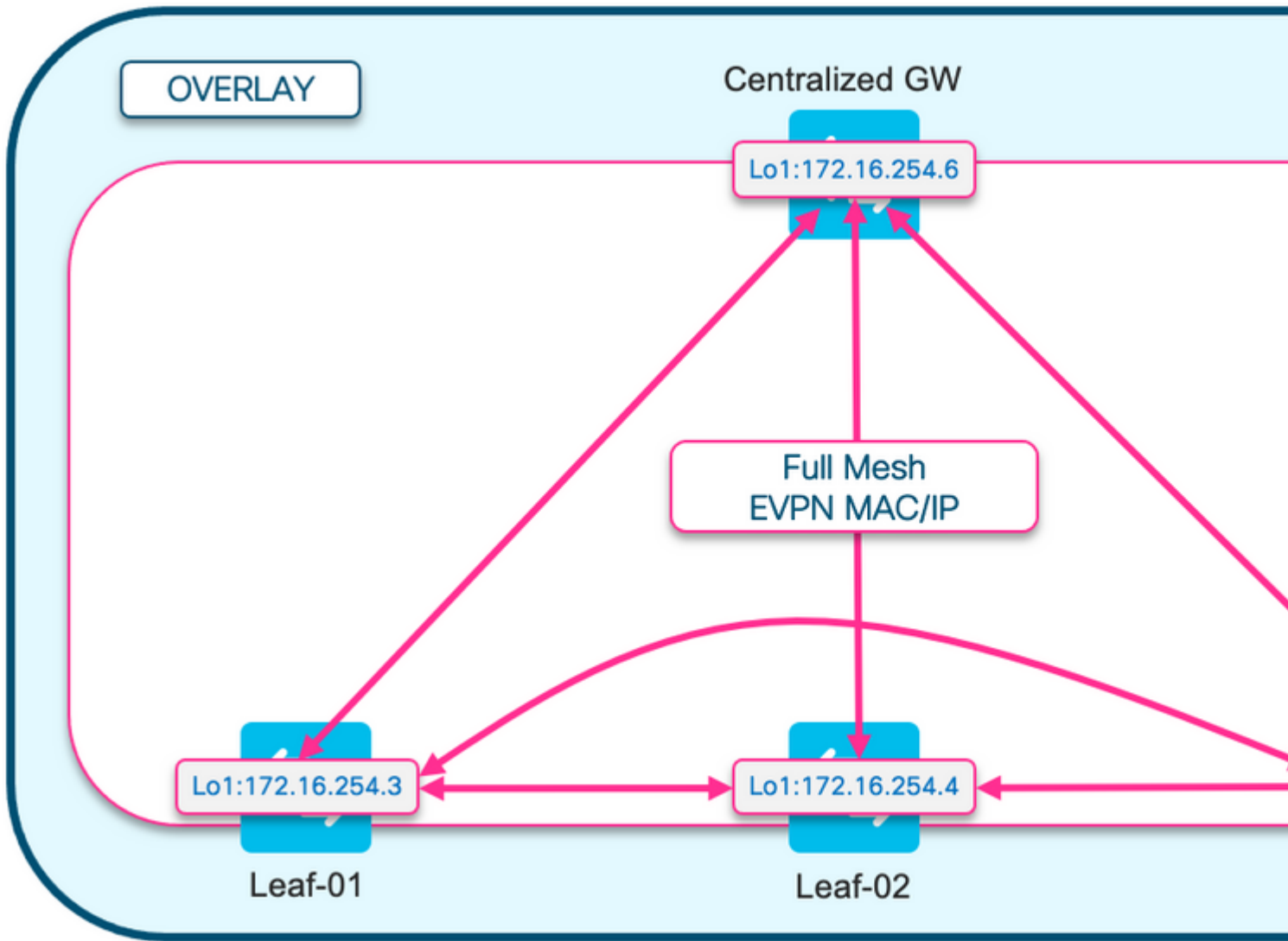
## Terminology

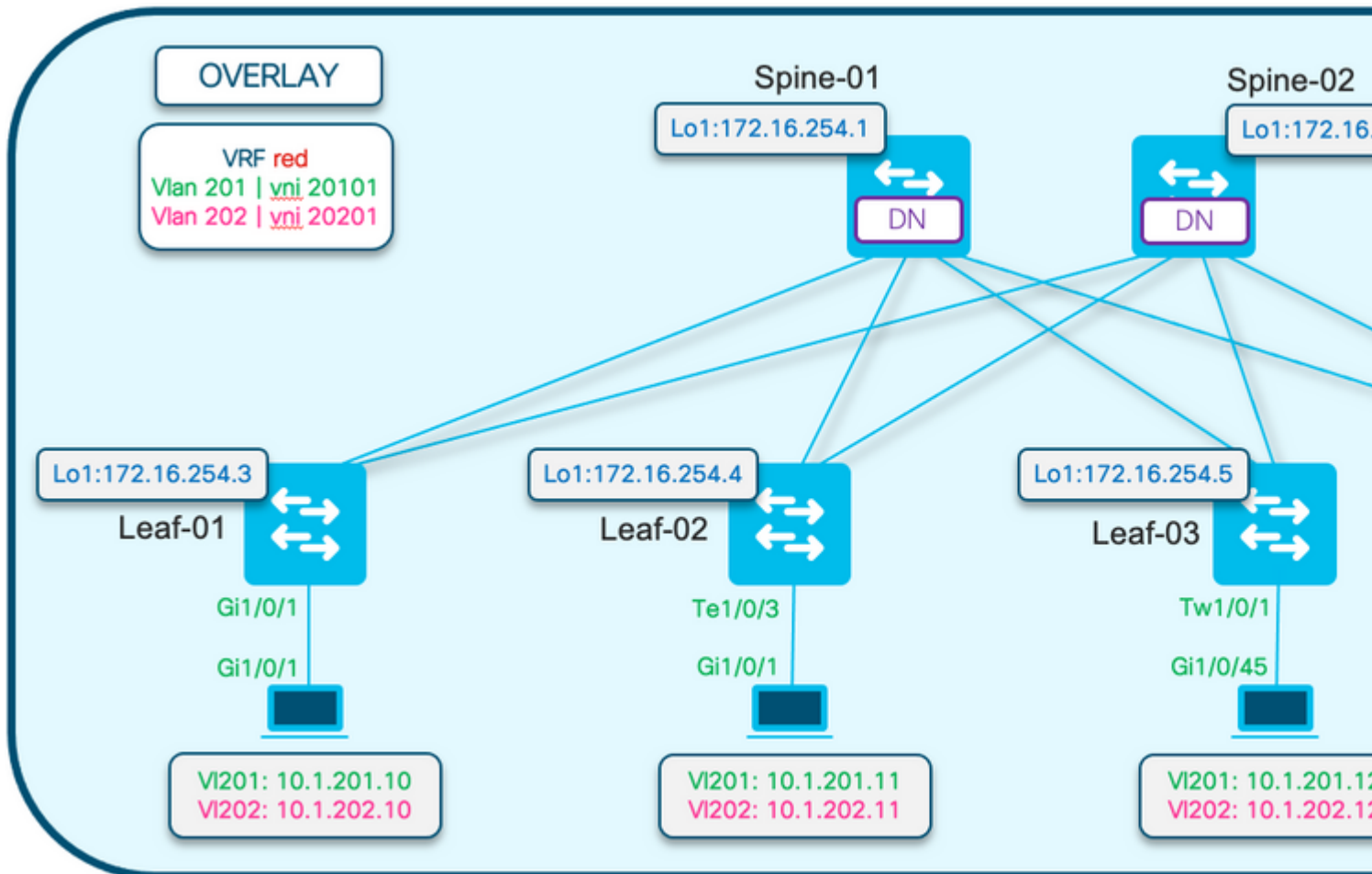
|               |   |   |
|---------------|---|---|
| <b>VRF</b>    | Virtual Routing Forwarding                  | Defines a layer 3 routing domain that be separated from other VRF and global IPv4/IPv6 routing domain   |
| <b>AF</b>     | Address Family                              | Defines which type prefixes and routing info BGP handles  |
| <b>AS</b>     | Autonomous System                           | A set of Internet routable IP prefixes that belong to a network or a collection of networks that are all managed, controlled and supervised by a single entity or organization  |
| <b>EVPN</b>   | Ethernet Virtual Private Network            | Extension that allows BGP to transport Layer 2 MAC and Layer 3 IP information is EVPN and uses Multi-Protocol Border Gateway Protocol (MP-BGP) as the protocol to distribute reachability information that pertains to the VXLAN overlay network.   |
| <b>VXLAN</b>  | Virtual Extensible LAN (Local Area Network) | VXLAN is designed to overcome the inherent limitations of VLANs and STP. It is a proposed IETF standard [RFC 7348] to provide the same Ethernet Layer 2 network services as VLANs do, but with greater flexibility. Functionally, it is a MAC-in-UDP encapsulation protocol that runs as a virtual overlay on a Layer 3 underlay network. |
| <b>CGW</b>    | Centralized Gateway                         | And implementation of EVPN where the gateway SVI are not on each leaf. Instead, all routing is done by a specific leaf using asymmetric IRB (Integrated Routing and Bridging)   |
| <b>DEF GW</b> | Default Gateway                             | A BGP extended community attribute added to the MAC/IP prefix via the command "default-gateway advertise enable" under the 'l2vpn evpn' configuration section.  |
| <b>IMET</b>   | Inclusive Multicast Ethernet Tag (Route)    | Also called BGP type-3 route. This route type is used in EVPN to deliver BUM (broadcast / unknown unicast / multicast) traffic between VTEPs.   |

## Configure

## Network Diagram







## Configurations

This section shows configuration samples from Leaf-01 & the Border Leaf (CGW) switches. Other leaf configurations are the same, so are not repeated here.

- The examples in this document are based on **ingress replication** and demonstrates how to apply policy to control what VTEPs can communicate broadcast and other flooded traffic

### Leaf-01 (Base EVPN Config)

```
<#root>
```

```
Leaf-01#
```

```
show run | sec l2vpn
l2vpn evpn
```

```
replication-type static
```

```
flooding-suppression address-resolution disable <-- Disables ARP caching so ARP is always sent up to t
```

```
router-id Loopback1
l2vpn evpn
```

```
instance 201
```

```
vlan-based
encapsulation vxlan
```

```
replication-type ingress
  multicast advertise enable
  l2vpn evpn
```

```
instance 202
  vlan-based
  encapsulation vxlan
```

```
replication-type ingress
  multicast advertise enable
```

```
<#root>
```

```
Leaf-01#
```

```
show run | sec vlan config
```

```
vlan configuration 201
  member evpn-instance 201 vni 20101
vlan configuration 202
  member evpn-instance 202 vni 20201
```

```
<#root>
```

```
Leaf-01#
```

```
show run int nve 1
```

```
Building configuration...
```

```
Current configuration : 327 bytes
```

```
!
interface nve1
  no ip address
  source-interface Loopback1
  host-reachability protocol bgp
  member vni 20201
```

```
ingress-replication
```

```
  member vni 20101
```

```
ingress-replication
```

```
end
```

```
<#root>
```

```
ip bgp-community new-format      <-- Required to see community in aa:nn format
```

## Leaf-01 (Route Filtering)

**Determine** what route-target to match.

- For CLI simplicity, the example in this document uses auto route-target functionality.
- Since this is automatic, run this command to know how to configure the community list

```
<#root>
```

```
Leaf-01#
```

```
show l2vpn evpn route-target
```

```
Route Target          EVPN Instances
```

```
65001:201
```

```
201
```

```
<-- Route-targets for the Vlan/VNI of interest
```

```
65001:202
```

```
202
```

```
<#root>
```

```
ip extcommunity-list expanded ALLOW-RT2 permit 65001:
```

```
20[0-9]
```

```
<-- match Route-targets 65001:200 - 65001:209
```

```
!
```

```
ip community-list standard BLOCK-RT3 permit 999:999
```

```
<-- Arbitrary RT used to mark IMET prefixes as they are advertised
```

```
<#root>
```

```
route-map POLICY-IN deny 5
  match community BLOCK-RT3 exact-match
```

```
<-- Deny prefixes that match RT 999:999 in standard community list
```

```
!
```

```
route-map POLICY-IN permit 10
```

```
<-- Permit any other prefixes that do not contain 999:999
```

```
!
```

```
route-map POLICY-OUT permit 5
  match extcommunity ALLOW-RT2
```

```
<-- Match the auto-RT in the extended community list
```

```
  match evpn route-type 3
```

```
<-- AND match route-type 3 prefixes
```

```
  set community 999:999
```

```
<-- Set additional standard community to advertised prefixes
```



```
!  
route-map POLICY-OUT permit 10  
  
<-- Permit prefixes that are not RT3 to be sent out without additional attributes added
```

```
<#root>
```

```
router bgp 65001  
  bgp router-id 172.16.255.3  
  bgp log-neighbor-changes  
  no bgp default ipv4-unicast  
  neighbor 172.16.255.1 remote-as 65001  
  neighbor 172.16.255.1 update-source Loopback0  
  neighbor 172.16.255.2 remote-as 65001  
  neighbor 172.16.255.2 update-source Loopback0  
  !  
  address-family ipv4  
  exit-address-family  
  !  
  address-family ipv4 mvpn  
    neighbor 172.16.255.1 activate  
    neighbor 172.16.255.1 send-community both  
    neighbor 172.16.255.2 activate  
    neighbor 172.16.255.2 send-community both  
  exit-address-family  
  !  
  
  address-family l2vpn evpn  
  
    neighbor 172.16.255.1 activate  
  
    neighbor 172.16.255.1 send-community both      <-- Send both standard and extended community attributes  
    neighbor 172.16.255.1 route-map POLICY-IN in  <-- Apply inbound policy to deny prefixes with 999:999  
    neighbor 172.16.255.1 route-map POLICY-OUT out <-- Apply outbound policy to match RT3 / apply standard  
  
    neighbor 172.16.255.2 activate  
    neighbor 172.16.255.2 send-community both  
  
  neighbor 172.16.255.2 route-map POLICY-IN in  
  neighbor 172.16.255.2 route-map POLICY-OUT out  
  
  exit-address-family
```

## CGW (Base Config)

```
<#root>
```

```
CGW#  
  
show running-config | beg l2vpn evpn instance 201  
  
l2vpn evpn instance 201 vlan-based  
  encapsulation vxlan  
  replication-type ingress
```

```
default-gateway advertise enable <-- adds the BGP attribute EVPN DEF GW:0:0 to the MAC/IP prefix
multicast advertise enable
!
l2vpn evpn instance 202 vlan-based
encapsulation vxlan
replication-type ingress

default-gateway advertise enable <-- adds the BGP attribute EVPN DEF GW:0:0 to the MAC/IP prefix
multicast advertise enable
```

<#root>

CGW#

```
show running-config | sec vlan config
```

```
vlan configuration 201
 member evpn-instance 201 vni 20101
vlan configuration 202
 member evpn-instance 202 vni 20201
```

<#root>

CGW#

```
show run int nve 1
```

Building configuration...

Current configuration : 313 bytes

!

```
interface nve1
```

```
 no ip address
```

```
 source-interface Loopback1
```

```
 host-reachability protocol bgp
```

```
 member vni 10101 mcast-group 225.0.0.101
```

```
 member vni 10102 mcast-group 225.0.0.102
```

```
 member vni 20101 ingress-replication local-routing <-- 'ingress-replication' (Unicast all BUM traffic)
```

```
 member vni 20201 ingress-replication local-routing
```

```
 member vni 50901 vrf green
```

```
end
```

<#root>

CGW#

```
show run interface vlan 201
```

Building configuration...

Current configuration : 231 bytes

!

```
interface Vlan201
```

```
mac-address 0000.beef.cafe      <-- MAC is static in this example for viewing simplicity. This is no
vrf forwarding red              <-- SVI is in VRF red
ip address 10.1.201.1 255.255.255.0
no ip redirects
ip local-proxy-arp             <-- Sets CGW to Proxy reply even for local subnet ARP requests
ip pim sparse-mode
ip route-cache same-interface   <-- This is auto added when local-proxy-arp is configured. However,
ip igmp version 3
no autostate
end
```

CGW#

```
show run interface vlan 202
```

Building configuration...

Current configuration : 163 bytes

```
!
interface Vlan202
  mac-address 0000.beef.cafe
  vrf forwarding red
  ip address 10.1.202.1 255.255.255.0
  no ip redirects
  ip local-proxy-arp
  ip pim sparse-mode
  ip route-cache same-interface
  ip igmp version 3
  no autostate
end
```

<#root>

CGW#

```
sh run vrf red
```

Building configuration...

Current configuration : 873 bytes

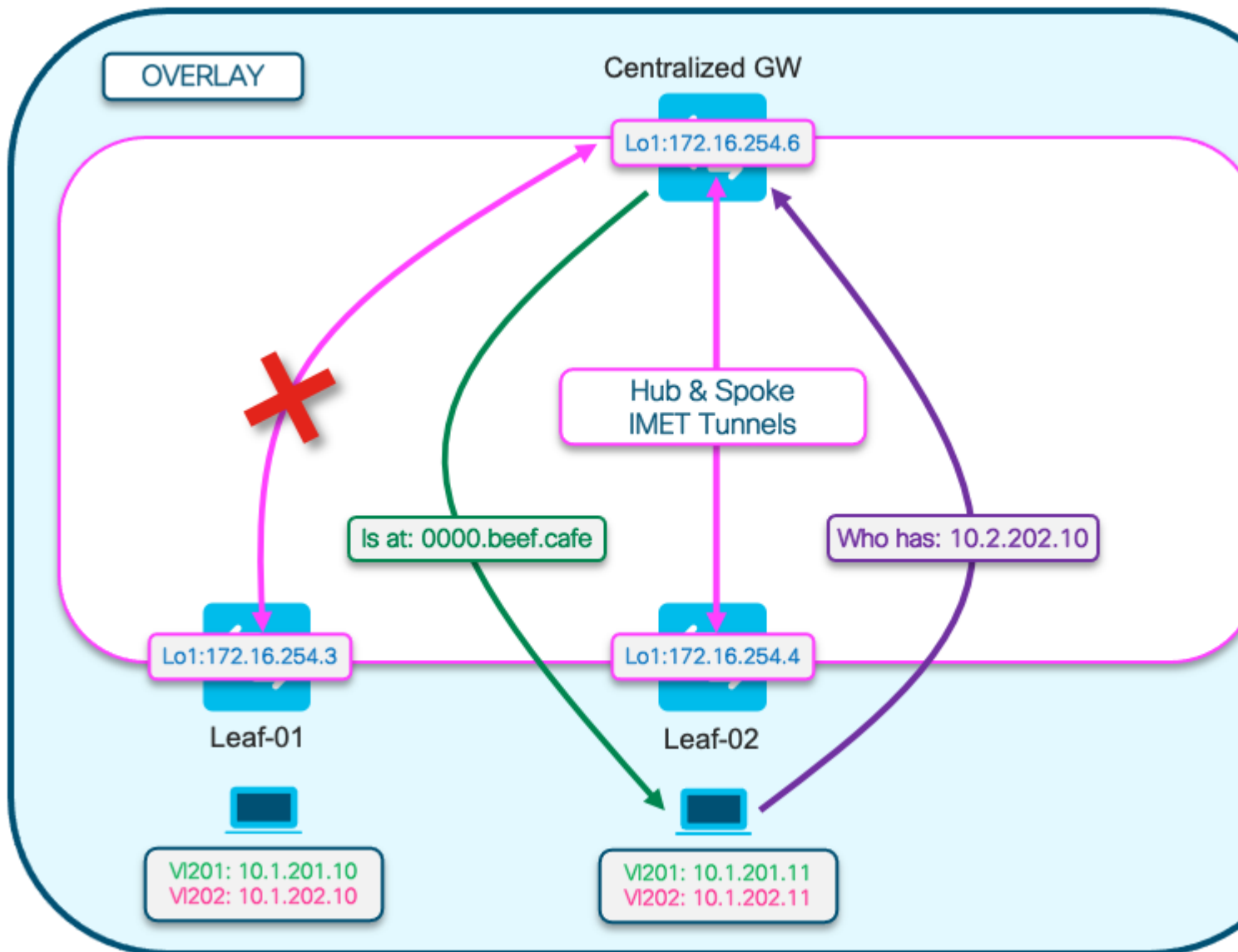
```
vrf definition red
```

```
rd 2:2
!
address-family ipv4
  route-target export 2:2
  route-target import 2:2
  route-target export 2:2 stitching
  route-target import 2:2 stitching
exit-address-family
```

**Note:** At the CGW there is no BGP policy applied. The CGW is allowed to receive and send all prefix types (RT2, RT5 / RT3).

## Verify

### Network Diagram



â€f

This diagram is to help visualize the flow of the ARP resolution process described in this section.

- The ARP Request is show in purple.
  - This ARP request is to resolve the MAC address of the hos 10.1.202.10 off Leaf-01
  - Notice that the purple line terminates at the CGW, and does not reach Leaf-01
- The ARP Reply is shown in green and contains the MAC of the CGW SVI for Vlan 202
  - Notice that the green line comes from the CGW, not from the actual host
- The red X is to indicate that this communication did not involve sending traffic to Leaf-01

## Route-Type 3 Policy (Leaf-01)

**Verify** that the route-maps applied to Leafs is properly filtering. We should only see IMET prefixes from the CGW, and no other leafs.

- Only one leaf is shown here, but this check should be done on all relevant leafs

---

**Note:** Once the route-maps are applied BGP must be cleared to the neighbors for the policy settings to take effect.

---

There are 2 methods to see what route-type 3 are installed:

- **Checking** BGP entries
- **Checking** l2route

### Check BGP entries (Leaf-01)

```
<#root>
```

```
Leaf-01#
```

```
show bgp l2vpn evpn route-type 3 | inc Tunnel End
```

```
    PMSI Attribute: Flags:0x0, Tunnel type:IR, length 4, vni:20101, tunnel identifier:
```

```
< Tunnel Endpoint: 172.16.254.6 >
```

```
<-- No RT3 prefixes present other than the CGW 172.16.254.6
```

```
    PMSI Attribute: Flags:0x0, Tunnel type:IR, length 4, vni:20201, tunnel identifier:
```

```
< Tunnel Endpoint: 172.16.254.6 >
```

```
    PMSI Attribute: Flags:0x0, Tunnel type:IR, length 4, vni:20101, tunnel identifier:
```

```
< Tunnel Endpoint: 172.16.254.6 >
```

```
    PMSI Attribute: Flags:0x0, Tunnel type:IR, length 4, vni:20201, tunnel identifier:
```

```
< Tunnel Endpoint: 172.16.254.6 >
```

### Check l2route

```
<#root>
```

```
Leaf-01-F241.03.23-9300#
```

```
show l2route evpn imet
```

| EVI | ETAG | Prod | Router IP Addr | Type | Label |
|-----|------|------|----------------|------|-------|
|-----|------|------|----------------|------|-------|

---

|     |   |  |  |  |  |
|-----|---|--|--|--|--|
| 201 | 0 |  |  |  |  |
|-----|---|--|--|--|--|

BGP

```

172.16.254.6      6      20101
172.16.254.6
No
<-- Only remote IMET producer is the CGW
201      0      L2VPN      172.16.254.3      6      20101
202      0
BGP
172.16.254.6      6      20201
172.16.254.6
No
202      0      L2VPN      172.16.254.3      6      20201
<-- Only remote IMET producer is the CGW

```

---

**Note:** The validation of route-map filters is done only on the access VTEPs. The CGW accepts all Type-3 prefixes so does not implement any route-maps.

---

## Local Routing (CGW)

Verify that the CGW replies with its own MAC when local segment ARP resolution occurs.

- There is no show command other than verification of configurations shown in configure section
- We can verify this via confirming what MAC is programmed on the host when it resolves ARP for a host on another Leaf.

Capture of an ARP resolution request sent from host off Leaf-02 to host off Leaf-01 (This was done using EPC capture on the Leaf-02 access interface facing the host)

- Leaf-01 host actual MAC: 0006.f601.cd45
- CGW MAC that responded to the ARP request: 0000.beef.cafe

```
<#root>
```

```
Leaf-02-F241.03.23-9400#
```

```
show mon cap 1 buff br | i ARP
```

```
32
```

```
10.356291 00:06:f6:17:ee:c4 -> ff:ff:ff:ff:ff:ff ARP 64
```

```
Who has 10.1.202.10? Tell 10.1.202.11 <-- ARP request from Leaf-02 host
```

```
33
```

```
10.357140 00:00:be:ef:ca:fe -> 00:06:f6:17:ee:c4 ARP 68 10.1.202.10
```

```
is at 00:00:be:ef:ca:fe <-- ARP reply is the CGW MAC
```

Leaf-02-F241.03.23-9400#

show mon cap 1 buff det | b Frame 32

Frame 32: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface /tmp/epc\_ws/wif\_to\_ts\_p  
<...snip...>

[Protocols in frame: eth:ethertype:vlan:ethertype:arp]

Ethernet II, Src: 00:06:f6:17:ee:c4 (00:06:f6:17:ee:c4), Dst: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)

Destination: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)

Address: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)

.... ..1. .... = LG bit: Locally administered address (this is NOT the factory default)

.... ..1 .... = IG bit: Group address (multicast/broadcast)

Source: 00:06:f6:17:ee:c4 (00:06:f6:17:ee:c4)

Address: 00:06:f6:17:ee:c4 (00:06:f6:17:ee:c4)

.... ..0. .... = LG bit: Globally unique address (factory default)

.... ..0 .... = IG bit: Individual address (unicast)

Type: 802.1Q Virtual LAN (0x8100)

802.1Q Virtual LAN, PRI: 0, DEI: 0,

ID: 202 <-- Vlan 202

000. .... = Priority: Best Effort (default) (0)

...0 .... = DEI: Ineligible

.... 0000 1100 1010 = ID: 202

Type: ARP (0x0806)

Padding: 00000000000000000000000000000000

Trailer: 00000000

Address Resolution Protocol (request)

<-- ARP Request

Hardware type: Ethernet (1)

Protocol type: IPv4 (0x0800)

Hardware size: 6

Protocol size: 4

Opcode: request (1)

Sender MAC address: 00:06:f6:17:ee:c4 (00:06:f6:17:ee:c4)

Sender IP address: 10.1.202.11

Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)

Target IP address: 10.1.202.10

<-- Leaf-02 Host

Frame 33: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface /tmp/epc\_ws/wif\_to\_ts\_p  
<...snip...>

Ethernet II, Src: 00:00:be:ef:ca:fe (00:00:be:ef:ca:fe), Dst: 00:06:f6:17:ee:c4 (00:06:f6:17:ee:c4)

Destination: 00:06:f6:17:ee:c4 (00:06:f6:17:ee:c4)

Address: 00:06:f6:17:ee:c4 (00:06:f6:17:ee:c4)

.... ..0. .... = LG bit: Globally unique address (factory default)

.... ..0 .... = IG bit: Individual address (unicast)

Source: 00:00:be:ef:ca:fe (00:00:be:ef:ca:fe)

Address: 00:00:be:ef:ca:fe (00:00:be:ef:ca:fe)

.... ..0. .... = LG bit: Globally unique address (factory default)

.... ..0 .... = IG bit: Individual address (unicast)

Type: CiscoMetaData (0x8909)

Cisco MetaData

Version: 1

Length: 1

Options: 0x0001

SGT: 0

Type: ARP (0x0806)

Padding: 000000000000000000000000

Trailer: 0000000000000000

#### Address Resolution Protocol (reply)

Hardware type: Ethernet (1)  
Protocol type: IPv4 (0x0800)  
Hardware size: 6  
Protocol size: 4  
Opcode: reply (2)

Sender MAC address: 00:00:be:ef:ca:fe (00:00:be:ef:ca:fe) <-- ARP Reply is the CGW MAC 0000.beef.c

Sender IP address: 10.1.202.10

Target MAC address: 00:06:f6:17:ee:c4 (00:06:f6:17:ee:c4)

<-- MAC of host off Leaf-02 0006.f617.eec4

Target IP address: 10.1.202.11

## Troubleshoot

### Debug Route-Map Policy (Inbound)

**Confirm** inbound route-maps are working as expected

**Enable** L2VPN EVPN update debugs

```
<#root>
```

```
Leaf-02#
```

```
debug bgp l2vpn evpn updates
```

```
BGP updates debugging is on for address family: L2VPN E-VPN
```

```
Leaf-02#
```

```
debug bgp l2vpn evpn updates events
```

```
BGP update events debugging is on for address family: L2VPN E-VPN
```

**Clear** bgp address-family to instantiate the Route policy

```
<#root>
```

```
Leaf-02#
```

```
clear bgp l2vpn evpn *
```

**Verify** Route-type 3 are only accepted from the CGW & denied from all other Leafs

```
<#root>
```



Leaf-02#

```
show log | i rcvd \[3\]
```

```
*Jul  4 06:40:41.556: BGP(10): 172.16.255.2
```

```
rcvd [3][172.16.254.6:202][0][32][172.16.254.6]/17 <-- Only accepted Type-3 is from the CGW (172.16.2
```

```
*Jul  4 06:40:41.557: BGP(10): 172.16.255.2
```

```
rcvd [3][172.16.254.6:201][0][32][172.16.254.6]/17
```

```
*Jul  4 06:40:41.557: BGP(10): 172.16.255.2
```

```
rcvd [3][172.16.254.3:202][0][32][172.16.254.3]/17 -- DENIED due to: route-map;
```

```
*Jul  4 06:40:41.557: BGP(10): 172.16.255.2
```

```
rcvd [3][172.16.254.5:202][0][32][172.16.254.5]/17 -- DENIED due to: route-map;
```

```
*Jul  4 06:40:41.557: BGP(10): 172.16.255.2
```

```
rcvd [3][172.16.254.3:201][0][32][172.16.254.3]/17 -- DENIED due to: route-map;
```

```
*Jul  4 06:40:41.557: BGP(10): 172.16.255.2
```

```
rcvd [3][172.16.254.5:201][0][32][172.16.254.5]/17 -- DENIED due to: route-map;
```

## Verify Route-Map Policy (Outbound)

**Verify** that the standard community is applied to the Type 3 prefixes from Leafs by checking them on the CGW.

- This sample output shows BGP prefixes from Leaf-01 & Leaf-02 with the community attribute injected for Vlan 202 segment

**Tip:** Remember, the CGW does not filter any prefixes so we can verify the complete BGP table from the CGW perspective.

```
<#root>
```

```
CGW#
```

```
show bgp l2vpn evpn route-type 3
```

```
BGP routing table entry for [3][172.16.254.6:202][0][32][172.16.254.3]/17, version 461855
```

```
Paths: (1 available, best #1,
```

```
table evi_202
```

```
)
```

```
<-- The EVI context for the vlan 202 segment
```

```
Not advertised to any peer
```

```
Refresh Epoch 4
```

```
Local, imported path from [3][172.16.254.3:202][0][32][172.16.254.3]/17 (global)
```

```
172.16.254.3
```

```

(metric 3) (via default) from 172.16.255.2 (172.16.255.2)
  Origin incomplete, metric 0, localpref 100, valid, internal, best
    Community: 999:999
    Extended Community:
RT:65001:202
ENCAP:8 EVPN Mcast Flags:1
  Originator: 172.16.255.3, Cluster list: 172.16.255.2
  PMSI Attribute: Flags:0x0,
Tunnel type:IR
, length 4, vni:20201, tunnel identifier: <
Tunnel Endpoint: 172.16.254.3
>
<-- Type-3 tunnel

  rx pathid: 0, tx pathid: 0x0
  Updated on Jan 22 2025 19:02:18 UTC
BGP routing table entry for [3][172.16.254.6:202][0][32][172.16.254.4]/17, version 605955
Paths: (1 available, best #1,
table evi_202
)
  Not advertised to any peer
  Refresh Epoch 4
  Local, imported path from [3][172.16.254.4:202][0][32][172.16.254.4]/17 (global)
172.16.254.4
(metric 3) (via default) from 172.16.255.2 (172.16.255.2)
  Origin incomplete, metric 0, localpref 100, valid, internal, best
    Community: 999:999
    Extended Community:
RT:65001:202
ENCAP:8 EVPN Mcast Flags:1
  Originator: 172.16.255.4, Cluster list: 172.16.255.2
  PMSI Attribute: Flags:0x0, Tunnel type:IR, length 4, vni:20201, tunnel identifier: <
Tunnel Endpoint: 172.16.254.4
>
  rx pathid: 0, tx pathid: 0x0
  Updated on Jan 30 2025 18:50:49 UTC

```

## Related Information

- [Technical Support & Documentation - Cisco Systems](#)
- [BGP EVPN VXLAN Configuration Guide, Cisco IOS XE Dublin 17.11.x \(Catalyst 9500 Switches\)](#)
- [Catalyst EVPN Support Docs \(LogAdvisor\)](#)