Configure External VRF Connectivity and Route Leaking

This chapter contains these sections:

External VRF connectivity

External Layer 3 connections for VXLAN BGP EVPN fabrics

External Layer 3 connections are network extension methods that

- use per-VRF IP routing to provide connectivity between a VXLAN BGP EVPN fabric and external networks,
- commonly implement VRF Lite or Inter-AS Option A for Layer 3 extension, and
- enable scalable, segmented, and secure inter-domain routing across data center boundaries.

The terms “VRF Lite” and “Inter-AS Option A” both refer to techniques for back-to-back VRF connectivity. These enable the logical separation of traffic and policy controls at the peering boundaries of two domains (such as between a data center fabric and an external IP/MPLS backbone).

For example, connecting a tenant VRF in the VXLAN BGP EVPN fabric to a WAN router via a point-to-point link using VRF Lite enables the tenant to communicate externally while maintaining traffic separation from other tenants.

VXLAN BGP EVPN fabrics and VRF-lite mechanisms

A VXLAN BGP EVPN fabric is a network overlay architecture that:

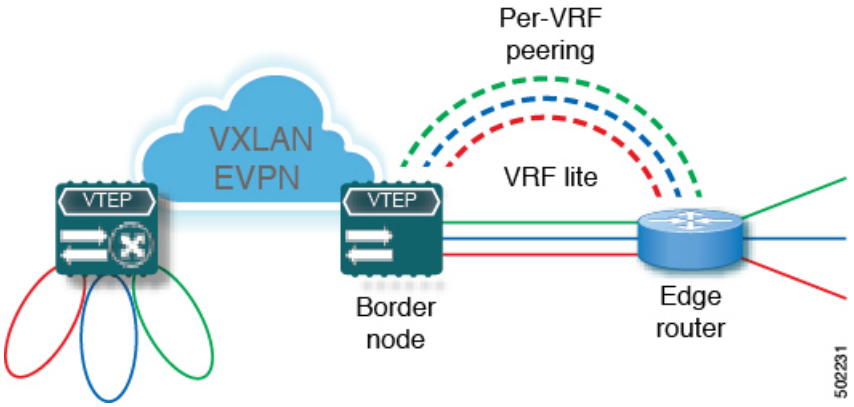
- enables scalable Layer 2 and Layer 3 segmentation across a data center network,
- uses MP-BGP with the EVPN address family as the control plane for route exchange between all edge devices (VTEPs) and route reflectors, and

- supports external connectivity by exporting prefixes from MP-BGP EVPN to IPv4/IPv6 per-VRF peerings toward external routers.

External connectivity and per-VRF peering in VXLAN BGP EVPN fabrics

In this architecture, the Edge devices (VTEPs) acting as border nodes handle external routing tasks. Various routing protocols can be used for per-VRF peering. eBGP is typically preferred, but IGPs such as OSPF, IS-IS, or EIGRP can be used if redistribution is implemented.

Figure 1: External Layer-3 connectivity – VRF-lite



Guidelines for external VRF connectivity and route leaking

These guidelines and limitations apply to external Layer 3 connectivity for VXLAN BGP EVPN fabrics.

- External VRF connectivity is supported on Cisco Nexus 9504 and 9508 platform switches with Cisco Nexus 96136YC-R and 9636C-RX line cards.
- Use a physical Layer 3 interface (parent interface) for external Layer 3 connectivity (VRF default) if permitted.
- Beginning with Cisco NX-OS Release 9.3(5), configure VTEPs to support VXLAN-encapsulated traffic only over parent interfaces if subinterfaces are configured.
- Do not use a parent interface to multiple subinterfaces for external Layer 3 connectivity (such as Ethernet1/1 for a VRF default). Use a subinterface instead.
- Do not configure VTEPs to support VXLAN-encapsulated traffic over subinterfaces, regardless of VRF participation or IEEE 802.1Q encapsulation.
- Do not mix subinterfaces for VXLAN and non-VXLAN VLANs.
- Do not use the **import map** command under address-family ipv4 unicast to control what gets imported into the EVPN table L3VNI counterpart.
- If TRM is configured, do not use SVIs to interconnect to the external router.

VXLAN BGP EVPN with eBGP for VRF-lite

Configure a VRF for VXLAN routing and external connectivity using BGP

Use this task when you need to enable inter-VRF or external connectivity for tenants in a VXLAN-based fabric.

Before you begin

Ensure you have:

- Access the border node CLI with appropriate privileges.
- Identified required VRF names, VNI numbers, and BGP configuration details.

Procedure

Step 1 Enter configuration mode: **configure terminal**

Step 2 Create or select the VRF context: **vrf context** *vrf-name*

Step 3 Specify the Layer 3 VNI for the VRF: **vni** *number*

The VNI associated with the VRF is often referred to as a Layer 3 VNI, L3VNI, or L3VPN. The L3VNI is configured as the common identifier across the participating VTEPs.

Step 4 Assign a route distinguisher (RD): **rd** {**auto** | *rd*}

RD uniquely identifies a VTEP within an L3VNI. Supported formats: ASN2:NN, ASN4:NN, or IPV4:NN.

Step 5 Configure the address family for either IPv4 or IPv6: **address-family ipv4 unicast** or **address-family ipv6 unicast**

Sets up the specified IPv4 or IPv6 unicast address family.

Step 6 Set the route target (RT) for import and export: **route-target both** {**auto** | *rt*}

RT is used for per-VRF prefix import and export policy. Supported formats: ASN2:NN, ASN4:NN, or IPV4:NN. Manually configured RTs are required to support asymmetric VNIs.

Step 7 (Optional) Set RTs specifically for EVPN: **route-target both** {**auto** | *rt*} **evpn**

Step 8 Repeat steps 2 to 7 for each required L3VNI.

The VRF and associated Layer 3 VNI(s) are configured on the border node, ready for VXLAN routing and connectivity via BGP. For a configuration example, see Example of VXLAN BGP EVPN eBGP VRF-lite connectivity, on page 13.

What to do next

Verify the configuration and ensure BGP neighbors are established.

Configure the L3VNI fabric-facing VLAN and SVI on the border node

Before you begin

Ensure you have:

- Access the border node CLI with appropriate privileges.
- Identified the required VLAN ID, VN-Segment ID, SVI number, IP addresses, and VRF context for each L3VNI.

Use this procedure to configure a VLAN and SVI in global configuration mode for VXLAN EVPN L3VNI routing on a border node.

Procedure

Step 1 Create the VLAN for the L3VNI: **vlan** *vlan_id*

Step 2 Map the L3VNI to the VLAN for VXLAN EVPN routing: **vn-segment** *vn_segment_id*

Step 3 Configure the SVI (Switch Virtual Interface) for the VLAN: **interface vlan** *svi_number*

Step 4 Set the MTU value for VXLAN requirements: **mtu** *mtu_value*

The recommended MTU is 9216.

Step 5 Associate the SVI with the required VRF context: **vrf member** *vrf_name*

Step 6 Disable ICMP redirects: **no ip redirects**

Step 7 Enable IPv4 forwarding on the interface: **ip forward**

Step 8 (Optional) Assign the IPv6 address to the SVI: **ipv6 address** *ipv6_address*

Step 9 Disable ICMPv6 redirects: **no ipv6 redirects**

The L3VNI is mapped to a VLAN and SVI on the border node, with appropriate VRF and forwarding settings for VXLAN EVPN routing. For a configuration example, see Example of VXLAN BGP EVPN eBGP VRF-lite connectivity, on page 13.

What to do next

Repeat this process for each required L3VNI.

Configure the VTEP on the border node

Before you begin

Ensure you have administrator access to the switch CLI.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Step 2 Access the NVE interface configuration mode: **interface nve1**

Step 3 Add the Layer 3 VNI to the overlay and associate it with the appropriate tenant VRF: **member vni** *vni* **associate-vrf**

Repeat this command for each tenant VRF by replacing the vni with the required VNI number.

The border node VTEP is configured for all required tenant VRFs with their associated Layer-3 VNIs. For a configuration example, see Example of VXLAN BGP EVPN eBGP VRF-lite connectivity, on page 13.

Configure a BGP VRF instance for IPv4 per-VRF peering on the border node

Perform this procedure on each border node that participates in the fabric to support external IPv4 connectivity for specific VRFs.

Before you begin

Ensure you have:

- Access to the border node CLI with appropriate privileges.
- A list of VRF names (L3VNIs) that require IPv4 peering.
- Autonomous system (AS) numbers and neighbor IP addresses.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Step 2 Create the BGP process using the local AS number: **router bgp** *autonomous-system-number*

Valid AS number: 1–4294967295.

Step 3 Configure these settings for each VRF that requires IPv4 peering.

a) Specify the VRF context: **vrf** *vrf-name*
b) Configure the IPv4 unicast address family: **address-family ipv4 unicast**
c) Enable advertisement of EVPN routes in the IPv4 address family: **advertise l2vpn evpn**
d) (Optional) Configure ECMP for the required iBGP and eBGP prefixes.

Table 1: ECMP command for iBGP and eBGP prefixes

If	Then
iBGP	maximum-paths ibgp *number* For iBGP, the number can range from 1 to 64.
eBGP	**maximum-paths** *number*

e) Define the eBGP neighbor and remote AS number: **neighbor** *ip-address* **remote-as** *remote-as-number*
f) Specify the interface for eBGP peering (update source): **update-source** *type or id*
g) Activate the IPv4 address family for prefix exchange: **address-family ipv4 unicast**

Note
Repeat the VRF configuration for each L3VNI that requires IPv4 external connectivity.

Step 4 Exit configuration mode and save your changes.

The border node is configured for BGP VRF-based IPv4 per-VRF peering and can establish external IPv4 connectivity. For a configuration example, see Example of VXLAN BGP EVPN eBGP VRF-lite connectivity, on page 13.

What to do next

Verify the BGP configuration and neighbor status.

Configure a BGP VRF instance for IPv6 per-VRF peering on the border node

Perform this procedure on each border node that participates in the fabric to support external IPv6 connectivity for specific VRFs.

Before you begin

Ensure you have:

- Access to the border node CLI with appropriate privileges.
- A list of VRF names (L3VNIs) that require IPv6 peering.
- Autonomous system (AS) numbers and neighbor IP addresses.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Step 2 Create the BGP process using the local AS number: **router bgp** *autonomous-system-number*

Valid AS number: 1–4294967295.

Step 3 Configure these settings for each VRF that requires IPv6 peering.

a) Specify the VRF context: **vrf** *vrf-name*
b) Configure the IPv6 unicast address family: **address-family ipv6 unicast**
c) Enable advertisement of EVPN routes in the ipv6 address family: **advertise l2vpn evpn**
d) (Optional) Configure ECMP for the required iBGP and eBGP prefixes.

Table 2: ECMP command for iBGP and eBGP prefixes

If	Then
iBGP	maximum-paths ibgp *number* For iBGP, the number can range from 1 to 64.
eBGP	**maximum-paths** *number*

e) Define the eBGP neighbor and remote AS number: **neighbor** *ip-address* **remote-as** *remote-as-number*
f) Specify the interface for eBGP peering (update source): **update-source** *type or id*
g) Activate the IPv6 address family for prefix exchange: **address-family ipv6 unicast**

Note
Repeat the VRF configuration for each L3VNI that requires IPv6 external connectivity.

Step 4 Exit configuration mode and save your changes.

The border node is configured for BGP VRF-based IPv6 per-VRF peering and can establish external IPv6 connectivity. For a configuration example, see Example of VXLAN BGP EVPN eBGP VRF-lite connectivity, on page 13.

What to do next

Verify the BGP configuration and neighbor status.

Configure the sub-interface instance for per-VRF peering on the border node

Use this task when you need to set up per-VRF peering using sub-interfaces for individual VRF contexts on a Cisco NX-OS device.

Before you begin

Ensure you have:

- Access the border node CLI with appropriate privileges.
- Identified the parent interface, sub-interface identifier, VRF names, VLAN ID, and IP address for each peering.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Step 2 Configure the parent interface and enable Layer 3 mode.

a) Enter the interface configuration mode for the parent interface: **interface Ethernet** *port*
b) Disable Layer 2 switching on the interface: **no switchport**
c) Bring up the parent interface: **no shutdown**
d) Exit interface configuration mode: **exit**

Step 3 Configure the sub-interface for the specific VRF peering.

a) Enter the sub-interface configuration mode: interface Ethernet *port*
b) Configure the VLAN encapsulation for the sub-interface: **encapsulation dot1q** *vlan-id*
c) Associate the sub-interface with the VRF context: **vrf member** *vrf-name*
d) Assign the IP address to the sub-interface: ip address *ip-address subnet-mask*
e) Bring up the sub-interface: **no shutdown**

Step 4 Repeat the sub-interface configuration for each additional per-VRF peering.

The border node is ready for per-VRF peering, with each VRF mapped to a unique sub-interface. For a configuration example, see Example of VXLAN BGP EVPN eBGP VRF-lite connectivity, on page 13.

What to do next

Verify connectivity and routing for each configured VRF instance.

VXLAN BGP EVPN - default-route, route filtering on external connectivity

Default routes for external connectivity

A default route for external connectivity is a network routing method that:

- enables outbound traffic to reach destinations outside the local network,
- uses route advertisements within a VXLAN BGP EVPN fabric to establish a default forwarding path, and
- incorporates route filtering to prevent default-route propagation beyond intended boundaries.

When advertising a default route into a VXLAN BGP EVPN fabric, ensure that the default route remains internal to the fabric and is not advertised externally. Implement route filtering mechanisms to prevent unintended distribution of the default route outside the fabric. This approach maintains the integrity and security of network routing policies.

Configure the default route in the border nodes VRF

Before you begin

- Identify the VRF name to configure.
- Identify the next-hop IPv4 address, IPv6 address, or both to use for the default route.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Step 2 Specify the VRF context you want to configure: **vrf context** *vrf-name*

Step 3 Configure the IPv4 default route for the VRF: **ip route 0.0.0.0/0** *next-hop-ipv4*

Step 4 (Optional) Configure the IPv6 default route for the VRF: **ipv6 route 0::/0** *next-hop-ipv6*

The specified VRF now includes default IPv4 or IPv6 routes. Packets with unknown destinations are forwarded to the next-hop gateway. For a configuration example, see Route advertisement and filtering features for VXLAN BGP EVPN border nodes, on page 15.

Configure the BGP VRF instance for IPv4/IPv6 default-route advertisement on the border node

Perform this task on each border node requiring default-route advertisement for external connectivity in environments leveraging Layer 3 virtual networks (L3VNIs).

Before you begin

Ensure you have administrator access to the switch CLI.

Obtain the autonomous system numbers, VRF names, and BGP neighbor information.

Procedure

Step 1 Enter global configuration mode:

```
configure terminal
```

Step 2 Configure BGP under the appropriate autonomous system, then specify the VRF:

```
router bgp <autonomous-system-number>
vrf <vrf-name>
```

Step 3 Enable the IPv4 unicast address family, then advertise IPv4 default routes within the VRF:

```
address-family ipv4 unicast
network 0.0.0.0/0
```

Step 4 Enable the IPv6 unicast address family, then advertise IPv4 and IPv6 default routes within the VRF:

```
address-family ipv6 unicast
network ::/0
```

Step 5 Define the eBGP neighbor and remote AS, and specify the update source interface if required:

```
neighbor <ip-address> remote-as <remote-as-number>
update-source <interface>
```

Step 6 Activate the IPv4 or IPv6 address family for this neighbor:

```
address-family {ipv4 | ipv6} unicast
```

Step 7 (Optional) Attach a route-map for egress route filtering on the neighbor:

```
route-map <name> out
```

Step 8 Repeat steps 2–7 for each VRF and L3VNI that require external connectivity and default-route advertisement.

Step 9 Save the configuration.

```
copy running-config startup-config
```

The BGP VRF instance on the border node now advertises IPv4 and IPv6 default routes to eBGP neighbors, enabling proper external connectivity and route propagation for your VRFs. For a configuration example, see Route advertisement and filtering features for VXLAN BGP EVPN border nodes, on page 15.

Configure route filtering for IPv4 default-route advertisement

Use this task to prevent a switch from advertising the IPv4 default route via external connectivity, while permitting other routes.

Before you begin

Ensure you have:

- Access the border node CLI with appropriate privileges.
- Identified required prefix-list names and route-map names.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Accesses system configuration to allow changes.

Step 2 Create an IPv4 prefix list to identify the default route: **ip prefix-list** *name* **seq 5 permit 0.0.0.0/0**

Defines a filter that matches the IPv4 default route.

Step 3 Create a route-map with a deny policy for the default route: **route-map** *name* **deny 10**

Configures the route-map to block advertisements matching the default route.

Step 4 Match the prefix list in the route-map: **match ip address prefix-list** *name*

Associates the prefix-list filter with the deny statement.

Step 5 Add a permit policy at the end of the route-map to allow all other routes: **route-map** *name* **permit 1000**

Permits advertisement of routes that do not match the default route.

The switch filters out IPv4 default routes from advertisement via external connectivity and permits non-default routes. For a configuration example, see Route advertisement and filtering features for VXLAN BGP EVPN border nodes, on page 15.

What to do next

Verify route advertisements using appropriate show commands and monitor routing updates to confirm correct filtering.

Configure route filtering for IPv6 default-route advertisement

Use this task to prevent a switch from advertising the IPv6 default route via external connectivity, while permitting other routes.

Before you begin

Ensure you have:

- Access the border node CLI with appropriate privileges.
- Identified required prefix-list names and route-map names.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Accesses system configuration to allow changes.

Step 2 Create an IPv6 prefix list to identify the default route: **ipv6 prefix-list** *name* **seq 5 permit 0::/0**

Defines a filter that matches the IPv6 default route.

Step 3 Create a route-map with a deny policy for the default route: **route-map** *name* **deny 10**

Configures the route-map to block advertisements matching the default route.

Step 4 Match the IPv6 prefix list in the route-map: **match ipv6 address prefix-list** *name*

Associates the prefix-list filter with the deny statement.

Step 5 Add a permit policy at the end of the route-map to allow all other routes: **route-map** *name* **permit 1000**

Permits advertisement of routes that do not match the default route.

The switch filters out IPv6 default routes from advertisement via external connectivity and permits non-default routes. For a configuration example, see Route advertisement and filtering features for VXLAN BGP EVPN border nodes, on page 15.

What to do next

Verify route advertisements using appropriate show commands and monitor routing updates to confirm correct filtering.

Default-route distribution and host-route filters

A default-route distribution and host-route filter is a route management feature in VXLAN BGP EVPN fabrics that:

- advertises all known routes to external network connections by default,
- allows selective filtering to prevent advertisement of IPv4 /32 or IPv6 /128 host routes when not beneficial, and
- helps administrators control which routes are shared externally to optimize scalability and network policy enforcement.

In some network scenarios, advertising specific host routes (such as IPv4 /32 or IPv6 /128) to external networks can lead to unnecessary complexity or resource usage. The host-route filter feature enables precise control over which routes are included in external advertisements, supporting streamlined connectivity and efficient network management.

If a network only requires default routes to be advertised externally, administrators can enable host-route filtering to suppress unnecessary host-specific routes, reducing the size and complexity of the route table advertised to upstream peers.

Without applying host-route filters, a VXLAN BGP EVPN fabric might share all detailed host routes externally, which can overwhelm upstream devices and result in suboptimal routing decisions.

Configure the BGP VRF instance for IPv4/IPv6 host-route filtering on the border node

This task enables granular route filtering by configuring BGP for designated VRFs with route-maps.

Before you begin

Ensure you have:

- Access to the border node CLI with appropriate privileges.
- Required autonomous system (AS) number, VRF names, neighbor IP addresses, remote AS numbers, and interface identifiers.
- Preconfigured route-maps (for example, "permitall").

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Step 2 Configure BGP with your AS number: **router bgp** *autonomous-system-number*

Initializes BGP process for your AS.

Step 3 Specify the VRF to configure: **vrf** *vrf-name*

Step 4 Define the eBGP neighbor and remote AS: **neighbor** *IP address* **remote-as** *remote-AS-number*

Sets up peering with an external BGP neighbor.

Step 5 Specify the update-source interface for peering: **update-source** *interface*

Step 6 Activate the IPv4 or IPv6 address family for prefix exchange: **address-family** {**ipv4** | **ipv6**} **unicast**

Step 7 Attach a route-map for egress route filtering: **route-map** *route-map-name* **out**

Step 8 Repeat steps 3 to 7 for each L3VNI that requires host-route filtering.

The border node BGP VRF instances are configured with host-route filtering for IPv4/IPv6 as specified, applying the selected route-maps. For a configuration example, see Route advertisement and filtering features for VXLAN BGP EVPN border nodes, on page 15.

Configure route filtering for IPv4 host-route advertisement

Use this task to prevent a switch from advertisement of IPv4 host routes by configuring prefix lists and route maps.

Before you begin

Ensure you have:

- Access the border node CLI with appropriate privileges.
- Identified required prefix-list names and route-map names.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Accesses system configuration to allow changes.

Step 2 Create an IPv4 prefix list to identify the default route: **ip prefix-list** *name* **seq 5 permit 0.0.0.0/0 eq 32**

Defines a filter that matches the IPv4 host route.

Step 3 Create a route-map with a deny policy for the host route: **route-map** *name* **deny 10**

Configures the route-map to block advertisements matching the host route.

Step 4 Match the prefix list in the route-map: **match ip address prefix-list** *name*

Associates the prefix-list filter with the deny statement.

Step 5 Add a permit policy at the end of the route-map to allow all other routes: **route-map** *name* **permit 1000**

Permits advertisement of routes that do not match the host route.

The IPv4 host-route is filtered and will not be advertised via external connectivity. For a configuration example, see Route advertisement and filtering features for VXLAN BGP EVPN border nodes, on page 15 .

Configure route filtering for IPv6 host-route advertisement

Use this task to prevent a switch from advertisement of IPv6 host routes by configuring prefix lists and route maps.

Before you begin

Ensure you have:

- Access the border node CLI with appropriate privileges.
- Identified required prefix-list names and route-map names.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Accesses system configuration to allow changes.

Step 2 Create an IPv6 prefix list to identify the default route: **ipv6 prefix-list** *name* **seq 5 permit 0::/0 eq 128**

Defines a filter that matches the IPv6 host route.

Step 3 Create a route-map with a deny policy for the host route: **route-map** *name* **deny 10**

Configures the route-map to block advertisements matching the host route.

Step 4 Match the IPv6 prefix list in the route-map: **match ipv6 address prefix-list** *name*

Associates the prefix-list filter with the deny statement.

Step 5 Add a permit policy at the end of the route-map to allow all other routes: **route-map** *name* **permit 1000**

Permits advertisement of routes that do not match the host route.

The IPv6 host route is filtered and will not be advertised via external connectivity. For a configuration example, see Route advertisement and filtering features for VXLAN BGP EVPN border nodes, on page 15 .

Example of VXLAN BGP EVPN eBGP VRF-lite connectivity

This example demonstrates how to establish external connectivity from a VXLAN BGP EVPN fabric to an external router using eBGP and VRF-lite.

Key configuration facts:

- The VXLAN BGP EVPN border node acts as the neighbor device to the external router.

- You can configure a local VRF name on the border node that is different from the VRF name on the external router. The L3VNI, however, must remain consistent across the VXLAN BGP EVPN fabric.
- The configuration supports both IPv4 and IPv6 (dual-stack); you can use either protocol as needed.

Example configuration:

```
vrf context myvrf_50001
  vni 50001
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
  address-family ipv6 unicast
    route-target both auto
    route-target both auto evpn
!
vlan 2000
  vn-segment 50001
!
interface Vlan2000
  no shutdown
  mtu 9216
  vrf member myvrf_50001
  no ip redirects
  ip forward
  ipv6 address use-link-local-only
  no ipv6 redirects
!
interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback1
  member vni 50001 associate-vrf
!
router bgp 65002
  vrf myvrf_50001
    router-id 192.0.2.6
    address-family ipv4 unicast
      advertise l2vpn evpn
      maximum-paths ibgp 2
      maximum-paths 2
    address-family ipv6 unicast
      advertise l2vpn evpn
      maximum-paths ibgp 2
      maximum-paths 2
    neighbor 192.0.2.95
      remote-as 65099
      address-family ipv4 unicast
    neighbor 2001:DB8::95/64
      remote-as 65099
      address-family ipv4 unicast
!
interface Ethernet1/3
  no switchport
  no shutdown
interface Ethernet1/3.2
  encapsulation dot1q 2
  vrf member myvrf_50001
  ip address 192.0.2.31/24
  ipv6 address 2001:DB8::31/64
  no shutdown
```

Route advertisement and filtering features for VXLAN BGP EVPN border nodes

VXLAN BGP EVPN border nodes support advertising IPv4 and IPv6 default routes within the fabric. These nodes also provide route filtering capabilities for external connectivity. This capability allows administrators to control which routes are propagated to external routers. It improves network security and reduces unnecessary route advertisement.

Key features:

- Advertises IPv4 0.0.0.0/0 and IPv6 ::/0 default routes from the fabric to external peers.
- Filters host routes (IPv4 /32 and IPv6 /128) to prevent advertising them to external routers.
- Allows use of prefix lists and route maps in BGP configurations to selectively permit or deny route advertisements.

Configuration summary:

- Prefix lists for default and host routes.

```
ip prefix-list default-route seq 5 permit 0.0.0.0/0 le 1
ipv6 prefix-list default-route-v6 seq 5 permit 0::/0
!
ip prefix-list host-route seq 5 permit 0.0.0.0/0 eq 32
ipv6 prefix-list host-route-v6 seq 5 permit 0::/0 eq 128
```

- Route maps to control route advertisement.

```
route-map extcon-rmap-filter deny 10
  match ip address prefix-list default-route
route-map extcon-rmap-filter deny 20
  match ip address prefix-list host-route
route-map extcon-rmap-filter permit 1000
!
route-map extcon-rmap-filter-v6 deny 10
  match ipv6 address prefix-list default-route-v6
route-map extcon-rmap-filter-v6 deny 20
  match ip address prefix-list host-route-v6
route-map extcon-rmap-filter-v6 permit 1000
```

- Static default routes in the VRF.

```
vrf context myvrf_50001
  ip route 0.0.0.0/0 192.0.2.1
  ipv6 route 0::/0 2001:DB8::95/64
```

- Applying route maps to BGP neighbors.

```
router bgp 65002
  vrf myvrf_50001
    address-family ipv4 unicast
      network 0.0.0.0/0
    address-family ipv6 unicast
      network 0::/0
    neighbor 192.0.2.1
      remote-as 65099
      address-family ipv4 unicast
        route-map extcon-rmap-filter out
    neighbor 2001:DB8::1/64
      remote-as 65099
      address-family ipv4 unicast
        route-map extcon-rmap-filter-v6 out
```

VXLAN BGP EVPN border nodes enable effective control of route advertisements to external peers, improving security and network efficiency by allowing only designated routes.

External router configuration reference

External routers connect to the VXLAN BGP EVPN border node. These routers require specific configuration settings to ensure proper integration. The essential parameters are as follows:

- **VRF name**: The VRF name is local to each device. It does not need to match the VRF name used within the VXLAN BGP EVPN fabric, but should be used consistently throughout the configuration for clarity.
- **Address family support**: The router supports both IPv4 and IPv6 for dual-stack operation. You can configure either protocol as needed for your deployment.
- **Interface and neighbor settings**: Interfaces can be configured with dot1q encapsulation and assigned to the chosen VRF. Neighbor relationships are established for both IPv4 and IPv6 peers.

Example configuration:

```
vrf context myvrf_50001
!
router bgp 65099
  vrf myvrf_50001
    address-family ipv4 unicast
      maximum-paths 2
    address-family ipv6 unicast
      maximum-paths 2
    neighbor 192.0.2.31
      remote-as 65002
      address-family ipv4 unicast
    neighbor 2001:DB8::1/64
      remote-as 65002
      address-family ipv4 unicast
!
interface Ethernet1/3
  no switchport
  no shutdown
interface Ethernet1/3.2
  encapsulation dot1q 2
  vrf member myvrf_50001
  ip address 192.0.2.1/24
  Ipv6 address 2001:DB8::95/64
  no shutdown
```

VXLAN BGP EVPN with OSPF for VRF-lite

Configure VRF for VXLAN routing and external connectivity using OSPF

Use this task when integrating VXLAN routed networks with external OSPF domains using per-VRF BGP configuration on the border node.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Step 2 Configure BGP with the appropriate autonomous system number: **router bgp** *autonomous-system-number*

Step 3 Specify the required VRF instance: **vrf** *vrf-name*

Step 4 Configure the IPv4 address family: **address-family ipv4 unicast**

Step 5 Enable the advertisement of EVPN routes within the address family: **advertise l2vpn evpn**

Step 6 Enable equal-cost multipathing (ECMP) for iBGP prefixes: **maximum-paths ibgp** *number*

Step 7 Define redistribution from OSPF into BGP using a route map: **redistribute ospf** *name* **route-map** *name*

Step 8 Repeat steps 3 through 7 for each additional VRF peering you need to configure.

The border node is configured for VXLAN routing with external OSPF connectivity per VRF. Routing information is exchanged correctly between VXLAN and OSPF domains. For a configuration example, see VXLAN BGP EVPN configuration options for OSPF-based VRF-lite connectivity, on page 19.

What to do next

Validate connectivity between VXLAN and OSPF domains, and verify route propagation as needed in your network.

Configure a route-map for BGP to OSPF redistribution

Perform this task when integrating BGP with OSPF, especially in VXLAN BGP EVPN fabrics where iBGP route types must be matched.

Before you begin

- Ensure you have administrative access to the switch CLI.
- Verify that BGP and OSPF processes are already configured.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Step 2 Create a route-map to permit redistribution from BGP to OSPF: **route-map** *name* **permit 10**

Step 3 Match internal BGP route types, required if you use iBGP in your topology: **match route-type internal**

You can now redistribute BGP routes that match the specified criteria into OSPF. For a configuration example, see VXLAN BGP EVPN configuration options for OSPF-based VRF-lite connectivity, on page 19.

What to do next

- Verify OSPF has learned the redistributed routes from BGP.

Configure OSPF for per-VRF peering on the border node

Per-VRF peering allows multiple isolated VRFs to participate in OSPF routing on a border node, enabling selective redistribution of BGP routes into OSPF.

Before you begin

- Ensure that BGP is configured and operational on the border node.
- Verify you have administrative access to the device.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Step 2 Configure an OSPF instance: **router ospf** *instance*

Step 3 Specify the VRF for OSPF: **vrf** *vrf-name*

Step 4 Define redistribution from BGP to OSPF: **redistribute bgp** *autonomous-system-number* **route-map** *name*

Step 5 Repeat steps 3 and 4 for each additional VRF peering.

Apply the VRF configuration and the redistribution configuration to each VRF that requires OSPF peering.

OSPF is enabled for specified VRFs on the border node, and BGP routes are redistributed to OSPF as configured. For a configuration example, see VXLAN BGP EVPN configuration options for OSPF-based VRF-lite connectivity, on page 19.

What to do next

Verify OSPF neighbors are established for each VRF and confirm route redistribution is functioning as expected.

Configure sub-interface instances on the border node for per-VRF peering

Before you begin

- Ensure you have administrative access to the border node CLI.
- Gather required VRF names, interface types/IDs, VLAN IDs, IP addresses, and OSPF details.

Perform this task to configure sub-interface instances on the border node.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Step 2 Select the parent interface: **interface** *parent-type/id*

Step 3 Disable Layer-2 switching on the parent interface: **no switchport**

Step 4 Enable the parent interface: **no shutdown**

Step 5 Exit interface configuration mode: **exit**

Step 6 For each VRF peering, perform these steps:

a) Define the sub-interface: **interface** *parent-type/id.sub-interface-number*

b) Assign the VLAN ID to the sub-interface: **encapsulation dot1q** *vlan-id*

VLAN ID range: 2 to 4093.

c) Associate the sub-interface with the correct VRF: **vrf** *vrf-name*

d) Assign the IP address: **ip address** *ip-address*

e) Define OSPF network-type for sub-interface: **ip ospf network point-to-point**

f) Configure the OSPF instance: **ip router ospf** *process-name* **area** *area-id*

g) Enable the sub-interface: **no shutdown**

Repeat these sub-steps for each required VRF peering.

Each VRF has a dedicated sub-interface for peering, which has VLAN, IP addressing, and OSPF configuration. Dynamic routing for each VRF is now enabled on the border node. For a configuration example, see VXLAN BGP EVPN configuration options for OSPF-based VRF-lite connectivity, on page 19.

VXLAN BGP EVPN configuration options for OSPF-based VRF-lite connectivity

This topic provides key configuration elements for deploying VXLAN BGP EVPN with OSPF to enable external connectivity using VRF-lite. The configuration illustrates how the border node connects as a neighbor to an external router, highlights VRF and interface settings, and ensures L3VNI consistency across the fabric.

Configuration overview

Use this configuration with IPv4 OSPFv2 deployments. The VRF names may differ between internal and external routers; however, the L3VNI must remain consistent.

Configuration components

Component	Description
route-map extcon-rmap-BGP-to-OSPF	Permits internal route type redistribution from BGP to OSPF
route-map extcon-rmap-OSPF-to-BGP	Permits redistribution from OSPF to BGP
vrf context myvrf_50001	Defines the VRF context and maps to VNI 50001
address-family ipv4 unicast	Specifies IPv4 unicast routing for the VRF
route-target both auto evpn	Enables EVPN route-target for Layer 2 and Layer 3 service
vlan 2000 / vn-segment 50001	Maps VLAN to VXLAN segment and associates with VRF
interface Vlan2000	Configures SVI; assigns VRF and enables IP forwarding
interface nve1	Configures VXLAN Network Virtualization Edge to use BGP reachability and loopback source
router bgp 65002	Enables BGP routing for the VRF and redistributes OSPF routes
router ospf EXT	Establishes OSPF process for external routing and redistributes BGP routes
interface Ethernet1/3 & Ethernet1/3.2	Configures uplink interface, adds VRF, IP address, and OSPF parameters

Sample configuration

```
route-map extcon-rmap-BGP-to-OSPF permit 10
  match route-type internal
route-map extcon-rmap-OSPF-to-BGP permit 10
!
vrf context myvrf_50001
  vni 50001
  rd auto
  address-family ipv4 unicast
```

```
    route-target both auto
    route-target both auto evpn
!
vlan 2000
  vn-segment 50001
!
interface Vlan2000
  no shutdown
  mtu 9216
  vrf member myvrf_50001
  no ip redirects
  ip forward
!
interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback1
  member vni 50001 associate-vrf
!
router bgp 65002
  vrf myvrf_50001
    router-id 10.2.0.6
    address-family ipv4 unicast
      advertise l2vpn evpn
      maximum-paths ibgp 2
      maximum-paths 2
      redistribute ospf EXT route-map extcon-rmap-OSPF-to-BGP
!
router ospf EXT
  vrf myvrf_50001
    redistribute bgp 65002 route-map extcon-rmap-BGP-to-OSPF
!
interface Ethernet1/3
  no switchport
  no shutdown
interface Ethernet1/3.2
  encapsulation dot1q 2
  vrf member myvrf_50001
  ip address 192.0.2.31/24
  ip ospf network point-to-point
  ip router ospf EXT area 0.0.0.0
  no shutdown
```

Route leaking

Centralized VRF route-leaking for VXLAN BGP EVPN fabrics

A centralized VRF route-leaking mechanism is a routing function that:

- enables the import and export of prefixes between VRFs through comprehensive route-policy control,
- performs route-leaking locally at a specific network device (the leaking point), and
- advertises the leaked routes to remote VTEPs or external routers in VXLAN BGP EVPN fabrics.

The advantage of centralized VRF route-leaking is that only the VTEP acting as the leaking point requires this configuration and capability, while all other VTEPs in the network remain neutral to the function.

Recommendation: Centralized VRF route-leaking

- Import each prefix into each VRF to ensure full cross-VRF reachability.
- Enable the **feature bgp** command before using the **export vrf default** command.
- Note that a VTEP with a less specific local prefix in its VRF might not be able to reach a more specific prefix in a different VRF.
- Ensure VXLAN routing in hardware and packet reencapsulation at the VTEP are in place for centralized VRF route-leaking with BGP EVPN.
- Beginning with Cisco NX-OS Release 9.3(5), use asymmetric VNIs to support centralized VRF route-leaking. For more information, see About VXLAN EVPN with Downstream VNI.

Centralized VRF route-leaking - Shared Internet with custom VRF

Centralized VRF route-leaking

A centralized VRF route-leaking method is a Layer 3 segmentation technique that:

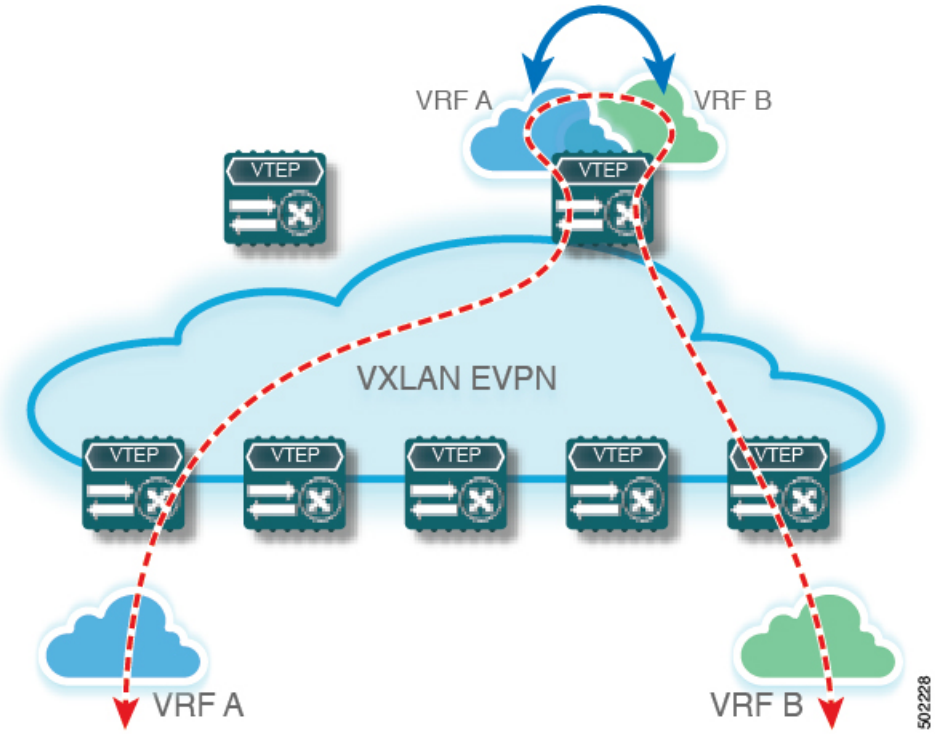
- enables the export and advertisement of select routes between isolated VRFs,
- prevents unintended route leaks (such as not leaking default routes from Blue and Red VRFs to the Shared Internet VRF), and
- optimizes reachability across VXLAN BGP EVPN fabrics through policy-based route control.

Shared Internet with VRF route-leaking for VXLAN BGP EVPN fabrics uses a border node to export the default route from the Shared Internet VRF. The border node selectively re-advertises this default route into custom VRFs, such as Blue and Red. The border node also advertises less specific prefixes (aggregates) to other VTEPs and destination VRFs as needed. BGP EVPN prevents routing loops by not re-exporting previously imported prefixes.

The default route is exported from the Shared Internet VRF and advertised into both VRF Blue and VRF Red on the border node. Aggregated prefixes for Blue or Red are advertised from the border node to the other VTEPs and the destination VRF.

The default route in VRF Blue or Red should not be leaked back to the Shared Internet VRF.

Figure 2: Centralized VRF route-leaking - Shared Internet with custom VRF



Configure Internet VRF on the border node

Use this procedure to configure IPv4 or IPv6 Internet VRF on a border node for VXLAN routing scenarios.

This procedure applies equally to IPv6.

Before you begin

- Ensure you have administrator access to the border node.
- Identify the VRF name and VNI number you want to assign.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Step 2 Configure the VRF context: **vrf context** *vrf-name*

Step 3 Specify the VNI for the VRF: **vni** *number*

The VNI assigned is known as the Layer 3 VNI (L3VNI), used as a common identifier across participating VTEPs.

Step 4 Configure the default route in the shared internet VRF to the external router: **ip route 0.0.0.0/0** *next-hop*

Step 5 Specify the route distinguisher (RD) for the VRF: **rd auto**

The RD uniquely identifies each VTEP for the L3VNI.

Step 6 Configure the IPv4 unicast address family: **address-family ipv4 unicast**

Step 7 Configure route targets (RTs) for import and export of IPv4 prefixes: **route-target both** {**auto** | *rt*}

Manually configured RTs are required for asymmetric VNIs. The supported formats include ASN2:NN, ASN4:NN, and IPV4:NN.

Step 8 Configure a special route target (RT) for the import and export of the shared IPv4 prefixes: **route-target both** *shared-vrf-rt* **evpn**

An additional import/export map for further qualification is supported.

The border node is configured with an Internet VRF and ready to route external traffic through VXLAN. For a configuration example, see Centralized VRF route-leaking configurations for shared Internet with custom VRF, on page 25.

What to do next

Verify connectivity and reachability through the new VRF as required.

Configure a custom VRF on the border node

Create a custom VRF instance on a border node and configure route filtering for IPv4 default routes.

Use this task to ensure that only specific routes are advertised when you use a border node. This method supports both IPv4 and IPv6.

Before you begin

- Ensure you have administrator privileges on the switch.
- Determine the names for your VRF, prefix list, and route map.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Step 2 Configure IPv4 prefix-list for default-route filtering: **ip prefix-list** *name* **seq 5 permit 0.0.0.0/0**

Step 3 Create a route map that begins with a deny statement to prevent leaking the default route: **route-map** *name* **deny 10**

Step 4 Match the IPv4 prefix-list that contains the default-route: **match ip address prefix-list** *name*

Step 5 Create a route-map with a trailing allow statement to advertise routes that do not match, using route leaking: **route-map** *name* **permit 20**

A custom VRF instance is created on the border node, with route filtering configured to prevent the default route from being advertised. Only intended routes are advertised, supporting both IPv4 and IPv6 as specified.

Configure a custom VRF context on the border node

Use this task when you want to create a custom VRF that supports both IPv4 and IPv6 on the border node of a VXLAN-enabled network.

Before you begin

- Ensure you have administrator access to the border node.
- Identify the VRF name, VNI, and route-map names.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Step 2 Configure the VRF context: **vrf context** *vrf-name*

Step 3 Specify the VNI for the VRF: **vni** *number*

The VNI assigned is known as the Layer 3 VNI (L3VNI), used as a common identifier across participating VTEPs.

Step 4 Specify the route distinguisher (RD) for the VRF: **rd auto**

The RD uniquely identifies each VTEP for the L3VNI.

Step 5 Configure a default route in the common VRF to direct traffic towards the border node with shared Internet VRF: **ip route 0.0.0.0/0 Null0**

Step 6 Configure the IPv4 unicast address family: **address-family ipv4 unicast**

Step 7 Configure route targets (RTs) for import and export of IPv4 prefixes: **route-target both** {**auto** | *rt*}

Manually configured RTs are required for asymmetric VNIs. The supported formats include ASN2:NN, ASN4:NN, and IPV4:NN.

Step 8 Configure a special route target (RT) for the import and export of the shared IPv4 prefixes: **route-target both** {**auto** | *rt*} **evpn**

An additional import/export map for further qualification is supported.

Step 9 Apply a route-map on routes being imported into this routing table: **import map** *name*

The border node is now configured with a custom VRF context, providing segmentation and routing for VXLAN traffic. For a configuration example, see Centralized VRF route-leaking configurations for shared Internet with custom VRF, on page 25.

Configure a custom VRF instance in BGP on the border node

Use this task to segment routing using a custom VRF in BGP on a border node. This setup enables separation of routing tables and advertisement of EVPN routes. The procedure applies equally to IPv6.

Before you begin

- Ensure you have administrative access to the switch.
- Gather the required target autonomous system number for BGP and VRF name.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Step 2 Configure BGP and specify the autonomous system number: **router bgp** *autonomous-system-number*

Step 3 Configure the VRF context: **vrf** *vrf-name*

Step 4 Configure the IPv4 unicast address family: **address-family ipv4 unicast**

Step 5 Enable the advertisement of EVPN routes within the IPv4 address family: **advertise l2vpn evpn**

Step 6 Add a default route network statement: **network 0.0.0.0/0**

Step 7 Enable equal-cost multipath (ECMP) for iBGP prefixes: **maximum-paths ibgp** *number*

Step 8 Enable ECMP for eBGP prefixes: **maximum-paths** *number*

The custom VRF is configured in BGP on the border node. Default route and ECMP are enabled, and EVPN routes are now advertised within the IPv4 address family. For a configuration example, see Centralized VRF route-leaking configurations for shared Internet with custom VRF, on page 25.

What to do next

Verify BGP and VRF operation. Confirm route propagation and EVPN route advertisement.

Centralized VRF route-leaking configurations for shared Internet with custom VRF

Centralized VRF route-leaking allows tenant VRFs, such as Blue and Red, to access the shared Internet through the Shared VRF. The configuration uses route-maps and prefix-lists to control which prefixes are exchanged between VRFs. This approach intentionally manages route propagation.

Key attributes

- The Shared VRF acts as the centralized internet exit point.
- Tenant VRFs (Blue and Red) import only permitted routes from the Shared VRF.
- Prefix-lists and route-maps prevent unwanted route leakage between VRFs.
- Route-targets are configured to facilitate EVPN-based route exchange.

Configuration example

This sample configuration demonstrates centralized VRF route-leaking with the Shared, Blue, and Red VRFs on a VXLAN BGP EVPN Border Node.

```
vrf context Shared
  vni 51099
  ip route 0.0.0.0/0 10.9.9.1
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
    route-target both 99:99
    route-target both 99:99 evpn
!
```

```
vlan 2199
  vn-segment 51099
!
interface Vlan2199
  no shutdown
  mtu 9216
  vrf member Shared
  no ip redirects
  ip forward
!
ip prefix-list PL_DENY_EXPORT seq 5 permit 0.0.0.0/0
!
route-map RM_DENY_IMPORT deny 10
 match ip address prefix-list PL_DENY_EXPORT
route-map RM_DENY_IMPORT permit 20
!
vrf context Blue
  vni 51010
  ip route 0.0.0.0/0 Null0
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
    route-target both 99:99
    route-target both 99:99 evpn
    import map RM_DENY_IMPORT
!
vlan 2110
  vn-segment 51010
!
interface Vlan2110
  no shutdown
  mtu 9216
  vrf member Blue
  no ip redirects
  ip forward
!
vrf context Red
  vni 51020
  ip route 0.0.0.0/0 Null0
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
    route-target both 99:99
    route-target both 99:99 evpn
    import map RM_DENY_IMPORT
!
vlan 2120
  vn-segment 51020
!
interface Vlan2120
  no shutdown
  mtu 9216
  vrf member Blue
  no ip redirects
  ip forward
!
interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback1
  member vni 51099 associate-vrf
  member vni 51010 associate-vrf
```

```
  member vni 51020 associate-vrf
!
router bgp 65002
  vrf Shared
    address-family ipv4 unicast
      advertise l2vpn evpn
      aggregate-address 10.10.0.0/16
      aggregate-address 10.20.0.0/16
      maximum-paths ibgp 2
      maximum-paths 2
  vrf Blue
    address-family ipv4 unicast
      advertise l2vpn evpn
      network 0.0.0.0/0
      maximum-paths ibgp 2
      maximum-paths 2
  vrf Red
    address-family ipv4 unicast
      advertise l2vpn evpn
      network 0.0.0.0/0
      maximum-paths ibgp 2
      maximum-paths
```

Result

This configuration enables tenant VRFs Blue and Red to access the shared Internet via the Shared VRF, while strictly managing the routes imported and exported between VRFs for security and policy compliance.

Centralized VRF route-leaking - Shared Internet with VRF default

Centralized VRF route leaks with Shared Internet and VRF default

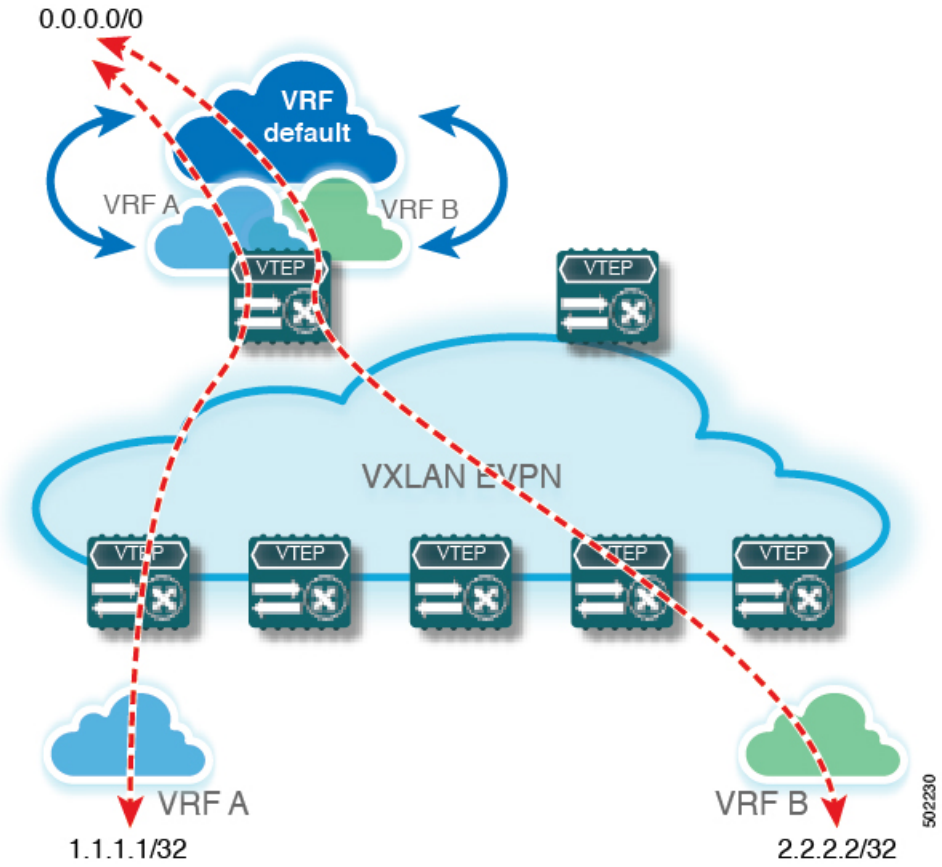
Centralized VRF route leaks are a routing mechanism that:

- enable controlled route sharing between different VRFs,
- allow specific prefixes, such as the default route, to be exported and re-advertised across VRFs (such as Blue and Red) on a Border Node, and
- prevent routing loops and undesired route propagation through export/import policy controls in BGP EVPN fabrics.

Shared Internet scenario with VRF default

In a shared Internet scenario with VRF default, centralized VRF route leaks allow selective advertisement of default and aggregate routes between VRFs. BGP EVPN mechanisms ensure that prefixes imported from other VRFs are not re-exported, mitigating the risk of routing loops. Properly configure less specific prefixes and export policies to ensure correct functionality.

Figure 3: Centralized VRF route-leaking – Shared Internet with VRF default



Configure VRF default on the border node

Use this procedure to set up a default route from VRF default to an external router. This is applicable for both IPv4 and IPv6 configurations.

Before you begin

Ensure you have administrative access to the border node and have identified the appropriate next-hop address for the default route.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Step 2 Configure the default route in the shared internet VRF to the external router: **ip route 0.0.0.0/0** *next-hop*

The border node directs all default VRF traffic to the specified external router, enabling external network connectivity. For a configuration example, see Centralized VRF route-leaking configuration options, on page 32.

What to do next

Verify routing by checking the routing table and testing network connectivity to external destinations.

Configure a BGP instance for VRF default on the border node

Use this procedure to configure BGP for the VRF default context on a border node. This configuration applies equally to both IPv4 and IPv6 networks.

Before you begin

Before you begin, ensure you are in privileged EXEC mode and have the required autonomous system number and network prefix details.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Step 2 Configure BGP and specify the autonomous system number: **router bgp** *autonomous-system-number*

Step 3 Configure the IPv4 unicast address family: **address-family ipv4 unicast**

Step 4 Create less specific prefix aggregate in VRF default: **aggregate-address** *prefix/mask*

Step 5 Enable ECMP for eBGP prefixes: **maximum-paths** *number*

BGP is configured for the VRF default context on the border node. The device can now aggregate prefixes and apply ECMP for eBGP routes. For a configuration example, see Centralized VRF route-leaking configuration options, on page 32.

Configure a custom VRF on the border node

Create a custom VRF instance on a border node and configure route filtering for IPv4 default routes.

Use this task to ensure that only specific routes are advertised when you use a border node. This method supports both IPv4 and IPv6.

Before you begin

- Ensure you have administrator privileges on the switch.
- Determine the names for your VRF, prefix list, and route map.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Step 2 Configure IPv4 prefix-list for default-route filtering: **ip prefix-list** *name* **seq 5 permit 0.0.0.0/0**

Step 3 Create a route map that begins with a deny statement to prevent leaking the default route: **route-map** *name* **deny 10**

Step 4 Match the IPv4 prefix-list that contains the default-route: **match ip address prefix-list** *name*

Step 5 Create a route-map with a trailing allow statement to advertise routes that do not match, using route leaking: **route-map** *name* **permit 20**

A custom VRF instance is created on the border node, with route filtering configured to prevent the default route from being advertised. Only intended routes are advertised, supporting both IPv4 and IPv6 as specified.

Configure a filter to permit prefixes from the default VRF on the border node

Use this task to control which prefixes are leaked from the default VRF to customer VRFs and remote VTEPs, including IPv6 routes if needed.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Step 2 Create a route-map with an allow statement: **route-map** *name* **permit 10**

Use this route-map to advertise permitted routes via route-leaking to the customer VRF and remote VTEPs.

The route-map settings permit only the desired prefixes from the default VRF. These prefixes are advertised to customer VRFs and remote VTEPs. For a configuration example, see Centralized VRF route-leaking configuration options, on page 32.

Configure a custom VRF context on the border node

Use this procedure to configure both IPv4 and IPv6 VRF contexts on a border node in a VXLAN BGP EVPN deployment.

Before you begin

- Ensure you have administrative CLI access to the border node.
- Collect the required VRF, VNI, and route-target values.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Step 2 Configure the VRF context: **vrf context** *vrf-name*

Step 3 Specify the VNI for the VRF: **vni** *number*

The VNI assigned is known as the Layer 3 VNI (L3VNI), used as a common identifier across participating VTEPs.

Step 4 Specify the route distinguisher (RD) for the VRF: **rd auto**

The RD uniquely identifies each VTEP for the L3VNI.

Step 5 (Optional) If the VRF does not have a default upstream route, add a default route to Null0: **ip route 0.0.0.0/0 Null0**

Step 6 Configure the IPv4 unicast address family: **address-family ipv4 unicast**

Step 7 Configure route targets for the EVPN instance for both import and export of prefixes.

a) Configure route targets (RTs) for import and export of IPv4 prefixes: **route-target both** {**auto** | *rt*}

Manually configured RTs are required for asymmetric VNIs. The supported formats include ASN2:NN, ASN4:NN, and IPV4:NN.

b) Configure RTs for EVPN import/export: **route-target both** {**auto** | *rt*} **evpn**

Configure manually if asymmetric VNI support is required.

c) Configure a special RT for importing IPv4 prefixes from the shared VRF: **route-target both** *shared-vrf-rt*

An additional import/export map for further qualification is supported.

d) Configure a special RT for importing IPv4 prefixes from the shared VRF for EVPN: **route-target both** *shared-vrf-rt* **evpn**

An additional import/export map for further qualification is supported.

Step 8 (Optional) Permit all routes from VRF default into the custom VRF using a route-map: **import vrf default map** *name*

The custom VRF context is now configured on the border node. It is ready for use in VXLAN/EVPN fabric operations. For a configuration example, see Centralized VRF route-leaking configuration options, on page 32.

What to do next

Verify connectivity and proper route-leaking between VRFs if required.

Configure a custom VRF instance in BGP on the border node

Use this task to segment routing using a custom VRF in BGP on a border node. This setup enables separation of routing tables and advertisement of EVPN routes. The procedure applies equally to IPv6.

Before you begin

- Ensure you have administrative access to the switch.
- Gather the required target autonomous system number for BGP and VRF name.

Procedure

Step 1 Enter global configuration mode: **configure terminal**

Step 2 Configure BGP and specify the autonomous system number: **router bgp** *autonomous-system-number*

Step 3 Configure the VRF context: **vrf** *vrf-name*

Step 4 Configure the IPv4 unicast address family: **address-family ipv4 unicast**

Step 5 Enable the advertisement of EVPN routes within the IPv4 address family: **advertise l2vpn evpn**

Step 6 Add a default route network statement: **network 0.0.0.0/0**

Step 7 Enable equal-cost multipath (ECMP) for iBGP prefixes: **maximum-paths ibgp** *number*

Step 8 Enable ECMP for eBGP prefixes: **maximum-paths** *number*

The custom VRF is configured in BGP on the border node. Default route and ECMP are enabled, and EVPN routes are now advertised within the IPv4 address family. For a configuration example, see Centralized VRF route-leaking configurations for shared Internet with custom VRF, on page 25.

What to do next

Verify BGP and VRF operation. Confirm route propagation and EVPN route advertisement.

Centralized VRF route-leaking configuration options

Centralized VRF route-leaking enables controlled communication between the default VRF and custom VRFs. It also supports the import and export of routes in a VXLAN BGP EVPN border node design.

Configuration options include:

- **Border node VRF definitions**: Set VNI; apply import/export route maps for VRF default and custom VRFs.
- **Prefix lists and route maps**: Permit or deny specific prefixes, including default route, between VRFs.
- **Interface and VLAN configuration**: Associate VLANs with VRFs and enable forwarding.
- **NVE interface configuration**: Enable VXLAN BGP EVPN and associate VNIs with VRFs.
- **BGP configuration**: Advertise aggregate and default routes as needed in each VRF.

Sample configuration

Use prefix lists and route maps to control which routes are exchanged between VRFs. This configuration provides examples of these options.

```
ip route 0.0.0.0/0 10.9.9.1
!
ip prefix-list PL_DENY_EXPORT seq 5 permit 0.0.0.0/0
!
route-map permit 10
match ip address prefix-list PL_DENY_EXPORT
route-map RM_DENY_EXPORT permit 20
route-map RM_PERMIT_IMPORT permit 10
!
vrf context Blue
  vni 51010
  ip route 0.0.0.0/0 Null0
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
    import vrf default map RM_PERMIT_IMPORT
    export vrf default 100 map RM_DENY_EXPORT allow-vpn
!
vlan 2110
  vn-segment 51010
!
interface Vlan2110
  no shutdown
  mtu 9216
  vrf member Blue
```

```
  no ip redirects
  ip forward
!
vrf context Red
  vni 51020
  ip route 0.0.0.0/0 Null0
  rd auto
  address-family ipv4 unicast
    route-target both auto
    route-target both auto evpn
    import vrf default map RM_PERMIT_IMPORT
    export vrf default 100 map RM_DENY_EXPORT allow-vpn
!
vlan 2120
  vn-segment 51020
!
interface Vlan2120
  no shutdown
  mtu 9216
  vrf member Blue
  no ip redirects
  ip forward
!
interface nve1
  no shutdown
  host-reachability protocol bgp
  source-interface loopback1
  member vni 51010 associate-vrf
  member vni 51020 associate-vrf
!
router bgp 65002
  address-family ipv4 unicast
      aggregate-address 10.10.0.0/16
      aggregate-address 10.20.0.0/16
      maximum-paths 2
      maximum-paths ibgp 2
  vrf Blue
    address-family ipv4 unicast
      advertise l2vpn evpn
      network 0.0.0.0/0
      maximum-paths ibgp 2
      maximum-paths 2
  vrf Red
    address-family ipv4 unicast
      advertise l2vpn evpn
      network 0.0.0.0/0
      maximum-paths ibgp 2
      maximum-paths 2
```