Configuring MPLS Layer 3 VPNs

This chapter describes how to configure Multiprotocol Label Switching (MPLS) Layer 3 Virtual Private Networks (VPNs) on Cisco Nexus 9508 switches.

- Information About MPLS Layer 3 VPNs, on page 1
- Prerequisites for MPLS Layer 3 VPNs, on page 5
- Guidelines and Limitations for MPLS Layer 3 VPNs, on page 5
- Default Settings for MPLS Layer 3 VPNs, on page 6
- Configuring MPLS Layer 3 VPNs, on page 7

Information About MPLS Layer 3 VPNs

An MPLS Layer 3 VPN consists of a set of sites that are interconnected by an MPLS provider core network. At each customer site, one or more customer edge (CE) routers or Layer 2 switches attach to one or more provider edge (PE) routers. This section includes the following topics:

- MPLS Layer 3 VPN Definition
- How an MPLS Layer 3 VPN Works
- Components of MPLS Layer 3 VPNs
- Hub-and-Spoke Topology
- OSPF Sham-Link Support for MPLS VPN

MPLS Layer 3 VPN Definition

MPLS-based Layer 3 VPNs are based on a peer model that enables the provider and the customer to exchange Layer 3 routing information. The provider relays the data between the customer sites without direct customer involvement.

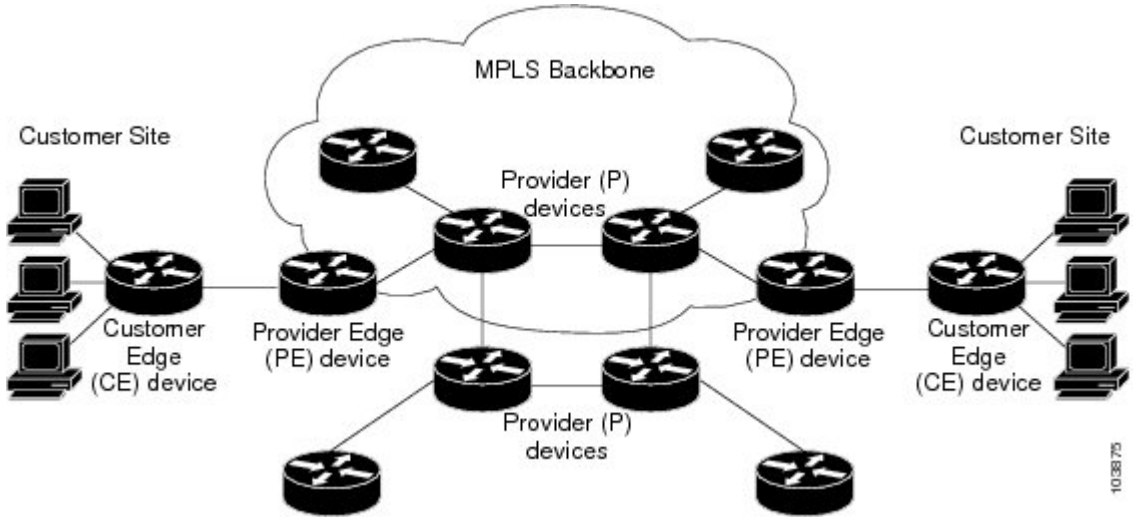
When you add a new site to an MPLS Layer 3 VPN, you must update the provider edge router that provides services to the customer site.

MPLS Layer 3 VPNs include the following components:

- Provider (P) router—A router in the core of the provider network. P routers run MPLS switching and do not attach VPN labels (an MPLS label in each route assigned by the PE router) to routed packets.

- Provider edge (PE) router—A router that attaches the VPN label to incoming packets that are based on the interface or subinterface on which they are received. A PE router attaches directly to a CE router.
- Customer edge (CE) router—An edge router on the network of the provider that connects to the PE router on the network. A CE router must interface with a PE router.

Figure 1: Basic MPLS Layer 3 VPN Terminology



How an MPLS Layer 3 VPN Works

MPLS Layer 3 VPN functionality is enabled at the edge of an MPLS network. The PE router performs the following tasks:

- Exchanges routing updates with the CE router
- Translates the CE routing information into VPN routes
- Exchanges Layer 3 VPN routes with other PE routers through the Multiprotocol Border Gateway Protocol (MP-BGP)

Components of MPLS Layer 3 VPNs

An MPLS-based Layer 3 VPN network has three components:

1. VPN route target communities—A VPN route target community is a list of all members of a Layer 3 VPN community. You must configure the VPN route targets for each Layer 3 VPN community member.
2. Multiprotocol BGP peering of VPN community PE routers—Multiprotocol BGP propagates VRF reachability information to all members of a VPN community. You must configure Multiprotocol BGP peering in all PE routers within a VPN community.
3. MPLS forwarding—MPLS transports all traffic between all VPN community members across a VPN enterprise or service provider network.

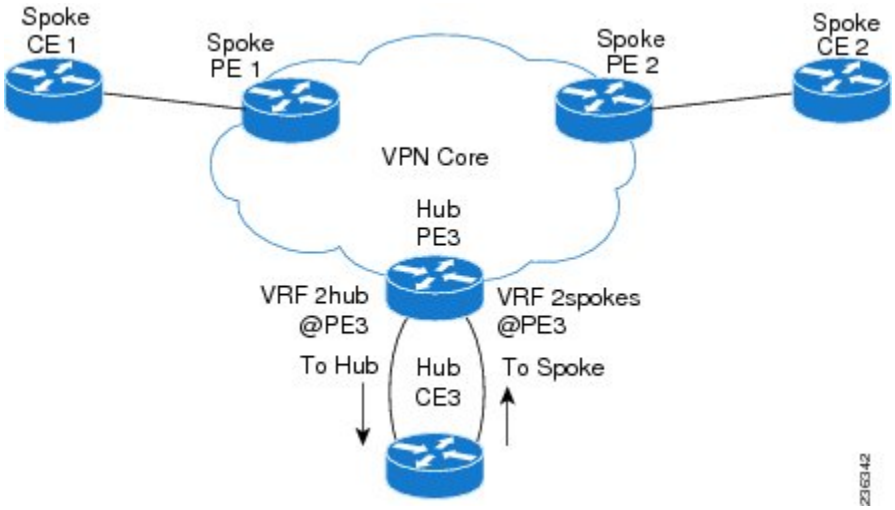
A one-to-one relationship does not necessarily exist between customer sites and VPNs. A site can be a member of multiple VPNs. However, a site can associate with only one VRF. A customer-site VRF contains all the routes that are available to the site from the VPNs of which it is a member.

Hub-and-Spoke Topology

A hub-and-spoke topology prevents local connectivity between subscribers at the spoke provider edge (PE) routers and ensures that a hub site provides subscriber connectivity. Any sites that connect to the same PE router must forward intersite traffic using the hub site. This topology ensures that the routing at the spoke sites moves from the access-side interface to the network-side interface or from the network-side interface to the access-side interface but never from the access-side interface to the access-side interface. A hub-and-spoke topology allows you to maintain access restrictions between sites.

A hub-and-spoke topology prevents situations where the PE router locally switches the spokes without passing the traffic through the hub site. This topology prevents subscribers from directly connecting to each other. A hub-and-spoke topology does not require one VRF for each spoke.

Figure 2: Hub-and-Spoke Topology



As shown in the figure, a hub-and-spoke topology is typically set up with a hub PE that is configured with two VRFs:

- VRF 2hub with a dedicated link connected to the hub customer edge (CE)
- VRF 2spokes with another dedicated link connected to the hub CE.

Interior Gateway Protocol (IGP) or external BGP (eBGP) sessions are usually set up through the hub PE-CE links. The VRF 2hub imports all the exported route targets from all the spoke PEs. The hub CE learns all routes from the spoke sites and readvertises them back to the VRF 2spoke of the hub PE. The VRF 2spoke exports all these routes to the spoke PEs.

If you use eBGP between the hub PE and hub CE, you must allow duplicate autonomous system (AS) numbers in the path which is normally prohibited. You can configure the router to allow this duplicate AS number at the neighbor of VRF 2spokes of the hub PE and also for VPN address family neighbors at all the spoke PEs. In addition, you must disable the peer AS number check at the hub CE when distributing routes to the neighbor at VRF 2spokes of the hub PE.

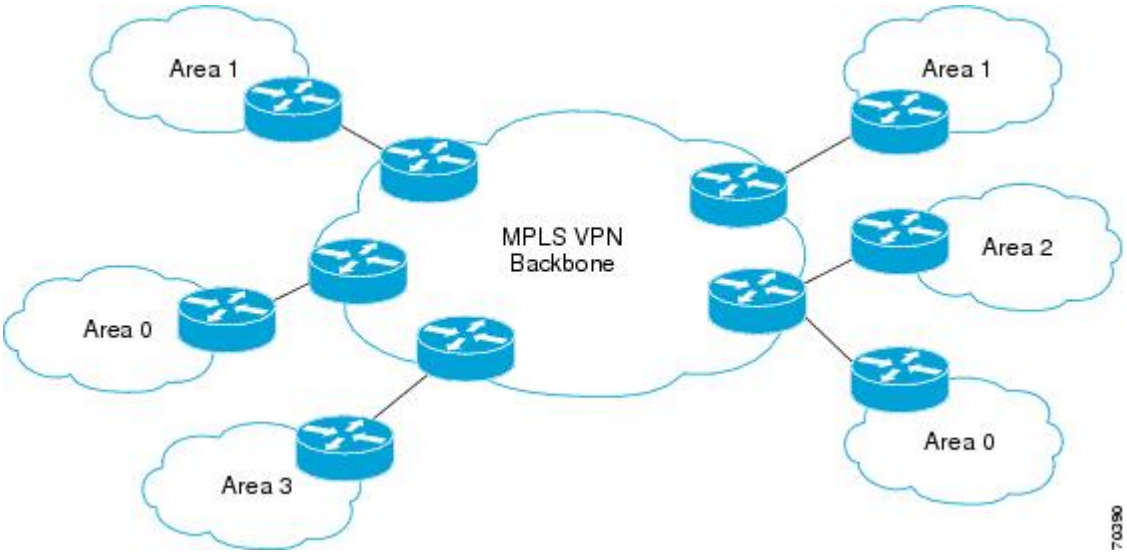
OSPF Sham-Link Support for MPLS VPN

In a Multiprotocol Label Switching (MPLS) VPN configuration, you can use the Open Shortest Path First (OSPF) protocol to connect customer edge (CE) devices to service provider edge (PE) devices in the VPN backbone. Many customers run OSPF as their intrasite routing protocol, subscribe to a VPN service, and want to exchange routing information between their sites using OSPF (during migration or on a permanent basis) over an MPLS VPN backbone.

The benefits of the OSPF sham-link support for MPLS VPN are as follows:

- Client site connection across the MPLS VPN Backbone—A sham link ensures that OSPF client sites that share a backdoor link can communicate over the MPLS VPN backbone and participate in VPN services.
- Flexible routing in an MPLS VPN configuration—In an MPLS VPN configuration, the OSPF cost that is configured with a sham link allows you to decide if OSPF client site traffic is routed over a backdoor link or through the VPN backbone.

The figure below shows an example of how VPN client sites that run OSPF can connect over an MPLS VPN backbone.



When you use OSPF to connect PE and CE devices, all routing information learned from a VPN site is placed in the VPN routing and forwarding (VRF) instance that is associated with the incoming interface. The PE devices that attach to the VPN use the Border Gateway Protocol (BGP) to distribute VPN routes to each other. A CE device can learn the routes to other sites in the VPN by peering with its attached PE device. The MPLS VPN super backbone provides an additional level of routing hierarchy to interconnect the VPN sites that are running OSPF.

When OSPF routes are propagated over the MPLS VPN backbone, additional information about the prefix in the form of BGP extended communities (route type, domain ID extended communities) is appended to the BGP update. This community information is used by the receiving PE device to decide the type of link-state advertisement (LSA) to be generated when the BGP route is redistributed to the OSPF PE-CE process. In this way, internal OSPF routes that belong to the same VPN and are advertised over the VPN backbone are seen as interarea routes on the remote sites.

Prerequisites for MPLS Layer 3 VPNs

MPLS Layer 3 VPNs has the following prerequisites:

- Ensure that you have configured MPLS and Label Distribution Protocol (LDP) in your network. All routers in the core, including the PE routers, must be able to support MPLS forwarding.
- Ensure that you have installed the correct license for MPLS and any other features you will be using with MPLS.

Guidelines and Limitations for MPLS Layer 3 VPNs

MPLS Layer 3 VPNs have the following configuration guidelines and limitations:

- You can configure MPLS Layer 3 VPN (LDP) on Cisco Nexus 3600-R and Cisco Nexus 9504 and 9508 platform switches with the N9K-X9636C-RX, N9K-X9636C-R, N9K-X96136YC-R, and N9K-X9636Q-R line cards.
- Ensure that MPLS IP forwarding is not enabled on the interface which terminates tunnel endpoint, as it is not supported.
- You must enable MPLS IP forwarding on interfaces where the forwarding decisions are made based on the labels of incoming packets. If a VPN label is allocated by per prefix mode, MPLS IP forwarding must be enabled on the link between PE and CE.
- Because of the hardware limitation on the trap resolution on Cisco Nexus 9508 platform switches with the N9K-X9636C-R and N9K-X9636Q-R line cards, uRPF may not be applied on supervisor bound packets via in-band.
- On Cisco Nexus 9500 platform switches with the -R series line cards, RACL is applied only to routed traffic so that the bridge traffic does not hit RACL. This applies to all Multicast OSPF control traffic.
- On Cisco Nexus 9500 platform switches with the -R series line cards, Control Packets with Explicit-NULL label is not prioritized when sending to SUP. This may result in control protocols flapping when explicit-NULL is configured.
- Per-label statistics at a scale of 500K is not supported on Cisco Nexus 9500 platform switches with the -R series line cards because of the hardware limitation.
- ARP scaling on Cisco Nexus 9500 platform switches with the -R series line cards is limited to 64K if all the 64K MACs are different. This limitation also applies if there are several Equal Cost Multiple Paths (ECMP) configured on the interface.
- Packets with MPLS Explicit-NULL may not be parsed correctly with default line card profile.
- MPLS Layer 3 VPNs support the following CE-PE routing protocols:
    - BGP (IPv4 and IPv6)
    - Enhanced Interior Gateway Protocol (EIGRP) (IPv4)
    - Open Shortest Path First (OSPFv2)
    - Routing Information Protocol (RIPv2)

- Set statements in an import route map are ignored.
- The BGP minimum route advertisement interval (MRAI) value for all iBGP and eBGP sessions is zero and is not configurable.
- In a high scale setup with many BGP routes getting redistributed into EIGRP, modify the EIGRP signal timer to ensure that the EIGRP convergence time is higher than the BGP convergence time. This process allows all the BGP routes to be redistributed into EIGRP, before EIGRP signals convergence.
- MPLS Layer 3 VPNs are supported on M3 Series modules.
- When OSPF is used as a protocol between PE and CE devices, the OSPF metric is preserved when routes are advertised over the VPN backbone. The metric is used on the remote PE devices to select the correct route. Do not modify the metric value when OSPF is redistributed to BGP and when BGP is redistributed to OSPF. If you modify the metric value, routing loops might occur.
- MPLS Traffic Engineering (RSVP) is not supported on Cisco Nexus 9508 platform switches with the N9K-X9636C-R and N9K-X9636Q-R line cards, .
- Beginning Cisco NX-OS Release 9.3(1), the behavior of the BGP pre-best path point of insertion (POI) is changed. In this release, the NX-OS RPM, BGP, and HMM software use a single cost community ID (either 128 for internal routes or 129 for external routes) to identify a BGP VPNv4 route as an EIGRP originated route. Only the routes that have the pre-best path value set to cost community ID 128 or 129 are installed in the URIB along with the cost extcommunity. Any non-EIGRP originated route carrying the above described cost community ID would be installed in URIB along with pre-best path cost community. As a result, URIB would use this cost to identify the better route between the route learnt via the iBGP and backdoor-EIGRP instead of the admin distance.

  Only the routes that have the pre-best path value set to cost community ID 128 or 129 are installed in the URIB along with the cost extcommunity.
- The Egress RACL (e-RACL) TCAM and MPLS Extended ECMP features are mutually exclusive. To enable MPLS Extended ECMP (**hardware profile mpls extended-ecmp)** on the Cisco Nexus N9K-X9636C-RX line card, set the e-RACL TCAM carving to 0.

Default Settings for MPLS Layer 3 VPNs

Table 1: Default MPLS Layer 3 VPN Parameters

Parameters	Default
L3VPN feature	Disabled
L3VPN SNMP notifications	Disabled
allowas-in (for a hub-and-spoke topology)	0
disable-peer-as-check (for a hub-and-spoke topology)	Disabled

Configuring MPLS Layer 3 VPNs

About OSPF Domain IDs and Tags

You can set the domain_ID for an OSPF router instance within a VRF. In OSPF, Cisco NX-OS uses the domain_ID and domain tag to control aspects of BGP route redistribution at the provider edge (PE) or customer edge (CE).

- You can configure a primary and secondary domain_ID for the redistributed OSPF routes.
- OSPF also uses a domain tag to identify the OSPF process ID.

The Cisco NX-OS implementation of domain IDs and domain tags complies with RFC 4577.

Note The OSPF primary and secondary domain_IDs and the domain tag are available only when MPLS L3VPN feature is enabled.

Configuring OSPF at the PE and CE Boundary

By using, domain IDs and domain tags, you can configure NX-OS to redistribute OSPF routes into BGP networks, and receive BGP redistributed routes into OSPF at the PE and CE boundary. See the following topics:

- About OSPF Domain IDs and Tags, on page 7
- Configuring the OSPF Domain ID, on page 8
- Configuring the Secondary Domain ID, on page 9
- Configuring the OSPF Domain Tag, on page 7

Configuring the OSPF Domain Tag

The domain tag specifies the OSPF process instance number that NX-OS redistributes into BGP at the PE or CE.

Before you begin

Make sure that MPLS and OSPFv2 are enabled.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters the configuration terminal.

	Command or Action	Purpose

	`switch-1# configure terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch-1(config)#`	

Step 2	**router ospf** *process-tag*
	Example:
	`switch-1(config)# router ospf 101`
	`switch-1(config-router)#`
		Enters router configuration mode to configure the OSPF router instance. The process tag is an alphanumeric string from 1 through 20 characters that identifies the router.

Step 3	**vrf** *vrf-name*
	Example:
	`switch-1(config-router)# vrf pubstest`
	`switch-1(config-router-vrf)#`
		Enter the specific VRF instance for OSPF. The VRF name is an alphanumeric string from 1 through 32 characters that identifies the VRF.

Step 4	**ospf domain-tag** *as-number*
	Example:
	`switch-1(config-router-vrf)# domain-tag 9999`
	`nxosv2(config-router-vrf)#`
		Sets the domain tag. The domain tag is an alphanumeric string from 0 through 2147483647 that identifies the AS number.

Configuring the OSPF Domain ID

You can set the domain_ID for an OSPF router instance within a VRF to control BGP route redistribution into OSPF at the CE or PE.

To remove this feature, use the **no domain-id** command.

Before you begin

Both the MPLS L3VPN and OSPFv2 feature must be enabled to use the OSPF domain_ID feature.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch-1# configure terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch-1(config)#`
		Enters the configuration terminal.

Step 2	**router ospf** *process-tag*
	Example:
	`switch-1(config)# router ospf 101`
	`switch-1(config-router)#`
		Enters router configuration mode to configure the OSPF router instance. The process tag is an alphanumeric string from 1 through 20 characters that identifies the router.

	Command or Action	Purpose

Step 3	**vrf** *vrf-name*
	Example:
	`switch-1(config-router)# vrf pubstest`
	`switch-1(config-router-vrf)#`
		Enter the specific VRF instance for OSPF. The VRF name is an alphanumeric string from 1 through 32 characters that identifies the VRF.

Step 4	**domain-id** { *id* | **type** *domain-type* **value** *value* | **Null** } **Example:** `switch-1(config-router-vrf)# domain-id 19.0.2.0`	Sets the domain_ID and additional parameters: • *id* specifies the domain ID in dotted decimal notation, for example, 1.2.3.4 • *type* specifies the domain type in four-byte notation, for example, 0005. • *value* specifies the domain value in 6 bytes of hexadecimal notation, for example, 0x0005. You can use the Null argument to clear the domain_ID.

Configuring the Secondary Domain ID

You can set a secondary domain_ID for an OSPF router instance within a VRF to control BGP route redistribution into OSPF at the CE or PE.

Use the **domain-id Null** command to unconfigure the domain_ID.

Before you begin

Make sure that OSPFv2 and MPLS features are enabled.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch-1# configure terminal`
	`Enter configuration commands, one per line. End with CNTL/Z.`
	`switch-1(config)#`
		Enters the configuration terminal.

Step 2	**router ospf** *process-tag*
	Example:
	`switch-1(config)# router ospf 101`
	`switch-1(config-router)#`
		Enters router configuration mode to configure the OSPF router instance. The process tag is an alphanumeric string from 1 through 20 characters that identifies the router.

	Command or Action	Purpose

Step 3	**vrf** *vrf-name*
	Example:
	`switch-1(config-router)# vrf pubstest`
	`switch-1(config-router-vrf)#`
		Enters the specific VRF instance for OSPF. The VRF name is an alphanumeric string from 1 through 32 characters that identifies the VRF.

Step 4	**domain-id** { *id* | **type** *domain-type* **value** *value* | **Null** } **Example:** `switch-1(config-router-vrf)# domain-id 19.0.2.0`	Sets the domain_ID for the autonomous system.

Configuring the Core Network

Assessing the Needs of MPLS Layer 3 VPN Customers

You can identify the core network topology so that it can best serve MPLS Layer 3 VPN customers.

- Identify the size of the network:
  - Identify the following to determine the number of routers and ports you need:
  - How many customers do you need to support?
  - How many VPNs are needed per customer?
  - How many virtual routing and forwarding instances are there for each VPN?
- Determine which routing protocols you need in the core network.
- Determine if you need MPLS VPN high availability support.

Note MPLS VPN nonstop forwarding and graceful restart are supported on select routers and Cisco NX-OS releases. You need to make sure that graceful restart for BGP and LDP is enabled.

- Configure the routing protocols in the core network.
- Determine if you need BGP load sharing and redundant paths in the MPLS Layer 3 VPN core.

Configuring MPLS in the Core

To enable MPLS on all routers in the core, you must configure a label distribution protocol. You can use either of the following as a label distribution protocol:

- MPLS Label Distribution Protocol (LDP).
- MPLS Traffic Engineering Resource Reservation Protocol (RSVP).

Configuring Multiprotocol BGP on the PE Routers and Route Reflectors

You can configure multiprotocol BGP connectivity on the PE routers and route reflectors.

Before you begin

- Ensure that graceful restart is enabled on all routers for BGP and LDP.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enables the BGP feature.

Step 3	**install feature-set mpls**
	Example:
	`switch(config)# install feature-set mpls`
	`switch(config)#`
		Installs the MPLS feature-set.

Step 4	**feature-set mpls**
	Example:
	`switch(config)# feature-set mpls`
	`switch(config)#`
		Enables the MPLS feature-set.

Step 5	**feature mpls l3vpn**
	Example:
	`switch(config)# feature mpls l3vpn`
	`switch(config)#`
		Enables the MPLS Layer 3 VPN feature.

Step 6	**router bgp** *as - number* **Example:** `switch(config)# router bgp 1.1`	Configures a BGP routing process and enters router configuration mode. The as-number argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
Step 7	**router-id** *ip-address* **Example:**	(Optional) Configures the BGP router ID. This IP address identifies this BGP speaker. This command triggers an automatic notification

	Command or Action	Purpose
	`switch(config-router)# router-id 192.0.2.255`	and session reset for the BGP neighbor sessions.

Step 8	**neighbor** *ip-address* **remote-as** *as-number*
	Example:
	`switch(config-router)# neighbor 209.165.201.1 remote-as 1.1`
	`switch(config-router-neighbor)#`
		Adds an entry to the iBGP neighbor table. The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.

Step 9	**address-family { vpnv4 | vpnv6 } unicast**
	Example:
	`switch(config-router-neighbor)# address-family vpnv4 unicast`
	`switch(config-router-neighbor-af)#`
		Enters address family configuration mode for configuring routing sessions, such as BGP, that uses standard VPNv4 or VPNv6 address prefixes.

Step 10	**send-community extended** **Example:** `switch(config-router-neighbor-af)# send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor.
Step 11	**show bgp { vpnv4 | vpnv6 } unicast neighbors** **Example:** `switch(config-router-neighbor-af)# show bgp vpnv4 unicast neighbors`	(Optional) Displays information about BGP neighbors.
Step 12	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Connecting the MPLS VPN Customers

Defining VRFs on the PE Routers to Enable Customer Connectivity

You must create VRFs on the PE routers to enable customer connectivity. You configure route targets to control which IP prefixes are imported into the customer VPN site and which IP prefixes are exported to the BGP network. You can optionally use an import or export route map to provide more fine-grained control over the IP prefixes that are imported into the customer VPN site or exported out of the VPN site. You can use a route map to filter routes that are eligible for import or export in a VRF, based on the route target extended community attributes of the route. The route map might, for example, deny access to selected routes from a community that is on the import route target list.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**install feature-set mpls**
	Example:
	`switch(config)# install feature-set mpls`
	`switch(config)#`
		Installs the MPLS feature-set.

Step 3	**feature-set mpls**
	Example:
	`switch(config)# feature-set mpls`
	`switch(config)#`
		Enables the MPLS feature-set.

Step 4	**feature-set mpls l3vpn**
	Example:
	`switch(config)# feature-set mpls l3vpn`
	`switch(config)#`
		Enables the MPLS Layer 3 VPN feature.

Step 5	**vrf context** *vrf-name*
	Example:
	`switch(config)# vrf context vpn1`
	`switch(config-vrf)#`
		Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode. The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 6	**rd** *route-distinguisher*
	Example:
	`switch(config-vrf)# rd 1.2:1`
	`switch(config-vrf)#`
		Configures the route distinguisher. The route-distinguisher argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter an RD in either of these formats:
		• 16-bit or 32-bit AS number: your 32-bit number, for example, 1.2:3
		• 32-bit IP address: your 16-bit number, for example, 192.0.2.1:1

Step 7	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af-ipv4)#`
		Specifies the IPv4 address family type and enters address family configuration mode.

	Command or Action	Purpose

Step 8	**route-target** { **import** | **export** } *route-target-ext-community* **}**
	Example:
	`switch(config-vrf-af-ipv4)# route-target import 1.0:1`
		Specifies a route-target extended community for a VRF as follows:
		• The import keyword imports routing information from the target VPN extended community.
		• The export keyword exports routing information to the target VPN extended community.
		• The route-target-ext-community argument adds the route-target extended community attributes to the VRF's list of import or export route-target extended communities. You can enter the route-target-ext-community argument in either of these formats:
		  • 16-bit or 32-bit AS number: your 32-bit number, for example, 1.2:3
		  • 32-bit IP address: your 16-bit number, for example, 192.0.2.1:1

Step 9	**maximum routes** *max-routes* [**threshold** *value*] [**reinstall**] **Example:** `switch(config-vrf-af-ipv4)# maximum routes 10000`	(Optional) Configures the maximum number of routes that can be stored in the VRF route table. The max-routes range is from 1 to 4294967295. The threshold value range is from 1 to 100.
Step 10	**import** [**vrf default** *max-prefix*] **map** *route-map* **Example:** `switch(config-vrf-af-ipv4)# import vrf default map vpn1-route-map`	(Optional) Configures an import policy for a VRF to import prefixes from the default VRF as follows: • The max-prefix range is from 1 to 2147483647. The default is 1000 prefixes. • The route-map argument specifies the route map to be used as an import route map for the VRF and can be any case-sensitive, alphanumeric string up to 63 characters.
Step 11	**show vrf** *vrf-name* **Example:** `switch(config-vrf-af-ipv4)# show vrf vpn1`	(Optional) Displays information about a VRF. The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.

	Command or Action	Purpose
Step 12	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring VRF Interfaces on PE Routers for Each VPN Customer

You can associate a virtual routing and forwarding instance (VRF) with an interface or subinterface on the PE routers.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface** *type number*
	Example:
	`switch(config)# interface Ethernet 5/0`
	`switch(config-if)#`
		Specifies the interface to configure and enters interface configuration mode as follows:
		• The type argument specifies the type of interface to be configured.
		• The number argument specifies the port, connector, or interface card number.

Step 3	**vrf member** *vrf-name* **Example:** `switch(config-if)# vrf member vpn1`	Associates a VRF with the specified interface or subinterface. The vrf-name argument is the name assigned to a VRF.
Step 4	**show vrf** *vrf-name* **interface** **Example:** `switch(config-if)# show vrf vpn1 interface`	(Optional) Displays information about interfaces associated with a VRF. The vrf-name argument is any case-sensitive alphanumeric string up to 32 characters.
Step 5	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring Routing Protocols Between the PE and CE Routers

Configuring Static or Directly Connected Routes Between the PE and CE Routers

You can configure the PE router for PE-to-CE routing sessions that use static routes.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**vrf context** *vrf-name*
	Example:
	`switch(config)# vrf context vpn1`
	`switch(config-vrf)#`
		Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode. The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 3	{ **ip ipv6** } **route** *prefix nexthop* **Example:** `switch(config-vrf)# ip route 192.0.2.1/28 ethernet 2/1`	Defines static route parameters for every PE-to-CE session. The prefix and nexthop are as follows: • IPv4—in dotted decimal notation • IPv6—in hex format.

Step 4	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af)#`
		Specifies the IPv4 address family type and enters address family configuration mode.

Step 5	**feature bgp** *as - number*
	Example:
	`switch(config-vrf-af)# feature bgp`
	`switch(config)#`
		Enables the BGP feature.

Step 6	**router bgp** *as - number* **Example:** `switch(config)# router bgp 1.1`	Configures a BGP routing process and enters router configuration mode. The as-number argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 7	**vrf** *vrf-name*
	Example:
	`switch(config-router)# vrf vpn1`
	`switch(config--router-vrf)#`
		Associates the BGP process with a VRF.
		The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.

	Command or Action	Purpose

Step 8	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af)#`
		Specifies the IPv4 address family type and enters address family configuration mode.

Step 9	**redistribute static route-map** *map-name* **Example:** `switch(config-router-vrf-af)# redistribute static route-map StaticMap`	Redistributes static routes into BGP. The map-name can be any case-sensitive, alphanumeric string up to 63 characters.
Step 10	**redistribute direct route-map** *map-name* **Example:** `switch(config-router-vrf-af)# redistribute direct route-map StaticMap`	Redistributes directly connected routes into BGP. The map-name can be any case-sensitive, alphanumeric string up to 63 characters.
Step 11	**show { ipv4 | ipv6 } route vrf** *vrf-name* **Example:** `switch(config-router-vrf-af)# show ip ipv4 route vrf vpn1`	(Optional) Displays information about routes. The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.
Step 12	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring BGP as the Routing Protocol Between the PE and CE Routers

You can use eBGP to configure the PE router for PE-to-CE routing sessions.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enables the BGP feature.

	Command or Action	Purpose

Step 3	**router bgp** *as - number*
	Example:
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		Configures a BGP routing process and enters router configuration mode.
		The as-number argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 4	**vrf** *vrf-name*
	Example:
	`switch(config-router)# vrf vpn1`
	`switch(config--router-vrf)#`
		Associates the BGP process with a VRF.
		The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 5	**neighbor** *ip-address***remote-as** *as-number*
	Example:
	`switch(config-router)# neighbor 209.165.201.1 remote-as 1.1`
	`switch(config-router-neighbor)#`
		Adds an entry to the iBGP neighbor table. The ip-address argument specifies the IP address of the neighbor in dotted decimal notation. The as-number argument specifies the autonomous system to which the neighbor belongs.

Step 6	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af)#`
		Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard IPv4 or IPv6 address prefixes.

Step 7	**show bgp { vpnv4 | vpnv6 } unicast neighbors vrf** *vrf-name* **Example:** `switch(config-router-neighbor-af)# show bgp vpnv4 unicast neighbors`	(Optional) Displays information about BGP neighbors. The vrf-name argument is any case-sensitive alphanumeric string up to 32 characters.
Step 8	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring RIPv2 Between the PE and CE Routers

You can use RIP to configure the PE router for PE-to-CE routing sessions.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature rip**
	Example:
	`switch(config)# feature rip`
	`switch(config)#`
		Enables the RIP feature.

Step 3	**router rip** *instance-tag* **Example:** `switch(config)# router rip Test1`	Enables RIP and enters router configuration mode. The instance-tag can be any case-sensitive, alphanumeric string up to 20 characters.

Step 4	**vrf** *vrf-name*
	Example:
	`switch(config-router)# vrf vpn1`
	`switch(config--router-vrf)#`
		Associates the RIP process with a VRF.
		The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 5	**address-family ipv4 unicast**
	Example:
	`switch(config-router-vrf)# address-family ipv4 unicast`
	`switch(config-router-vrf-af)#`
		Specifies the address family type and enters address family configuration mode.

Step 6	**redistribute** { **bgp** as | **direct** | { **egrip** | **ospf** | **rip** } *instance-tag* | static } **route-map** *map-name vrf-name* **Example:** `switch(config-router-vrf-af)# show ip rip vrf vpn1`	Redistributes routes from one routing domain into another routing domain. The as number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. The instance-tag can be any case-sensitive alphanumeric string up to 20 characters.
Step 7	**show ip rip vrf** *vrf-name* **Example:** `switch(config-router-vrf-af)# show ip rip vrf vpn1`	(Optional) Displays information about RIP. The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.
Step 8	**copy running-config startup-config** **Example:**	(Optional) Copies the running configuration to the startup configuration.

	Command or Action	Purpose
	`switch(config-router-vrf)# copy running-config startup-config`	

Configuring OSPF Between the PE and CE Routers

You can use OSPFv2 to configure the PE router for PE-to-CE routing sessions. You can optionally create an OSPF sham link if you have OSPF back door links that are not part of the MPLS network.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature ospf**
	Example:
	`switch(config)# feature ospf`
	`switch(config)#`
		Enables the OSPF feature.

Step 3	**router ospf** *instance-tag* **Example:** `switch(config)# router ospf Test1`	Enables OSPF and enters router configuration mode. The instance-tag can be any case-sensitive, alphanumeric string up to 20 characters.

Step 4	**vrf** *vrf-name*
	Example:
	`switch(config-router)# vrf vpn1`
	`switch(config--router-vrf)#`
		Enters router VRF configuration mode.
		The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 5	**area** area-id **sham-link** *source-address destination-address* **Example:** `switch(config-router-vrf)# area 1 sham-link 10.2.1.1 10.2.1.2`	(Optional) Configures the sham link on the PE interface within a specified OSPF area and with the loopback interfaces specified by the IP addresses as endpoints. You must configure the sham link at both PE endpoints.

Step 6	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-vrf-af)#`
		Specifies the address family type and enters address family configuration mode.

	Command or Action	Purpose
Step 7	**redistribute** { **bgp** as | **direct** | { **egrip** | **ospf** | **rip** } *instance-tag* | static } **route-map** *map-name* **Example:** `switch(config-router-vrf-af)# redistribute bgp 1.0 route-map BGPMap`	Redistributes BGP into the EIGRP. The autonomous system number of the BGP network is configured in this step. BGP must be redistributed into EIGRP for the CE site to accept the BGP routes that carry the EIGRP information. A metric must also be specified for the BGP network. The map-name can be any case-sensitive, alphanumeric string up to 63 characters.

Step 8	**autonomous-system** *as-number*
	Example:
	`switch(config-router-vrf-af)#`
	`autonomous-system 1.3`
		(Optional) Specifies the autonomous system number for this address family for the customer site.
		The as-number argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 9	**show ip egrip vrf** *vrf-name* **Example:** `switch(config-router-vrf-af)# show ipv4 eigrp vrf vpn1`	(Optional) Displays information about EIGRP in this VRF. The vrf-name can be any case-sensitive, alphanumeric string up to 32 characters
Step 10	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring EIGRP Between the PE and CE Routers

You can configure the PE router to use Enhanced Interior Gateway Routing Protocol (EIGRP) between the PE and CE routers to transparently connect EIGRP customer networks through an MPLS-enabled BGP core network so that EIGRP routes are redistributed through the VPN across the BGP network as internal BGP (iBGP) routes.

Before you begin

You must configure BGP in the network core.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature eigrp**
	Example:
	`switch(config)# feature eigrp`
	`switch(config)#`
		Enables the EIGRP feature.

Step 3	**router eigrp** *instance-tag* **Example:** `switch(config)# router eigrp Test1`	Configures an EIGRP instance and enters router configuration mode. The instance-tag can be any case-sensitive, alphanumeric string up to 20 characters.

Step 4	**vrf** *vrf-name*
	Example:
	`switch(config-router)# vrf vpn1`
	`switch(config-router-vrf)#`
		Enters router VRF configuration mode.
		The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 5	**address-family ipv4 unicast**
	Example:
	`switch(config-router-vrf)# address-family ipv4 unicast`
	`switch(config-router-vrf-af)#`
		(Optional) Enters address family configuration mode for configuring routing sessions that use standard IPv4 address prefixes.

Step 6	**redistribute bgp** *as-number* **route-map** *map-name* **Example:** `switch(config-router-vrf-af)# redistribute bgp 235354 route-map mtest1`	Redistributes routes from one routing domain into another routing domain. The *as number* can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. The instance-tag can be any case-sensitive alphanumeric string up to 20 characters
Step 7	**show ip ospf** *instance-tag* **vrf** *vrf-name* **Example:** `switch(config-router-vrf-af)# show ip rip vrf vpn1`	(Optional) Displays information about OSPF.
Step 8	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring PE-CE Redistribution in BGP for the MPLS VPN

You must configure BGP to distribute the PE-CE routing protocol on every PE router that provides MPLS Layer 3 VPN services if the PE-CE protocol is not BGP.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enables the BGP feature.

Step 3	**router bgp** *instance-tag*
	Example:
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		Configures a BGP routing process and enters router configuration mode. The as-number argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 4	**router id** *ip-address*
	Example:
	`switch(config-router)# router-id 192.0.2.255 1`
	`switch(config-router)#`
		(Optional) Configures the BGP router ID. This IP address identifies this BGP speaker. This command triggers an automatic notification and session reset for the BGP neighbor sessions.

Step 5	**router id** *ip-address* **remote-as** *as-number*
	Example:
	`switch(config-router)# neighbor 209.165.201.1 remote-as 1.2`
	`switch(config-router-neighbor)#`
		Adds an entry to the BGP or multiprotocol BGP neighbor table. The ip-address argument specifies the IP address of the neighbor in dotted decimal notation. The as-number argument specifies the autonomous system to which the neighbor belongs.

Step 6	**update-source loopback [0 | 1]** **Example:** `switch(config-router-neighbor)# update-source loopback 0#`	Specifies the source address of the BGP session.
Step 7	**address-family { ipv4 | ipv6 } unicast** **Example:**	Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 or VPNv6 address

	Command or Action	Purpose
	`switch(config-router-neighbor)# address-family vpnv4` `switch(config-router-neighbor-af)#`	prefixes. The optional unicast keyword specifies VPNv4 or VPNv6 unicast address prefixes.
Step 8	**send-community extended** **Example:** `switch(config-router-neighbor-af)# send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor.

Step 9	**vrf** *vrf-name*
	Example:
	`switch(config-router-neighbor-af)# vrf vpn1`
	`switch(config-router-vrf)#`
		Enters router VRF configuration mode.
		The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 10	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-router-vrf)# address-family ipv4 unicast`
	`switch(config-router-vrf-af)#`
		Enters address family configuration mode for configuring routing sessions that use standard IPv4 or IPv6 address prefixes.

Step 11	**redistribute** { **direct** | { **egrip** | **ospfv3** | **ospfv3** |**rip** } *instance-tag* | static } **route-map** *map-name* **Example:** `switch(config-router-af-vrf)# redistribute eigrp Test2 route-map EigrpMap`	Redistributes routes from one routing domain into another routing domain. The as number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. The instance-tag can be any case-sensitive, alphanumeric string up to 20 characters. The map-name can be any case-sensitive alphanumeric string up to 63 characters.
Step 12	**show bgp { ipv4 | ipv6 } unicast vrf** *vrf-name* **Example:** `switch(config-router--vrf-af)# show bgp ipv4 unicast vrf vpn1vpn1`	(Optional) Displays information about BGP. The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters.
Step 13	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring a Hub-and-Spoke Topology

Configuring VRFs on the Hub PE Router

You can configure hub and spoke VRFs on the hub PE router.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**install feature-set mpls**
	Example:
	`switch(config)# install feature-set mpls`
	`switch(config)#`
		Installs the MPLS feature-set.

Step 3	**feature-set mpls**
	Example:
	`switch(config)# feature-set mpls`
	`switch(config)#`
		Enables the MPLS feature-set.

Step 4	**feature-set mpls l3vpn**
	Example:
	`switch(config)# feature-set mpls l3vpn`
	`switch(config)#`
		Enables the MPLS Layer 3 VPN feature.

Step 5	**vrf context** *vrf-hub*
	Example:
	`switch(config)# vrf context 2hub`
	`switch(config-vrf)#`
		Defines the VPN routing instance for the PE hub by assigning a VRF name and enters VRF configuration mode. The vrf-hub argument is any case-sensitive alphanumeric string up to 32 characters.

Step 6	**rd** *route-distinguisher*
	Example:
	`switch(config-vrf)# rd 1.2:1`
	`switch(config-vrf)#`
		Configures the route distinguisher. The route-distinguisher argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter an RD in either of these formats:
		• 16-bit or 32-bit AS number: your 32-bit number, for example, 1.2:3
		• 32-bit IP address: your 16-bit number, for example, 192.0.2.1:1

Step 7	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af-ipv4)#`
		Specifies the IPv4 address family type and enters address family configuration mode.

	Command or Action	Purpose

Step 8	**route-target** { **import** | **export** } route-target-ext-community **}**
	Example:
	`switch(config-vrf-af-ipv4)# route-target import 1.0:1`
		Specifies a route-target extended community for a VRF as follows:
		• The **import** keyword imports routing information from the target VPN extended community.
		• The **export** keyword exports routing information to the target VPN extended community.
		• The route-target-ext-community argument adds the route-target extended community attributes to the VRF's list of import or export route-target extended communities. You can enter the route-target-ext-community argument in either of these formats:
		  • 16-bit or 32-bit AS number: your 32-bit number, for example, 1.2:3
		  • 32-bit IP address: your 16-bit number, for example, 192.0.2.1:1

Step 9	**vrf context** *vrf-spoke*
	Example:
	`switch(config-vrf-af-ipv4)# vrf context 2spokes`
	`switch(config-vrf)#`
		Defines the VPN routing instance for the PE spoke by assigning a VRF name and enters VRF configuration mode. The vrf-spoke argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 10	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af-ipv4)#`
		Specifies the IPv4 address family type and enters address family configuration mode.

Step 11	**route-target** { **import** | **export** } *route-target-ext-community* **}** **Example:** `switch(config-vrf-af-ipv4)# route-target export 1:100`	Specifies a route-target extended community for a VRF as follows: • Creates a route-target extended community for a VRF. The **import** keyword imports routing information from the target VPN extended community. The **export** keyword exports routing information to the target VPN extended community. The route-target-ext-community argument adds the route-target extended community attributes to the VRF's list of import or

	Command or Action	Purpose
		export route-target extended communities. You can enter the route-target-ext-community argument in either of these formats: • 16-bit or 32-bit AS number: your 32-bit number, for example, 1.2:3 • 32-bit IP address: your 16-bit number, for example, 192.0.2.1:1
Step 12	**show running-config vrf** *vrf-name* **Example:** `switch(config-vrf-af-ipv4)# show running-config vrf 2spokes`	(Optional) Displays the running configuration for the VRF. The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters. .
Step 13	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring eBGP on the Hub PE Router

You can use eBGP to configure PE-to-CE hub routing sessions.

Note If all CE sites are using the same BGP AS number, you must perform the following tasks:

- Configure either the BGP **as-override** command at the PE (hub) or the **allowas-in** command at the receiving CE router.
- To advertise BGP routes learned from one ASN back to the same ASN, configure the **disable-peer-as-check** command at the PE router to prevent loopback.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature-set mpls** **Example:**	Enables the MPLS feature-set.

	Command or Action	Purpose
	`switch(config)# feature-set mpls`	
Step 3	**feature mpls l3vpn** **Example:** `switch(config)# feature mpls l3vpn`	Enables the MPLS Layer 3 VPN feature.

Step 4	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enables the BGP feature.

Step 5	**router bgp** *as - number*
	Example:
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		Configures a BGP routing process and enters router configuration mode.
		The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 6	**neighbor** *ip-address* **remote-as** *as-number*
	Example:
	`switch(config-router)# neighbor 209.165.201.1 remote-as 1.2`
	`switch(config-router-neighbor)#`
		Adds an entry to the iBGP neighbor table.
		• The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.
		• The as-number argument specifies the autonomous system to which the neighbor belongs.

Step 7	**address-family** { **ipv4** | **ipv6** } **unicast**
	Example:
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		Specifies the IP address family type and enters address family configuration mode.

Step 8	**send-community extended** **Example:** `switch(config-router-neighbor-af)# send-community extended`	(Optional) Configures BGP to advertise extended community lists.

Step 9	**vrf** *vrf-hub*
	Example:
	`switch(config-router-neighbor-af)# vrf 2hub`
	`switch(config-router-vrf)#`
		Enters VRF configuration mode. The *vrf-hub* argument is any case-sensitive, alphanumeric string up to 32 characters.

	Command or Action	Purpose

Step 10	**neighbor** *ip-address* **remote-as** *as-number*
	Example:
	`switch(config-router-vrf)# neighbor 33.0.0.33 1 remote-as 150`
	`switch(config-router-vrf-neighbor)#`
		Adds an entry to the BGP or multiprotocol BGP neighbor table for this VRF.
		• The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.
		• The as-number argument specifies the autonomous system to which the neighbor belongs.

Step 11	**address-family** { **ipv4** | **ipv6** } **unicast**
	Example:
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router--vrf-neighbor-af)#`
		Specifies the IP address family type and enters address family configuration mode.

Step 12	**as-override** **Example:** `switch(config-router-vrf-neighbor-af)# as-override`	(Optional) Overrides the AS-number when sending an update. If all BGP sites are using the same AS number, of the following commands: • Configure the BGP as-override command at the PE (hub) or • Configure the allowas-in command at the receiving CE router.

Step 13	**vrf** *vrf-spoke*
	Example:
	`switch(config-router-vrf-neighbor-af)# vrf 2spokes`
	`switch(config-router-vrf)#`
		Enters VRF configuration mode. The vrf-spoke argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 14	**neighbor** *ip-address* **remote-as** *as-number*
	Example:
	`switch(config-router-vrf)# neighbor 33.0.0.33 1 remote-as 150`
	`switch(config-router-vrf-neighbor)#`
		Adds an entry to the BGP or multiprotocol BGP neighbor table for this VRF.
		• The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.
		• The as-number argument specifies the autonomous system to which the neighbor belongs.

Step 15	**address-family** { **ipv4** | **ipv6** } **unicast** **Example:**	Specifies the IP address family type and enters address family configuration mode.

	Command or Action	Purpose
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast switch(config-router--vrf-neighbor-af)#`	
Step 16	**allowas-in** [*number*] **Example:** `switch(config-router-vrf-neighbor-af)# allowas-in 3`	(Optional) Allows duplicate AS numbers in the AS path. Configure this parameter in the VPN address family configuration mode at the PE spokes and at the neighbor mode at the PE hub.
Step 17	**show running-config bgp** *vrf-name* **Example:** `switch(config-router-vrf-neighbor-af)# show running-config bgp`	(Optional) Displays the running configuration for BGP.
Step 18	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring eBGP on the Hub CE Router

You can use eBGP to configure PE-to-CE hub routing sessions.

Note If all CE sites are using the same BGP AS number, you must perform the following tasks:

- Configure either the as-override command at the PE (hub) or the allowas-in command at the receiving CE router.
- Configure the disable-peer-as-check command at the CE router.
- To advertise BGP routes learned from one ASN back to the same ASN, configure the disable-peer-as-check command at the PE router to prevent loopback.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal switch(config)#`	Enters global configuration mode.
Step 2	**feature-set mpls** **Example:** `switch(config)# feature-set mpls`	Enables the MPLS feature-set.

	Command or Action	Purpose
Step 3	**feature mpls l3vpn** **Example:** `switch(config)# feature mpls l3vpn`	Enables the MPLS Layer 3 VPN feature.

Step 4	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enables the BGP feature.

Step 5	**router bgp** *as - number*
	Example:
	`switch(config)# router bgp 1.1`
	`switch(config-router)#`
		Configures a BGP routing process and enters router configuration mode.
		The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 6	**neighbor** *ip-address***remote-as** *as-number*
	Example:
	`switch(config-router)# neighbor 209.165.201.1 remote-as 1.2`
	`switch(config-router-neighbor)#`
		Adds an entry to the iBGP neighbor table.
		• The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.
		• The as-number argument specifies the autonomous system to which the neighbor belongs.

Step 7	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		Specifies the IP address family type and enters address family configuration mode.

Step 8	**send-community extended** **Example:** `switch(config-router-neighbor-af)# send-community extended`	(Optional) Configures BGP to advertise extended community lists.

Step 9	**vrf** *vrf-hub*
	Example:
	`switch(config-router-neighbor-af)# vrf 2hub`
	`switch(config-router-vrf)#`
		Enters VRF configuration mode. The *vrf-hub* argument is any case-sensitive, alphanumeric string up to 32 characters.

	Command or Action	Purpose

Step 10	**neighbor** *ip-address***remote-as** *as-number*
	Example:
	`switch(config-router-vrf)# neighbor 33.0.0.33 1 remote-as 150`
	`switch(config-router-vrf-neighbor)#`
		Adds an entry to the BGP or multiprotocol BGP neighbor table for this VRF.
		• The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.
		• The as-number argument specifies the autonomous system to which the neighbor belongs.

Step 11	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router--vrf-neighbor-af)#`
		Specifies the IP address family type and enters address family configuration mode.

Step 12	**as-override** **Example:** `switch(config-router-vrf-neighbor-af)# as-override`	(Optional) Overrides the AS-number when sending an update. If all BGP sites are using the same AS number, of the following commands: • Configure the BGP **as-override** command at the PE (hub) or • Configure the **allowas-in** command at the receiving CE router.

Step 13	**vrf** *vrf-spoke*
	Example:
	`switch(config-router-vrf-neighbor-af)# vrf 2spokes`
	`switch(config-router-vrf)#`
		Enters VRF configuration mode. The vrf-spoke argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 14	**neighbor** *ip-address***remote-as** *as-number*
	Example:
	`switch(config-router-vrf)# neighbor 33.0.0.33 1 remote-as 150`
	`switch(config-router-vrf-neighbor)#`
		Adds an entry to the BGP or multiprotocol BGP neighbor table for this VRF.
		• The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.
		• The as-number argument specifies the autonomous system to which the neighbor belongs.

Step 15	**address-family { ipv4 | ipv6 } unicast** **Example:**	Specifies the IP address family type and enters address family configuration mode.

	Command or Action	Purpose
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast` `switch(config-router--vrf-neighbor-af)#`	
Step 16	**allowas-in** [*number*] **Example:** `switch(config-router-vrf-neighbor-af)# allowas-in 3`	(Optional) Allows duplicate AS numbers in the AS path. Configure this parameter in the VPN address family configuration mode at the PE spokes and at the neighbor mode at the PE hub.
Step 17	**show running-config bgp** *vrf-name* **Example:** `switch(config-router-vrf-neighbor-af)# show running-config bgp`	(Optional) Displays the running configuration for BGP.
Step 18	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring VRFs on the Spoke PE Router

You can configure hub and spoke VRFs on the spoke PE router.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**install feature-set mpls**
	Example:
	`switch(config)# install feature-set mpls`
	`switch(config)#`
		Installs the MPLS feature set.

Step 3	**feature-set mpls**
	Example:
	`switch(config)# feature-set mpls`
	`switch(config)#`
		Enables the MPLS feature-set.

Step 4	**feature-set mpls l3vpn**
	Example:
	`switch(config)# feature-set mpls l3vpn`
	`switch(config)#`
		Enables the MPLS Layer 3 VPN feature.

	Command or Action	Purpose

Step 5	**vrf context** *vrf-spoke*
	Example:
	`switch(config)# vrf context spoke`
	`switch(config-vrf)#`
		Defines the VPN routing instance for the PE spoke by assigning a VRF name and enters VRF configuration mode. The vrf-spoke argument is any case-sensitive, alphanumeric string up to 32 characters.

Step 6	**rd** *route-distinguisher*
	Example:
	`switch(config-vrf)# rd 1.101`
	`switch(config-vrf)#`
		Configures the route distinguisher. The route-distinguisher argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter an RD in either of these formats:
		• 16-bit or 32-bit AS number: your 32-bit number, for example, 1.2:3
		• 32-bit IP address: your 16-bit number, for example, 192.0.2.1:1

Step 7	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-vrf)# address-family ipv4 unicast`
	`switch(config-vrf-af-ipv4)#`
		Specifies the IPv4 address family type and enters address family configuration mode.

Step 8	**route-target** { **import** | **export** } route-target-ext-community **}**
	Example:
	`switch(config-vrf-af-ipv4)# route-target import 1.0:1`
		Specifies a route-target extended community for a VRF as follows:
		• The **import** keyword imports routing information from the target VPN extended community.
		• The **export** keyword exports routing information to the target VPN extended community.
		• The route-target-ext-community argument adds the route-target extended community attributes to the VRF's list of import or export route-target extended communities. You can enter the route-target-ext-community argument in either of these formats:
		  • 16-bit or 32-bit AS number: your 32-bit number, for example, 1.2:3
		  • 32-bit IP address: your 16-bit number, for example, 192.0.2.1:1

	Command or Action	Purpose
Step 9	**show running-config vrf** *vrf-name* **Example:** `switch(config-vrf-af-ipv4)# show running-config vrf 2spokes`	(Optional) Displays the running configuration for the VRF. The vrf-name argument is any case-sensitive, alphanumeric string up to 32 characters. .
Step 10	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring eBGP on the Spoke PE Router

You can use eBGP to configure PE spoke routing sessions.

Note If all CE sites are using the same BGP AS number, you must perform the following tasks:

- Configure the allowas-in command at the perceiving spoke router.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature-set mpls** **Example:** `switch(config)# feature-set mpls`	Enables the MPLS feature-set.
Step 3	**feature mpls l3vpn** **Example:** `switch(config)# feature mpls l3vpn`	Enables the MPLS Layer 3 VPN feature.

Step 4	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enables the BGP feature.

Step 5	**router bgp** *as - number* **Example:**	Configures a BGP routing process and enters router configuration mode.

	Command or Action	Purpose
	`switch(config)# router bgp 100` `switch(config-router)#`	The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Step 6	**neighbor** *ip-address***remote-as** *as-number*
	Example:
	`switch(config-router)# neighbor 63.63.0.63 remote-as 100`
	`switch(config-router-neighbor)#`
		Adds an entry to the iBGP neighbor table.
		• The ip-address argument specifies the IP address of the neighbor in dotted decimal notation.
		• The as-number argument specifies the autonomous system to which the neighbor belongs.

Step 7	**address-family { ipv4 | ipv6 } unicast**
	Example:
	`switch(config-router-vrf-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		Specifies the IPv4 or IPv6 address family type and enters address family configuration mode.

Step 8	**allowas-in** *number* **Example:** `switch(config-router-vrf-neighbor-af)# allowas-in 3`	(Optional) Allows an AS path with the PE ASN for a specified number of times. • The range is from 1 to 10. • If all BGP sites are using the same AS number, configure the following commands: **Note** Configure the **BGP as-override** command at the PE (hub) or Configure the **allowas-in** command at the receiving CE router. The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
Step 9	**send-community extended** **Example:** `switch(config-router-neighbor)# send-community extended`	(Optional) Configures BGP to advertise extended community lists.

	Command or Action	Purpose
Step 10	**show running-config bgp** **Example:** `switch(config-router-vrf-neighbor-af)# show running-config bgp`	(Optional) Displays the running configuration for BGP.
Step 11	**copy running-config startup-config** **Example:** `switch(config-router-vrf)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring MPLS using Hardware Profile Command

Beginning with release 7.0(3)F3(3), Cisco Nexus 9508 switches with N9K-X9636C-R, N9K-X9636C-RX, and N9K-X9636Q-R line cards supports multiple hardware profiles. You can configure MPLS and/or VXLAN using hardware profile configuration command in a switch. The hardware profile configuration command invokes appropriate configuration files that are available on the switch. VXLAN is enabled by default

Before you begin

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature bgp**
	Example:
	`switch(config)# feature bgp`
	`switch(config)#`
		Enables the BGP feature.

Step 3	**hardware profile** [**vxlan** | **mpls**] **module all** **Example:** `switch(config)# hardware profile mpls module all`	Enables MPLS on all the switch modules. .

Step 4	**show hardware profile module** [**all** | *number*]
	Example:
	`switch(config)# show hardware profile module all`
	`switch(config)#`
		Displays the hardware profile of all the modules or specific module.

Step 5	**show module internal sw info** | [**i** | **mpls**] **Example:**	Displays the switch software information.

	Command or Action	Purpose
	`switch(config)# show module internal sw info`	
Step 6	**show running configuration** | [**i** | **mpls**] **Example:** `switch(config)# show module internal sw info`	Displays the running configuration.