MPLS VPN VRF CLI for IPv4 and IPv6 VPNs

The MPLS VPN VRF CLI for IPv4 and IPv6 VPNs feature introduces Cisco commands that allow you to enable an IPv4 and IPv6 VPN in the same VRF instance and to simplify the migration from a single-protocol VRF configuration to a multiprotocol VRF configuration. A multiprotocol VRF allows you to share route targets policies (import and export) between IPv4 and IPv6 or to configure separate route-target policies for IPv4 and IPv6 VPNs.

This document describes how to configure a Virtual Private Network (VPN) routing and forwarding (VRF) instance for IPv4 and IPv6 VPNs and describes how to upgrade your existing single-protocol IPv4-only VRF to a multiprotocol VRF configuration.

- Finding Feature Information, on page 1
- Prerequisites for MPLS VPN VRF CLI for IPv4 and IPv6 VPNs, on page 1
- Restrictions for MPLS VPN VRF CLI for IPv4 and IPv6 VPNs, on page 2
- Information About MPLS VPN VRF CLI for IPv4 and IPv6 VPNs, on page 2
- How to Configure MPLS VPN VRF CLI for IPv4 and IPv6 VPNs, on page 4
- Configuration Examples for MPLS VPN VRF CLI for IPv4 and IPv6 VPNs, on page 13
- Additional References, on page 17
- Feature Information for MPLS VPN VRF CLI for IPv4 and IPv6 VPNs, on page 17
- Glossary, on page 18

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for MPLS VPN VRF CLI for IPv4 and IPv6 VPNs

- For migration—An IPv4 Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) virtual routing and forwarding (VRF) instance must exist.

- For a new VRF configuration—Cisco Express Forwarding and an MPLS label distribution method, either Label Distribution Protocol (LDP) or MPLS traffic engineering (TE), must be enabled on all devices in the core, including the provider edge (PE) devices.

Restrictions for MPLS VPN VRF CLI for IPv4 and IPv6 VPNs

- Once you have converted to a multiprotocol virtual routing and forwarding (VRF) instance, you cannot convert the VRF back to an IPv4-only single-protocol VRF.
- You can associate an interface with only one VRF. You cannot configure a VRF for IPv4 and a different VRF for IPv6 on the same interface.
- You can configure only IPv4 and IPv6 address families in a multiprotocol VRF. Other protocols (IPX, AppleTalk, and the like) are not supported.

Information About MPLS VPN VRF CLI for IPv4 and IPv6 VPNs

VRF Concepts Similar for IPv4 and IPv6 MPLS VPNs

Virtual Private Networks (VPNs) for IPv6 use the same virtual routing and forwarding (VRF) concepts that IPv4 Multiprotocol Label Switching (MPLS) VPNs use, such as address families, route distinguishers, route targets, and VRF identifiers. Customers that use both IPv4 and IPv6 VPNs might want to share VRF policies between address families. They might want a way to define applicable VRF policies for all address families, instead of defining VRF policies for an address family individually as they do for or a single-protocol IPv4-only VRF.

Prior to the introduction of the MPLS VPN VRF CLI for IPv4 and IPv6 VPNs feature, a VRF applied only to an IPv4 address family. A one-to-one relationship existed between the VRF name and a routing and forwarding table identifier, between a VRF name and a route distinguisher (RD), and between a VRF name and a VPN ID. This configuration is called a single-protocol VRF.

The MPLS VPN VRF CLI for IPv4 and IPv6 VPNs feature introduces support for a multiple address-family (multi-AF) VRF structure. The multi-AF VRF allows you to define multiple address families under the same VRF. A given VRF, identified by its name and a set of policies, can apply to both an IPv4 VPN and an IPv6 VPN at the same time. This VRF can be activated on a given interface, even though the routing and forwarding tables are different for the IPv4 and IPv6 protocols. This configuration is called a multiprotocol VRF.

Single-Protocol VRF to Multiprotocol VRF Migration

Prior to the introduction of the MPLS VPN VRF CLI for IPv4 and IPv6 VPNs feature, you could create a single-protocol IPv4-only virtual routing and forwarding (VRF) instance. You created a single-protocol VRF by entering the **ip vrf** command. To activate the single-protocol VRF on an interface, you entered the **ip vrf forwarding** (interface configuration) command.

After the introduction of the MPLS VPN VRF CLI for IPv4 and IPv6 VPNs feature, you create a multiprotocol VRF by entering the **vrf definition** command. To activate the multiprotocol VRF on an interface, you enter the **vrf forwarding** command.

The MPLS VPN VRF CLI for IPv4 and IPv6 VPNs feature introduces the **vrf upgrade-cli multi-af-mode** {**common-policies** | **non-common-policies**} [**vrf** *vrf-name*] command that forces VRF configuration migration from a single-protocol VRF model to a multiprotocol VRF model:

- If the route-target policies apply to all address families configured in the multi-AF VRF, use the **common-policies** keyword.
- If the route-target policies apply only to the IPv4 address family that you are migrating, use the **non-common-policies** keyword.

After you enter the **vrf upgrade-cli** command and save the configuration to NVRAM, the single-protocol VRF configuration is saved as a multiprotocol VRF configuration. In the upgrade process, the**ip vrf** command is converted to the **vrf definition** command (global configuration commands) and the **ip vrf forwarding** command is converted to the **vrf forwarding** command (interface configuration command). The **vrf upgrade-cli** command has a one-time immediate effect.

You might have both IPv4-only VRFs and multiprotocol VRFs on your device. Once you create a VRF, you can edit it using only the commands in the mode in which it was created. For example, you created a VRF named vrf2 with the following multiprotocol VRF commands:

```
Device# configure terminal
Enter configuration command, one per line. End with CNTL/Z
Device(config)# vrf definition vrf2
Device(config-vrf)# rd 2:2
Device(config-vrf)# route-target import 2:2
Device(config-vrf)# route-target export 2:2
Device(config-vrf)# end
```

If you try to edit VRF vrf2 with IPv4-only VRF commands, you receive the following message:

```
Device# configure terminal
Enter configuration command, one per line. End with CNTL/Z
Device(config)# ip vrf vrf2
% Use ‘vrf definition vrf2’ command
```

If you try to edit an IPv4-only VRF with the multiprotocol VRF commands, you receive this message, where <vrf-name> is the name of the IPv4-only VRF:

```
% Use ‘ip vrf <vrf-name>’ command
```

The **ip vrf** *name* and **ip vrf forwarding** (interface configuration) commands will be available for a period of time before they are removed. Use the **vrf upgrade-cli** command to migrate your older IPv4-only VRFs to the new multiprotocol VRF configuration. When you need to create a new VRF—whether the VRF is for an IPv4 VPN, or IPv6 VPN, or both—use the multiprotocol VRF **vrf definition** and **vrf forwarding** commands that support a multi-AF configuration.

Multiprotocol VRF Configuration Characteristics

In a multiprotocol virtual routing and forwarding (VRF) configuration, you can configure both IPv4 VRFs and IPv6 VRFs under the same address family or configure separate VRFs for each IPv4 or IPv6 address family. The multiprotocol VRF configuration has the following characteristics:

- The VRF name identifies a VRF, which might have both IPv4 and IPv6 address families. On the same interface, you cannot have IPv4 and IPv6 address families using different VRF names.

- The route distinguisher (RD), VPN ID, and Simple Network Management Protocol (SNMP) context are shared by both IPv4 and IPv6 address families for a given VRF.
- The policies (route target, for example) specified in multi-AF VRF mode, outside the address-family configuration, are defaults to be applied to each address family. Route targets are the only VRF characteristics that can be defined inside and outside an address family.

The following is also true when you associate a multiprotocol VRF with an interface:

- Binding an interface to a VRF (**vrf forwarding** *vrf-name* command) removes all IPv4 and IPv6 addresses configured on that interface.
- Once you associate a VRF with a given interface, all active address families belong to that VRF. The exception is when no address of the address-family type is configured, in which case the protocol is disabled.
- Configuring an address on an interface that is bound to a VRF requires that the address family corresponding to the address type is active for that VRF. Otherwise, an error message is issued stating that the address family must be activated first in the VRF.

Backward compatibility with the single-protocol VRF CLI is supported in with the introduction of the MPLS VPN VRF CLI for IPv4 and IPv6 VPNs feature. This means that you might have single-protocol and multiprotocol CLI on the same device, but not in the same VRF configuration.

The single-protocol CLI continues to allow you to define an IPv4 address within a VRF and an IPv6 address in the global routing table on the same interface.

How to Configure MPLS VPN VRF CLI for IPv4 and IPv6 VPNs

Configuring a VRF for IPv4 and IPv6 MPLS VPNs

Perform the following task to configure a virtual routing and forwarding (VRF) instance for IPv4 and IPv6 Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs). When you configure a VRF for both IPv4 and IPv6 VPNs (a multiprotocol VRF), you can choose to configure route-target policies that apply to all address families in the VRF, or you can configure route-target policies that apply to individual address families in the VRF.

The following task shows how to configure a VRF that has that has route-target policies defined for IPv4 and IPv6 VPNs in separate VRF address families.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*
5. **address-family** {**ipv4** | **ipv6**}
6. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
7. **exit-address-family**
8. **address-family** {**ipv4** | **ipv6**}

9. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition vrf1`	Configures a VRF routing table and enters VRF configuration mode. • The *vrf-name* argument is the name of the VRF.

Step 4	**rd** *route-distinguisher*
	Example:
	`Device(config-vrf)# rd 100:1`
		Creates routing and forwarding tables for a VRF.
		• The *route-distinguisher* argument specifies to add an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter a route distinguisher in either of these formats:
		  • 16-bit autonomous system number (ASN): your 32-bit number For example, 101:3.
		  • 32-bit IP address: your 16-bit number For example, 192.168.122.15:1.

Step 5	**address-family** {**ipv4** | **ipv6**} **Example:** `Device(config-vrf) address-family ipv4`	Enters VRF address family configuration mode to specify an address family for a VRF. • The **ipv4** keyword specifies an IPv4 address family for a VRF. • The **ipv6** keyword specifies an IPv6 address family for a VRF.
Step 6	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target both 100:2`	Creates a route-target extended community for a VRF. • The **import** keyword specifies to import routing information from the target VPN extended community. • The **export** keyword specifies to export routing information to the target VPN extended community.

	Command or Action	Purpose
		• The **both** keyword specifies to import both import and export 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, export, or both (import and export) route-target extended communities.
Step 7	**exit-address-family** **Example:** `Device(config-vrf-af)# exit-address-family`	Exits from VRF address family configuration mode.
Step 8	**address-family** {**ipv4** | **ipv6**} **Example:** `Device(config-vrf) address-family ipv6`	Enters VRF address family configuration mode to specify an address family for a VRF. • The **ipv4** keyword specifies an IPv4 address family for a VRF. • The **ipv6** keyword specifies an IPv6 address family for a VRF.
Step 9	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Device(config-vrf-af)# route-target both 100:3`	Creates a route-target extended community for a VRF. • The **import** keyword specifies to import routing information from the target VPN extended community. • The **export** keyword specifies to export routing information to the target VPN extended community. • The **both** keyword specifies to import both import and export 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, export, or both (import and export) route-target extended communities. Enter the **route-target** command one time for each target community.
Step 10	**end** **Example:** `Device(config-vrf-af)# end`	Returns to privileged EXEC mode.

Associating a Multiprotocol VRF with an Interface

Perform the following task to associate a multiprotocol virtual routing and forwarding (VRF) instance with an interface. Associating the VRF with an interface activates the VRF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **vrf forwarding** *vrf-name*
5. **ip address** *ip-address mask* [**secondary**]
6. **ipv6 address** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*}
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet 0/1`	Configures an interface type and enters interface configuration mode. • The *type* argument identifies the type of interface to be configured. • The *number* argument identifies the port, connector, or interface card number.
Step 4	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding vrf1`	Associates a VRF with an interface or subinterface. • The *vrf-name* argument is the name of the VRF.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.24.24.24 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask of the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is

	Command or Action	Purpose
		omitted, the configured address is the primary IP address.
Step 6	**ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*} **Example:** `Device(config-if)# ipv6 address 2001:0DB8:0300:0201::/64`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface. • The *ipv6-address* argument is the IPv6 address to be used. • The *prefix-length* argument is the length of the IPv6 prefix, which is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value. • The *prefix-name* argument is a general prefix that specifies the leading bits of the network to be configured on the interface. • The *sub-bits* argument is the subprefix bits and host bits of the address to be concatenated with the prefixes provided by the general prefix specified with the *prefix-name* argument. The *sub-bits* argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
Step 7	**end** **Example:** `Device(config-if) end`	Returns to privileged EXEC mode.

Verifying the MPLS VPN VRF CLI for IPv4 and IPv6 VPNs Configuration

Perform the following task to verify the MPLS VPN VRF CLI for IPv4 and IPv6 VPNs feature configuration, that is, to show that the virtual routing and forwarding (VRF) configuration is upgraded to a multi-AF multiprotocol VRF.

SUMMARY STEPS

1. **enable**
2. **show running-config vrf** [*vrf-name*]
3. **show vrf**
4. **show vrf detail** [*vrf-name*]
5. **exit**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password, if prompted. For example:

Example:

```
Device> enable
Device#
```

Step 2 **show running-config vrf** [*vrf-name*]

Verifies that the upgrade to a multi-AF multiprotocol VRF configuration was successful. The following is sample command output before the upgrade to a multi-AF multiprotocol VRF:

Example:

```
Device# show running-config vrf vpn2

Building configuration...
Current configuration : 604 bytes
ip vrf vpn2
 rd 1:1
 route-target both 1:1
!
!
interface Loopback1
 ip vrf forwarding vpn2
 ip address 10.43.43.43 255.255.255.255
!
```

The following is sample command output after you upgrade to a multi-AF multiprotocol VRF with common policies for all address families:

Example:

```
Device# show running-config vrf vpn1

Building configuration...
Current configuration : 604 bytes
vrf definition vpn1
 rd 1:1
 route-target both 1:1
!
 address-family lpv4
 exit-address-family
!
!
interface Loopback1
 ip vrf forwarding vpn1
 ip address 10.43.43.43 255.255.255.255
!
```

This configuration contains the **vrf definition** command. The **vrf definition** command replaces the **ip vrf** command in the multi-AF multiprotocol VRF configuration.

Step 3 **show vrf**

Verifies that the upgrade to a multi-AF multiprotocol VRF configuration was successful. The **show vrf** command replaces the **show ip vrf** command when a VRF configuration is updated to a multi-AF multiprotocol VRF configuration. The **show vrf** command displays the protocols defined for a VRF. The following command shows sample output after you upgrade a single-protocol VRF configuration to a multi-AF multiprotocol VRF configuration:

Example:

```
Device# show vrf vpn1

  Name                             Default RD     Protocols         Interfaces
  vpn1                             1:1            ipv4              Lo1/0
```

The following is sample output from the **show ip vrf vp1** command. Compare this to the output of the **show vrf vpn1** command. The protocols under the VRF are not displayed.

Example:

```
Device# show ip vrf vrf1

  Name      Default RD   Interface
  vpn1      1:1          Loopback1
```

The following is sample output from the **show vrf** command for multiprotocol VRFs, one of which contains both IPv4 and IPv6 protocols:

Example:

```
Device# show vrf

  Name                             Default RD     Protocols         Interfaces
  vpn1                             1:1            ipv4              Lo1/0
  vpn2                             100:3          ipv4              Lo23  AT3/0/0.1
  vpn4                             100:2          ipv4,ipv6
```

Step 4 **show vrf detail** [*vrf-name*]

Displays all characteristics of the defined VRF to verify that the configuration is as you expected. For example, if your VRF configuration for VRF vpn1 is as follows:

Example:

```
vrf definition vpn1
 route-target both 100:1
 route-target import 100:2
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 route-target both 100:1
 route-target import 100:3
 exit-address-family
```

This command displays the following:

Example:

```
Device# show vrf detail vpn1

VRF vpn1 (VRF Id = 3); default RD <not set>; default VPNID <not set>
```

```
  No interfaces
Address family ipv4 (Table ID = 3 (0x3)):
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:100:1
  Import VPN route-target communities
    RT:100:1                RT:100:2
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
Address family ipv6 (Table ID = 503316483 (0x1E000003)):
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:100:1
  Import VPN route-target communities
    RT:100:1                RT:100:3
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
```

Step 5 **exit**

Returns to user EXEC mode. For example:

Example:

```
Device# exit
Device>
```

Migrating from a Single-Protocol IPv4-Only VRF to a Multiprotocol VRF Configuration

Perform the following task to force migration from a single-protocol IPv4-only virtual routing and forwarding (VRF) configuration to a multiprotocol VRF configuration.

The multiprotocol VRF configuration allows you to define multiple address families under the same VRF. A given VRF, identified by its name and a set of policies, can apply to both an IPv4 VPN and an IPv6 VPN at the same time. This VRF can be activated on a given interface, even though the routing and forwarding tables are different for the IPv4 and IPv6 protocols.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf upgrade-cli multi-af-mode** {**common-policies** | **non-common-policies**} [**vrf** *vrf-name*]
4. **exit**
5. **show running-config vrf** [*vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vrf upgrade-cli multi-af-mode** {**common-policies** | **non-common-policies**} [**vrf** *vrf-name*] **Example:** `Device(config)# vrf upgrade-cli multi-af-mode common-policies vrf vpn4`	Upgrades a VRF instance or all VRFs configured on the device to support multiple address families under the same VRF. • The **multi-af-mode** keyword specifies an upgrade of a single-protocol VRF or all VRFs to a multiprotocol VRF that supports multi-AFs configuration. • The **common-policies** keyword specifies to copy the route-target policies to the common part of the VRF configuration so that the policies apply to all address families configured in the multi-AF VRF. • The **non-common-policies** keyword specifies to copy the route-target policies to the IPv4 address family part of the VRF configuration so that the policies apply only to IPv4. • The **vrf** keyword specifies a VRF for the upgrade to a multi-AF VRF configuration. • The *vrf-name* argument is the name of the single-protocol VRF to upgrade to a multi-AF VRF configuration.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits to privileged EXEC mode.
Step 5	**show running-config vrf** [*vrf-name*] **Example:** `Device# show running-config vrf vpn4`	Displays the subset of the running configuration of a device that is linked to a specific VRF instance or to all VRFs configured on the device. • The *vrf-name* argument is the name of the VRF of which you want to display the configuration.

	Command or Action	Purpose
		Note The Cisco software image that supports the multiprotocol VRF commands might not support the **show running-config vrf** command. You can use the **show running-config** command instead.

Configuration Examples for MPLS VPN VRF CLI for IPv4 and IPv6 VPNs

Example: Multiprotocol VRF Configuration Single Protocol with Noncommon Policies

The following is an example of a multiprotocol virtual routing and forwarding (VRF) configuration for a single protocol (IPv4) with route-target policies in the address family configuration:

```
vrf definition vrf2
 rd 2:2
 !
 address-family ipv4
 route-target export 2:2
 route-target import 2:2
 exit-address-family
```

The RD (2:2) applies to all address families defined for VRF vrf2.

Example: Multiprotocol VRF Configuration Multiprotocol with Noncommon Policies

The following is an example of a multiprotocol virtual routing and forwarding (VRF) configuration for IPv4 and IPv6 Virtual Private Networks (VPNs) in which the route-target policies are defined in the separate address family configurations:

```
vrf definition vrf2
 rd 2:2
 !
 address-family ipv4
 route-target export 2:2
 route-target import 2:2
 exit-address-family
 !
 address-family ipv6
 route-target export 3:3
 route-target import 3:3
 exit-address-family
```

Example: Multiprotocol VRF Configuration Multiprotocol with Common Policies

The following is an example of a multiprotocol virtual routing and forwarding (VRF) configuration for IPv4 and IPv6 Virtual Private Networks (VPNs) with route-target policies defined in the global part of the VRF:

```
vrf definition vrf2
 rd 2:2
 route-target export 2:2
 route-target import 2:2
  !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
```

The route-target policies are defined outside the address family configurations. Therefore, the policies apply to all address families defined in VRF vrf2.

Example: Multiprotocol VRF Configuration Multiprotocol with Common and Noncommon Policies

The following is an example of a multiprotocol virtual routing and forwarding (VRF) configuration with route-target policies defined in both global and address family areas:

- For IPv6, the route-target definitions are defined under the address family. These definitions are used and the route-target definitions in the global area are ignored. Therefore, the IPv6 Virtual Private Network (VPN) ignores import 100:2.
- For IPv4, no route-target policies are defined under the address family, therefore, the global definitions are used.

```
vrf definition vfr1
 route-target export 100:1
 route-target import 100:1
 route-target import 100:2
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 route-target export 100:1
 route-target import 100:1
 route-target import 100:3
 exit-address-family
```

Examples: Configuring a VRF for IPv4 and IPv6 VPNs

The following example shows how to configure a virtual routing and forwarding (VRF) instance for IPv4 and IPv6 Virtual Private Networks (VPNs):

```
configure terminal
!
vrf definition vrf1
 rd 100:1
```

```
!
 address-family ipv4
 route-target both 100:2
 exit-address-family
!
 address-family ipv6
 route-target both 100:3
 exit-address-family
```

In this example, noncommon policies are defined in the address family configuration.

The following is an example of a VRF for IPv4 and IPv6 that has common policies defined in the global part of the VRF configuration:

```
configure terminal
!
vrf definition vrf2
 rd 200:1
 route-target both 200:2
!
 address-family ipv4
 exit-address-family
!
 address-family ipv6
 exit-address-family
 end
```

Example: Associating a Multiprotocol VRF with an Interface

The following example shows how to associate a multiprotocol virtual routing and forwarding (VRF) instance with an interface:

```
configure terminal
!
interface Ethernet 0/1
 vrf forwarding vrf1
 ip address 10.24.24.24 255.255.255.255
 ipv6 address 2001:0DB8:0300:0201::/64
 end
```

Examples: Migrating from a Single-Protocol IPv4-Only VRF Configuration to a Multiprotocol VRF Configuration

This section contains examples that show how to migrate from a single-protocol IPv4-only virtual routing and forwarding (VRF) configuration to a multiprotocol VRF configuration.

This example shows a single-protocol IPv4-only VRF before the VRF CLI for IPv4 and IPv6 is entered on the device:

```
ip vrf vrf1
 rd 1:1
 route-target both 1:1
interface Loopback1
 ip vrf forwarding V1
 ip address 10.3.3.3 255.255.255.255
```

This example shows how to force the migration of the single-protocol VRF vrf1 to a multiprotocol VRF configuration:

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
!
Device(config)# vrf upgrade-cli multi-af-mode common-policies vrf vrf1
You are about to upgrade to the multi-AF VRF syntax commands.
You will loose any IPv6 address configured on interfaces
belonging to upgraded VRFs.
Are you sure ? [yes]: yes
Number of VRFs upgraded: 1
Device(config)# exit
```

This example shows the multiprotocol VRF configuration after the forced migration:

```
vrf definition vrf1
 rd 1:1
 route-target both 1:1
 !
 address-family ipv4
 exit-address-family
!
interface Loopback1
 vrf forwarding V1
 ip address 10.3.3.3 255.255.255.255
```

The following is another example of a multi-AF multiprotocol VRF configuration:

```
vrf definition vrf2
 rd 100:1
 address family ipv6
 route-target both 200:1
 exit-address-family
!
ip vrf vrf1
 rd 200:1
 route-target both 200:1
!
interface Ethernet0/0
 vrf forwarding vrf2
 ip address 10.50.1.2 255.255.255.0
 ipv6 address 2001:0DB8:0:1::/64
!
interface Ethernet0/1
 ip vrf forwarding vrf1
 ip address 10.60.1.2 255.255.255.0
 ipv6 address 2001:0DB8:1 :1::/64
```

In this example, all addresses (IPv4 and IPv6) defined for interface Ethernet0/0 are in VRF vrf2. For the interface Ethernet0/1, the IPv4 address is defined in VRF vrf1 but the IPv6 address is in the global IPv6 routing table.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco Master Command List, All Releases
MPLS and MPLS applications commands	Cisco IOS Multiprotocol Label Switching Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 1771	*A Border Gateway Protocol 4 (BGP-4)*
RFC 4364	*BGP MPLS/IP Virtual Private Networks (VPNs)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for MPLS VPN VRF CLI for IPv4 and IPv6 VPNs

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for MPLS VPN VRF CLI for IPv4 and IPv6 VPNs

Feature Name	Releases	Feature Information
MPLS VPN VRF CLI for IPv4 and IPv6 VPNs	12.2(33)SB 12.2(33)SRB 12.2(33)SXI 12.4(20)T Cisco IOS XE Release 3.1S	This document describes how to configure a multiprotocol Virtual Private Network (VPN) routing and forwarding (VRF) instance for IPv4 and IPv6 VPNs and describes how to upgrade your existing single-protocol IPv4-only VRF to a multiprotocol VRF configuration. The MPLS VPN VRF CLI for IPv4 and IPv6 VPNs feature introduces commands that allow you to enable an IPv4 and IPv6 VPN in the same Multiprotocol Label Switching (MPLS) VRF configuration and to simplify the migration from a single-protocol VRF configuration to a multiprotocol VRF configuration. In Cisco IOS Release 12.2(33)SB, this feature was introduced on the Cisco 10000 series router. In Cisco IOS Release 12.2(33)SRB, this feature was implemented on the Cisco 7600 series router. In Cisco IOS Release 12.2(33)SXI, this feature was integrated. In Cisco IOS Release 12.4(2)T, this feature was integrated. In Cisco IOS XE Release 3.1S, this feature was implemented on the Cisco ASR 1000 Series Aggregation Services Routers.
		The following commands were introduced or modified: **show vrf**, **vrf definition**, **vrf forwarding**, **vrf upgrade-cli**.

Glossary

6PE—IPv6 provider edge device or a Multiprotocol Label Switching (MPLS) label switch router (LSR) edge router using IPv6.

6VPE—IPv6 Virtual Private Network (VPN) provider edge device.

AF—address family. Set of related communication protocols in which all members use a common addressing mechanism to identify endpoints. Also called protocol family.

AFI—Address Family Identifier. Carries the identity of the network-layer protocol that is associated with the network address.

BGP—Border Gateway Protocol. A routing protocol used between autonomous systems. It is the routing protocol that makes the internet work. BGP is a distance-vector routing protocol that carries connectivity information and an additional set of BGP attributes. These attributes allow for a set of policies for deciding the best route to use to reach a given destination. BGP is defined by RFC 1771.

CE—customer edge device. A service provider device that connects to Virtual Private Network (VPN) customer sites.

FIB—Forwarding Information Base. Database that stores information about switching of data packets. A FIB is based on information in the Routing Information Base (RIB). It is the optimal set of selected routes that are installed in the line cards for forwarding.

HA—high availability. High availability is defined as the continuous operation of systems. For a system to be available, all components--including application and database servers, storage devices, and the end-to-end network--need to provide continuous service.

IP—Internet Protocol. Network-layer protocol in the TCP/IP stack offering a connectionless internetwork service. IP provides features for addressing, type-of-service specification, fragmentation and reassembly, and security.

IPv4—IP Version 4. Network layer for the TCP/IP protocol suite. IPv4 is a connectionless, best-effort packet switching protocol.

IPv6—IP Version 6. Replacement for IPv4. IPv6 is a next-generation IP protocol. IPv6 is backward compatible with and designed to fix the shortcomings of IPv4, such as data security and maximum number of user addresses. IPv6 increases the address space from 32 to 128 bits, providing for an unlimited number of networks and systems. It also supports quality of service (QoS) parameters for real-time audio and video.

MFI—MPLS Forwarding Infrastructure. In the Cisco MPLS subsystem, the data structure for storing information about incoming and outgoing labels and associated equivalent packets suitable for labeling.

MPLS—Multiprotocol Label Switching. MPLS is a method for forwarding packets (frames) through a network. It enables devices at the edge of a network to apply labels to packets (frames). ATM switches or existing devices in the network core can switch packets according to the labels with minimal lookup overhead.

PE—provider edge device. A device that is part of a service provider’s network and that is connected to a customer edge (CE) device. The PE device function is a combination of an MLS edge label switch router (LSR) function with some additional functions to support Virtual Private Networks (VPNs).

RD (IPv4)—route distinguisher. An 8-byte value that is concatenated with an IPv4 prefix to create a unique VPN IPv4 (VPNv4) prefix.

RD (IPv6)—route distinguisher. A 64-bit value that is prepended to an IPv6 prefix to create a globally unique VPN-IPv6 address.

RIB—Routing Information Base. The set of all available routes from which to choose the Forwarding Information Base (FIB). The RIB essentially contains all routes available for selection. It is the sum of all routes learned by dynamic routing protocols, all directly attached networks (that is-networks to which a given device has interfaces connected), and any additional configured routes, such as static routes.

RT—route target. Extended community attribute used to identify the Virtual Private Network (VPN) routing and forwarding (VRF) routing table into which a prefix is to be imported.

VPN—Virtual Private Network. Enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another. A VPN uses “tunneling” to encrypt all information at the IP level.

VRF—Virtual Private Network (VPN) routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE device.

VRF table—A routing and a forwarding table associated to a Virtual Private Network (VPN) routing and forwarding (VRF) instance. This is a customer-specific table, enabling the provider edge (PE) device to maintain independent routing states for each customer.