



# Implementing IPv6 VPN Provider Edge Transport over MPLS

IPv6 Provider Edge or IPv6 VPN Provider Edge (6PE/VPE) uses the existing MPLS IPv4 core infrastructure for IPv6 transport. 6PE/VPE enables IPv6 sites to communicate with each other over an MPLS IPv4 core network using MPLS label switched paths (LSPs).

This feature relies heavily on multiprotocol Border Gateway Protocol (BGP) extensions in the IPv4 network configuration on the provider edge (PE) router to exchange IPv6 reachability information (in addition to an MPLS label) for each IPv6 address prefix. Edge routers are configured as dual-stack, running both IPv4 and IPv6, and use the IPv4 mapped IPv6 address for IPv6 prefix reachability exchange.

For detailed information about the commands used to configure 6PE/VPE, see the *VPN and Ethernet Services Command Reference for Cisco ASR 9000 Series Routers*.

## Feature History for Implementing 6PE/VPE Transport over MPLS

Release	Modification
Release 3.9.1	This feature was introduced.
Release 4.0.0	Support was added for the 6PE and 6VPE features for IPv6 L3VPN on A9K-SIP-700.  Support was added for the BGP per VRF/CE label allocation for 6PE feature.
Release 4.1.0	Support for the Open Shortest Path First version 3 (OSPFv3) IPv6 VPN Provider Edge (6VPE) feature was added.

- [Prerequisites for Implementing 6PE/VPE, on page 1](#)
- [Information About 6PE/VPE, on page 2](#)
- [How to Implement 6PE/VPE, on page 5](#)
- [Configuration Examples for 6PE/VPE, on page 13](#)

## Prerequisites for Implementing 6PE/VPE

The following prerequisites are required to implement 6PE/VPE:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command.

If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

- Familiarity with MPLS and BGP4 configuration and troubleshooting.

## Information About 6PE/VPE

To configure the 6PE/VPE feature, you should understand the concepts that are described in these sections:

### Overview of 6PE/VPE

Multiple techniques are available to integrate IPv6 services over service provider core backbones:

- Dedicated IPv6 network running over various data link layers
- Dual-stack IPv4-IPv6 backbone
- Existing MPLS backbone leverage

These solutions are deployed on service providers' backbones when the amount of IPv6 traffic and the revenue generated are in line with the necessary investments and the agreed-upon risks. Conditions are favorable for the introduction of native IPv6 services, from the edge, in a scalable way, without any IPv6 addressing restrictions and without putting a well-controlled IPv4 backbone in jeopardy. Backbone stability is essential for service providers that have recently stabilized their IPv4 infrastructure.

Service providers running an MPLS/IPv4 infrastructure follow similar trends because several integration scenarios that offer IPv6 services on an MPLS network are possible. Cisco Systems has specially developed Cisco 6PE or IPv6 Provider Edge Router over MPLS, to meet all those requirements.

Inter-AS support for 6PE requires support of Border Gateway Protocol (BGP) to enable the address families and to allocate and distribute PE and ASBR labels.




---

**Note** Cisco IOS XR displays actual IPv4 next-hop addresses for IPv6 labeled-unicast and VPNv6 prefixes. IPv4-mapped-to-IPv6 format is not supported.

---

### Benefits of 6PE/VPE

Service providers who currently deploy MPLS experience these benefits of Cisco 6PE/VPE:

- Minimal operational cost and risk—No impact on existing IPv4 and MPLS services.
- Provider edge routers upgrade only—A 6PE/VPE router can be an existing PE router or a new one dedicated to IPv6 traffic.
- No impact on IPv6 customer edge routers—The ISP can connect to any customer CE running Static, IGP or EGP.
- Production services ready—An ISP can delegate IPv6 prefixes.

- IPv6 introduction into an existing MPLS service—6PE/VPE routers can be added at any time.

## IPv6 on the Provider Edge and Customer Edge Routers

### Service Provider Edge Routers

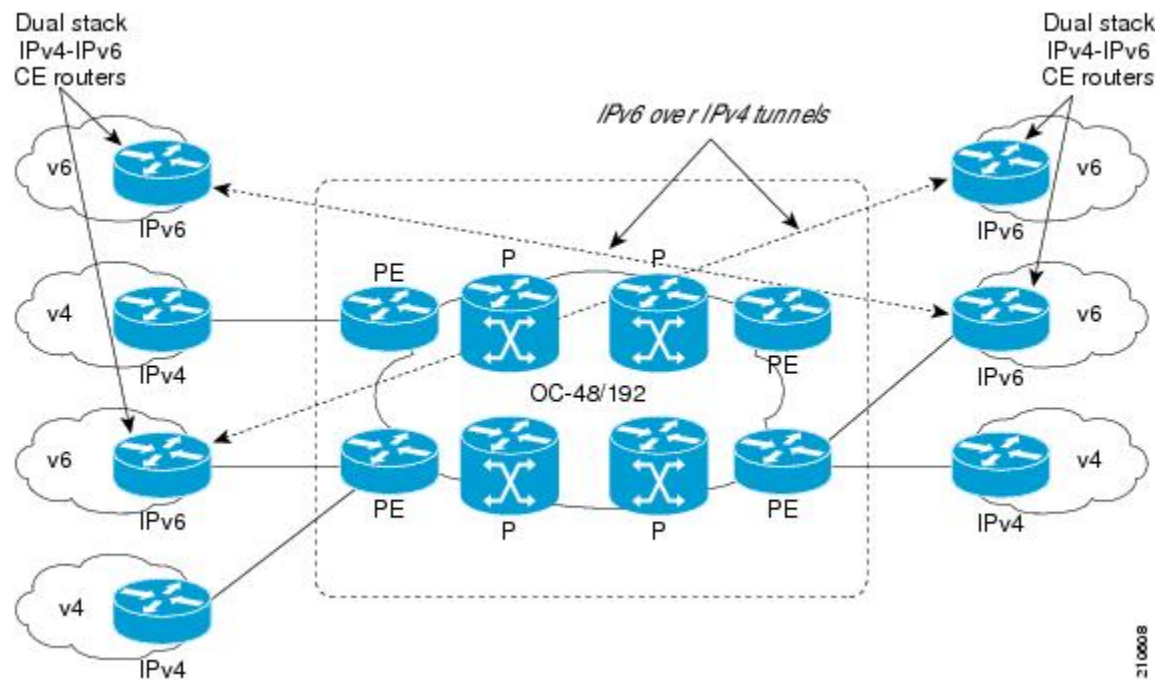
6PE is particularly applicable to service providers who currently run an MPLS network. One of its advantages is that there is no need to upgrade the hardware, software, or configuration of the core network, and it eliminates the impact on the operations and the revenues generated by the existing IPv4 traffic. MPLS is used by many service providers to deliver services to customers. MPLS as a multiservice infrastructure technology is able to provide layer 3 VPN, QoS, traffic engineering, fast re-routing and integration of ATM and IP switching.

### Customer Edge Routers

Using tunnels on the CE routers is the simplest way to deploy IPv6 over MPLS networks. It has no impact on the operation or infrastructure of MPLS and requires no changes to the P routers in the core or to the PE routers. However, tunnel meshing is required as the number of CEs to connect increases, and it is difficult to delegate a global IPv6 prefix for an ISP.

The following figure illustrates the network architecture using tunnels on the CE routers.

**Figure 1: IPv6 Using Tunnels on the CE Routers**



## IPv6 Provider Edge Multipath

Internal and external BGP multipath for IPv6 allows the IPv6 router to load balance between several paths (for example, same neighboring autonomous system (AS) or sub-AS, or the same metric) to reach its destination. The 6PE multipath feature uses multiprotocol internal BGP (MP-IBGP) to distribute IPv6 routes over the MPLS IPv4 core network and to attach an MPLS label to each route.

When MP-IBGP multipath is enabled on the 6PE router, all labeled paths are installed in the forwarding table with MPLS information (label stack) when MPLS information is available. This functionality enables 6PE to perform load balancing.

## OSPFv3 6VPE

The Open Shortest Path First version 3 (OSPFv3) IPv6 VPN Provider Edge (6VPE) feature adds VPN routing and forwarding (VRF) and provider edge-to-customer edge (PE-CE) routing support to Cisco IOS XR OSPFv3 implementation. This feature allows:

- Multiple VRF support per OSPFv3 routing process
- OSPFv3 PE-CE extensions

### Multiple VRF Support

OSPFv3 supports multiple VRFs in a single routing process that allows scaling to tens and hundreds of VRFs without consuming too much route processor (RP) resources.

Multiple OSPFv3 processes can be configured on a single router. In large-scale VRF deployments, this allows partition VRF processing across multiple RPs. It is also used to isolate default routing table or high impact VRFs from the regular VRFs. It is recommended to use a single process for all the VRFs. If needed, a second OSPFv3 process must be configured for IPv6 routing.




---

**Note** The maximum of four OSPFv3 processes are supported.

---

### OSPFv3 PE-CE Extensions

IPv6 protocol is being vastly deployed in today's customer networks. Service Providers (SPs) need to be able to offer Virtual Private Network (VPN) services to their customers for supporting IPv6 protocol, in addition to the already offered VPN services for IPv4 protocol.

In order to support IPv6, routing protocols require additional extensions for operating in the VPN environment. Extensions to OSPFv3 are required in order for OSPFv3 to operate at the PE-CE links.

### VRF Lite

VRF lite feature enables VRF deployment without BGP or MPLS based backbone. In VRF lite, the PE routers are directly connected using VRF interfaces. For OSPFv3, the following needs to operate differently in the VRF lite scenario, as opposed to the deployment with BGP or MPLS backbone:

- DN bit processing—In VRF lite environment, the DN bit processing is disabled.
- ABR status—In VRF context (except default VRF), OSPFv3 router is automatically set as an ABR, regardless to its connectivity to area 0. This automatic ABR status setting is disabled in the VRF lite environment.




---

**Note** To enable VRF Lite, issue the **capability vrf-lite** command in the OSPFv3 VRF configuration submode.

---

# How to Implement 6PE/VPE

This section includes these implementation procedures:

## Configuring 6PE/VPE

This task describes how to configure 6PE/VPE on PE routers to transport the IPv6 prefixes across the IPv4 cloud.

Ensure that you configure 6PE/VPE on PE routers participating in both the IPv4 cloud and IPv6 clouds.



---

**Note** For 6PE, you can use all routing protocols supported on Cisco IOS XR software such as BGP, OSPF, IS-IS, EIGRP, RIP, and Static to learn routes from both clouds. However, for 6VPE, you can use only the BGP, EIGRP and Static routing protocols to learn routes. Also, 6VPE supports OSPFv3 routing protocol between PE and CE routers.

---

### SUMMARY STEPS

1. **configure**
2. **router bgp** *as-number*
3. **neighbor** *ip-address*
4. **remote-as** *as-number*
5. **address-family ipv6** **labeled-unicast**
6. **exit**
7. **exit**
8. **address-family ipv6** **unicast**
9. **allocate-label** [**all** | **route-policy** *policy\_name*]
10. Use the **commit** or **end** command.

### DETAILED STEPS

---

**Step 1** **configure**

**Example:**

```
RP/0/RSP0/CPU0:router# configure
```

Enters the Global Configuration mode.

**Step 2** **router bgp** *as-number*

**Example:**

```
RP/0/RSP0/CPU0:router(config)# router bgp 1
```

Enters the number that identifies the autonomous system (AS) in which the router resides.

Range for 2-byte numbers is 1 to 65535. Range for 4-byte numbers is 1.0 to 65535.65535.

**Step 3** **neighbor** *ip-address*

**Example:**

```
RP/0/RSP0/CPU0:router(config-bgp) # neighbor 10.0.0.1
```

Enters neighbor configuration mode for configuring Border Gateway Protocol (BGP) routing sessions.

**Step 4** **remote-as** *as-number*

**Example:**

```
RP/0/RSP0/CPU0:router(config-bgp-nbr) # remote-as 100
```

Creates a neighbor and assigns a remote autonomous system number to it.

**Step 5** **address-family ipv6 labeled-unicast**

**Example:**

```
RP/0/RSP0/CPU0:router(config-bgp-nbr) # address-family ipv6 labeled-unicast
```

Specifies IPv6 labeled-unicast address prefixes.

**Note** This option is also available in IPv6 neighbor configuration mode and VRF neighbor configuration mode.

**Step 6** **exit**

**Example:**

```
RP/0/RSP0/CPU0:router(config-bgp-nbr-af) # exit
```

Exits BGP address-family submode.

**Step 7** **exit**

**Example:**

```
RP/0/RSP0/CPU0:router(config-bgp-nbr) # exit
```

Exits BGP neighbor submode.

**Step 8** **address-family ipv6 unicast**

**Example:**

```
RP/0/RSP0/CPU0:router(config-bgp) # address-family ipv6 unicast
```

Specifies IPv6 unicast address prefixes.

**Step 9** **allocate-label** [**all** | **route-policy** *policy\_name*]

**Example:**

```
RP/0/RSP0/CPU0:router(config-bgp-af)# allocate-label all
```

Allocates MPLS labels for specified IPv4 unicast routes.

**Note** The **route-policy** keyword provides finer control to filter out certain routes from being advertised to the neighbor.

**Step 10** Use the **commit** or **end** command.

**commit** - Saves the configuration changes and remains within the configuration session.

**end** - Prompts user to take one of these actions:

- **Yes** - Saves configuration changes and exits the configuration session.
- **No** - Exits the configuration session without committing the configuration changes.
- **Cancel** - Remains in the configuration mode, without committing the configuration changes.

## Configuring PE to PE Core

This task describes how to configure a Provider Edge (PE) to PE Core.

For information on configuring VPN Routing and Forwarding (VRF), refer to the *Implementing BGP* module of the *Routing Configuration Guide for Cisco ASR 9000 Series Routers*.

### SUMMARY STEPS

1. **configure**
2. **router bgp**
3. **address-family vpnv6 unicast**
4. **bgp dampening** [ *half-life* [ *reuse suppress max-suppress-time* ] ] **route-policy** *route-policy-name* ]
5. **bgp client-to-client reflection** { **cluster-id** | **disable** }
6. **neighbor** *ip-address*
7. **remote-as** *as-number*
8. **description** *text*
9. **password** { **clear** | **encrypted** } *password*
10. **shutdown**
11. **timers** *keepalive hold-time*
12. **update-source type** *interface-id*
13. **address-family vpnv6 unicast**
14. **route-policy** *route-policy-name* { **in** | **out** }
15. **exit**
16. **vrf** *vrf-name*
17. **rd** { *as-number : nn* | *ip-address : nn* | **auto** }
18. Use the **commit** or **end** command.

## DETAILED STEPS

---

### Step 1 **configure**

**Example:**

```
RP/0/RSP0/CPU0:router# configure
```

Enters the Global Configuration mode.

### Step 2 **router bgp**

**Example:**

```
RP/0/RSP0/CPU0:router(config)# router bgp 10
```

Specifies the BGP AS number and enters the BGP configuration mode, allowing you to configure the BGP routing process.

### Step 3 **address-family vpnv6 unicast**

**Example:**

```
RP/0/RSP0/CPU0:router(config-bgp)# address-family vpnv6 unicast
```

Specifies the vpnv6 address family and enters address family configuration submode.

### Step 4 **bgp dampening [ half-life [ reuse suppress max-suppress-time ] | route-policy route-policy-name ]**

**Example:**

```
RP/0/RSP0/CPU0:router(config-bgp-af)# bgp dampening 30 1500 10000 120
```

Configures BGP dampening for the specified address family.

### Step 5 **bgp client-to-client reflection { cluster-id | disable }**

**Example:**

```
RP/0/RSP0/CPU0:router(config-bgp-af)# bgp client-to-client  
reflection disable
```

Configures client to client route reflection.

### Step 6 **neighbor ip-address**

**Example:**

```
RP/0/RSP0/CPU0:router(config-bgp)# neighbor 10.0.0.1
```

Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address as a BGP peer.

### Step 7 **remote-as as-number**



**Example:**

```
RP/0/RSP0/CPU0:router(config-bgp-nbr)# remote-as 100
```

Creates a neighbor and assigns a remote autonomous system number to it.

**Step 8****description** *text***Example:**

```
RP/0/RSP0/CPU0:router(config-bgp-nbr)# description neighbor 172.16.1.1
```

Provides a description of the neighbor. The description is used to save comments and does not affect software function.

**Step 9****password** { **clear** | **encrypted** } *password***Example:**

```
RP/0/RSP0/CPU0:router(config-bgp-nbr)# password encrypted 123abc
```

Enables Message Digest 5 (MD5) authentication on the TCP connection between the two BGP neighbors.

**Step 10****shutdown****Example:**

```
RP/0/RSP0/CPU0:router(config-bgp-nbr)# router bgp 1
```

Terminates any active sessions for the specified neighbor and removes all associated routing information.

**Step 11****timers** *keepalive hold-time***Example:**

```
RP/0/RSP0/CPU0:router(config-bgp-nbr)# timers 12000 200
```

Set the timers for the BGP neighbor.

**Step 12****update-source type** *interface-id***Example:**

```
RP/0/RSP0/CPU0:router(config-bgp-nbr)# update-source TenGigE 0/1/5/0
```

Allows iBGP sessions to use the primary IP address from a specific interface as the local address when forming an iBGP session with a neighbor.

**Step 13****address-family vpnv6 unicast****Example:**

```
RP/0/RSP0/CPU0:router(config-bgp-nbr)# address-family vpnv6 unicast
```

Enters VPN neighbor address family configuration mode.

**Step 14** `route-policy route-policy-name { in | out }`

**Example:**

```
RP/0/RSP0/CPU0:router(config-bgp-nbr-af)# route-policy pe-pe-vpn-out out
```

Specifies a routing policy for an outbound route. The policy can be used to filter routes or modify route attributes.

**Step 15** `exit`

**Example:**

```
RP/0/RSP0/CPU0:router(config-bgp-nbr-af)# exit
```

Exits address family configuration and neighbor submode.

**Step 16** `vrf vrf-name`

**Example:**

```
RP/0/RSP0/CPU0:router(config-bgp)# vrf vrf-pe
```

Configures a VRF instance.

**Step 17** `rd { as-number : nn | ip-address : nn | auto }`

**Example:**

```
RP/0/RSP0/CPU0:router(config-bgp-vrf)# rd 345:567
```

Configures the route distinguisher.

Use the `auto` keyword if you want the router to automatically assign a unique RD to the VRF.

**Step 18** Use the `commit` or `end` command.

**commit** - Saves the configuration changes and remains within the configuration session.

**end** - Prompts user to take one of these actions:

- **Yes** - Saves configuration changes and exits the configuration session.
- **No** - Exits the configuration session without committing the configuration changes.
- **Cancel** - Remains in the configuration mode, without committing the configuration changes.

## Configuring OSPFv3 as the Routing Protocol Between the PE and CE Routers

Perform this task to configure provider edge (PE)-to-customer edge (CE) routing sessions that use Open Shortest Path First version 3 (OSPFv3).

### SUMMARY STEPS

1. `configure`
2. `router ospfv3 process-name`

3. **vrf** *vrf-name*
4. **capability** *vrf-lite*
5. **router-id** {*router-id* | *type interface-path-id*}
6. **domain-id type** { 0005 | 0105 | 0205 | 8005 } **value** *domain-id*
7. Do one of the following:
  - **redistribute bgp** *process-id* [ **metric** *metric-value* ] [ **metric-type** {1 | 2} ] [ **route-policy**1**route-policy** *policy-name* ] [ **tag** *tag-value* ]
  - **redistribute connected** [ **metric** *metric-value* ] [ **metric-type** {1 | 2} ] [ **route-policy***policy-name* ] [ **tag** *tag-value* ]
  - **redistribute ospf** *process-id* [ **match** {**external** [1 | 2] | **internal** | **nssa-external** [1 | 2]} ] [ **metric** *metric-value* ] [ **metric-type** {1 | 2} ] [ **route-policy** *policy-name* ] [ **tag** *tag-value* ]
  - **redistribute static** [ **metric** *metric-value* ] [ **metric-type** {1 | 2} ] [ **route-policy** *policy-name* ] [ **tag** *policy-name* ] [ **tag** *tag-value* ]
  - **redistribute eigrp** *process-id* [ **match** {**external** [1 | 2] | **internal** | **nssa-external** [1 | 2]} ] [ **metric** *metric-value* ] [ **metric-type** {1 | 2} ] [ **route-policy** *policy-name* ] [ **tag** *tag-value* ]
  - **redistribute rip** [ **metric** *metric-value* ] [ **metric-type** {1 | 2} ] [ **route-policy** *policy-name* ] [ **tag** *tag-value* ]
8. **area** *area-id*
9. **interface** {*type interface-path-id*}
10. Use the **commit** or **end** command.

## DETAILED STEPS

### Step 1 **configure**

**Example:**

```
RP/0/RSP0/CPU0:router# configure
```

Enters the Global Configuration mode.

### Step 2 **router ospfv3** *process-name*

**Example:**

```
RP/0/RSP0/CPU0:router(config)# router ospfv3 109
```

Enters OSPF configuration mode allowing you to configure the OSPF version 3 routing process.

### Step 3 **vrf** *vrf-name*

**Example:**

```
RP/0/RSP0/CPU0:router(config-ospf)# vrf vrf_1
```

Configures a VPN routing and forwarding (VRF) instance and enters VRF configuration mode for OSPF routing.

### Step 4 **capability** *vrf-lite*

**Example:**

```
RP/0/RSP0/CPU0:router(config-ospf-vrf)# capability vrf-lite
```

Enables VRF Lite feature.

**Step 5** `router-id {router-id | type interface-path-id }`

**Example:**

```
RP/0/RSP0/CPU0:router(config-ospf-vrf)# router-id 172.20.10.10
```

Configures the router ID for the OSPF routing process.

**Note** Router ID configuration is required for each VRF.

**Step 6** `domain-id type { 0005 | 0105 | 0205 | 8005 } value domain-id`

**Example:**

```
RP/0/RSP0/CPU0:router(config-ospf-vrf)# domain-id type 0005 value CAFE00112233
```

Specifies the domain ID.

**Step 7** Do one of the following:

- `redistribute bgp process-id [ metric metric-value ] [ metric-type {1 | 2} ] [ route-policy12route-policy policy-name ] [ tag tag-value ]`
- `redistribute connected [metric metric-value ] [ metric-type {1 | 2} ] [route-policy policy-name ] [ tag tag-value ]`
- `redistribute ospf process-id [ match {external [1 | 2] | internal | nssa-external [1 | 2]} ] [metric metric-value ] [ metric-type {1 | 2} ] [route-policy policy-name ] [ tag tag-value ]`
- `redistribute static [metric metric-value ] [ metric-type {1 | 2} ] [route-policy policy-name] [ tag policy-name ] [ tag tag-value ]`
- `redistribute eigrp process-id [match {external [1 | 2] | internal | nssa-external [1 | 2]} ] [metric metric-value ] [metric-type {1 | 2} ] [route-policy policy-name ] [ tag tag-value ]`
- `redistribute rip [ metric metric-value ] [ metric-type {1 | 2} ] [route-policy policy-name] [tag tag-value ]`

**Example:**

```
RP/0/RSP0/CPU0:router(config-ospf-vrf)# redistribute connected
```

Causes routes to be redistributed into OSPF. The routes that can be redistributed into OSPF are:

- Border Gateway Protocol (BGP)
- Connected
- Enhanced Interior Gateway Routing Protocol (EIGRP)
- OSPF
- Static
- Routing Information Protocol (RIP)

**Step 8** `area area-id`

**Example:**

```
RP/0/RSP0/CPU0:router(config-ospf-vrf)# area 0
```

Configures the OSPF area as area 0.

**Step 9** `interface {type interface-path-id}`

**Example:**

```
RP/0/RSP0/CPU0:router(config-ospf-vrf-ar)# interface GigabitEthernet 0/3/0/0
```

Associates interface GigabitEthernet 0/3/0/0 with area 0.

**Step 10** Use the **commit** or **end** command.

**commit** - Saves the configuration changes and remains within the configuration session.

**end** - Prompts user to take one of these actions:

- **Yes** - Saves configuration changes and exits the configuration session.
- **No** - Exits the configuration session without committing the configuration changes.
- **Cancel** - Remains in the configuration mode, without committing the configuration changes.

## Configuration Examples for 6PE/VPE

This section includes the following configuration example:

### Configuring 6PE on a PE Router: Example

This sample configuration shows the configuration of 6PE on a PE router:

```
interface TenGigE0/3/0/0
  ipv6 address 2001::1/64
  !
router isis ipv6-cloud
  net 49.0000.0000.0001.00
  address-family ipv6 unicast
    single-topology
  interface TenGigE0/3/0/0
    address-family ipv6 unicast
  !
!
router bgp 55400
  bgp router-id 54.6.1.1
  address-family ipv4 unicast
  !
  address-family ipv6 unicast
    network 55:5::/64
    redistribute connected
    redistribute isis ipv6-cloud
    allocate-label all
  !
neighbor 34.4.3.3
  remote-as 55400
  address-family ipv4 unicast
```

```
!
address-family ipv6 labeled-unicast
```

## Configuring 6VPE on a PE Router: Example

This sample configuration shows the configuration of 6VPE on a PE router:

```
vrf vpn1
address-family ipv6 unicast
import route-target
200:2
!
export route-target
200:2

interface Loopback0
ipv4 address 10.0.0.1 255.255.255.255

interface GigabitEthernet0/0/0/1
vrf vpn1
ipv6 address 2001:c003:a::2/64

router bgp 1
bgp router-id 10.0.0.1
bgp redistribute-internal
bgp graceful-restart
address-family ipv4 unicast
!

address-family vpnv6 unicast
!
neighbor 10.0.0.2 >>>> Remote peer loopback address.
remote-as 1
update-source Loopback0
address-family ipv4 unicast
!
address-family vpnv6 unicast
route-policy pass-all in
route-policy pass-all out
!

vrf vpn1
rd 100:2
bgp router-id 140.140.140.140
address-family ipv6 unicast
redistribute connected
!

neighbor 2001:c003:a::1
remote-as 6502
address-family ipv6 unicast
route-policy pass-all in
route-policy pass-all out
!
```