



# Configuring Virtual Private LAN Service (VPLS) and VPLS BGP-Based Autodiscovery

---

- [Finding Feature Information, on page 1](#)
- [Configuring VPLS, on page 1](#)
- [Configuring VPLS BGP-based Autodiscovery, on page 13](#)

## 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 at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <https://cfng.cisco.com/>. An account on Cisco.com is not required.

## Configuring VPLS

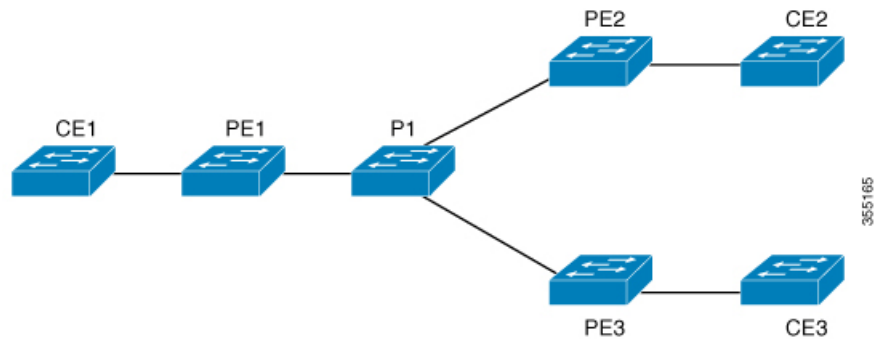
### Information About VPLS

#### VPLS Overview

VPLS (Virtual Private LAN Service) enables enterprises to link together their Ethernet-based LANs from multiple sites via the infrastructure provided by their service provider. From the enterprise perspective, the service provider's public network looks like one giant Ethernet LAN. For the service provider, VPLS provides an opportunity to deploy another revenue-generating service on top of their existing network without major capital expenditures. Operators can extend the operational life of equipment in their network.

Virtual Private LAN Services (VPLS) uses the provider core to join multiple attachment circuits together to simulate a virtual bridge that connects the multiple attachment circuits together. From a customer point of view, there is no topology for VPLS. All of the CE devices appear to connect to a logical bridge emulated by the provider core.

Figure 1: VPLS Topology



### Full-Mesh Configuration

The full-mesh configuration requires a full mesh of tunnel label switched paths (LSPs) between all the PEs that participate in the VPLS. With full-mesh, signaling overhead and packet replication requirements for each provisioned VC on a PE can be high.

You set up a VPLS by first creating a virtual forwarding instance (VFI) on each participating PE router. The VFI specifies the VPN ID of a VPLS domain, the addresses of other PE devices in the domain, and the type of tunnel signaling and encapsulation mechanism for each peer PE router.

The set of VFIs formed by the interconnection of the emulated VCs is called a VPLS instance; it is the VPLS instance that forms the logic bridge over a packet switched network. The VPLS instance is assigned a unique VPN ID.

The PE devices use the VFI to establish a full-mesh LSP of emulated VCs to all the other PE devices in the VPLS instance. PE devices obtain the membership of a VPLS instance through static configuration using the Cisco IOS CLI.

The full-mesh configuration allows the PE router to maintain a single broadcast domain. Thus, when the PE router receives a broadcast, multicast, or unknown unicast packet on an attachment circuit, it sends the packet out on all other attachment circuits and emulated circuits to all other CE devices participating in that VPLS instance. The CE devices see the VPLS instance as an emulated LAN.

To avoid the problem of a packet looping in the provider core, the PE devices enforce a "split-horizon" principle for the emulated VCs. That means if a packet is received on an emulated VC, it is not forwarded on any other emulated VC.

After the VFI has been defined, it needs to be bound to an attachment circuit to the CE device.

The packet forwarding decision is made by looking up the Layer 2 virtual forwarding instance (VFI) of a particular VPLS domain.

A VPLS instance on a particular PE router receives Ethernet frames that enter on specific physical or logical ports and populates a MAC table similarly to how an Ethernet switch works. The PE router can use the MAC address to switch those frames into the appropriate LSP for delivery to the another PE router at a remote site.

If the MAC address is not in the MAC address table, the PE router replicates the Ethernet frame and floods it to all logical ports associated with that VPLS instance, except the ingress port where it just entered. The PE router updates the MAC table as it receives packets on specific ports and removes addresses not used for specific periods.

### VPLS BGP Based Autodiscovery

VPLS Autodiscovery enables each Virtual Private LAN Service (VPLS) provider edge (PE) device to discover other PE devices that are part of the same VPLS domain. VPLS Autodiscovery also tracks PE devices when they are added to or removed from a VPLS domain. As a result, with VPLS Autodiscovery enabled, you no longer need to manually configure a VPLS domain and maintain the configuration when a PE device is added or deleted. VPLS Autodiscovery uses the Border Gateway Protocol (BGP) to discover VPLS members and set up and tear down pseudowires in a VPLS domain.

BGP uses the Layer 2 VPN (L2VPN) Routing Information Base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 virtual forwarding instance (VFI) is configured. The prefix and path information is stored in the L2VPN database, which allows BGP to make decisions about the best path. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, this endpoint information is used to configure a pseudowire mesh to support L2VPN-based services.

The BGP autodiscovery mechanism facilitates the configuration of L2VPN services, which are an integral part of the VPLS feature. VPLS enables flexibility in deploying services by connecting geographically dispersed sites as a large LAN over high-speed Ethernet in a robust and scalable IP Multiprotocol Label Switching (MPLS) network.

For scale information related to this feature, see [Cisco Catalyst 9500 Series Switches Data Sheet](#).

## Restrictions for VPLS

- Protocol-based CLI Method (interface pseudowire configuration) is not supported. Only VFI and Xconnect mode are supported.
- Flow-Aware Transport Pseudowire (FAT PW) is not supported.
- IGMP Snooping is not Supported. Multicast traffic floods with IGMP Snooping disabled.
- L2 Protocol Tunneling is not supported.
- Integrated Routing and Bridging (IRB) not supported.
- Virtual Circuit Connectivity Verification (VCCV) ping with explicit null is not supported.
- The switch is supported only as spoke in H-VPLS but not as hub.
- L2 VPN Interworking is not supported.
- **ip unnumbered** command is not supported in MPLS configuration.
- VC statistics are not displayed for flood traffic in the output of show mpls l2 vc vcid detail command.
- dot1q tunnel is not supported in the attachment circuit.

## Configuring PE Layer 2 Interfaces to CEs

### Configuring 802.1Q Trunks for Tagged Traffic from a CE

#### SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface** *interface-id*
4. **no ip address** *ip\_address mask* [secondary ]
5. **switchport**
6. **switchport trunk encapsulation dot1q**
7. **switchport trunk allow vlan** *vlan\_ID*
8. **switchport mode trunk**
9. **end**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>interface</b> <i>interface-id</i> <b>Example:</b> Device(config)# <b>interface</b> TenGigabitEthernet1/0/24	Defines the interface to be configured as a trunk, and enters interface configuration mode.
<b>Step 4</b>	<b>no ip address</b> <i>ip_address mask</i> [secondary ] <b>Example:</b> Device(config-if)# <b>no ip address</b>	Disables IP processing and enters interface configuration mode.
<b>Step 5</b>	<b>switchport</b> <b>Example:</b> Device(config-if)# <b>switchport</b>	Modifies the switching characteristics of the Layer 2-switched interface.
<b>Step 6</b>	<b>switchport trunk encapsulation dot1q</b> <b>Example:</b> Device(config-if)# <b>switchport trunk encapsulation dot1q</b>	Sets the switch port encapsulation format to 802.1Q.
<b>Step 7</b>	<b>switchport trunk allow vlan</b> <i>vlan_ID</i> <b>Example:</b>	Sets the list of allowed VLANs.

	Command or Action	Purpose
	Device(config-if)# <b>switchport trunk allow vlan 2129</b>	
<b>Step 8</b>	<b>switchport mode trunk</b> <b>Example:</b> Device(config-if)# <b>switchport mode trunk</b>	Sets the interface to a trunking VLAN Layer 2 interface.
<b>Step 9</b>	<b>end</b> <b>Example:</b> Device(config)# <b>end</b>	Returns to privileged EXEC mode.

## Configuring 802.1Q Access Ports for Untagged Traffic from a CE

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface *interface-id***
4. **no ip address *ip\_address mask* [secondary ]**
5. **switchport**
6. **switchport mode access**
7. **switchport access vlan *vlan\_ID***
8. **end**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. <ul style="list-style-type: none"><li>• Enter your password if prompted.</li></ul>
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>interface <i>interface-id</i></b> <b>Example:</b>	Defines the interface to be configured as a trunk, and enters interface configuration mode.

	Command or Action	Purpose
	Device(config)# <b>interface TenGigabitEthernet1/0/24</b>	
<b>Step 4</b>	<b>no ip address</b> <i>ip_address mask</i> [secondary ] <b>Example:</b> Device(config-if)# <b>no ip address</b>	Disables IP processing and enters interface configuration mode.
<b>Step 5</b>	<b>switchport</b> <b>Example:</b> Device(config-if)# <b>switchport</b>	Modifies the switching characteristics of the Layer 2-switched interface.
<b>Step 6</b>	<b>switchport mode access</b> <b>Example:</b> Device(config-if)# <b>switchport mode access</b>	Sets the interface type to nontrunking, nontagged single VLAN Layer 2 interface.
<b>Step 7</b>	<b>switchport access vlan</b> <i>vlan_ID</i> <b>Example:</b> Device(config-if)# <b>switchport access vlan 2129</b>	Sets the VLAN when the interface is in access mode.
<b>Step 8</b>	<b>end</b> <b>Example:</b> Device(config)# <b>end</b>	Returns to privileged EXEC mode.

## Configuring Layer 2 VLAN Instances on a PE

Configuring the Layer 2 VLAN interface on the PE enables the Layer 2 VLAN instance on the PE router to the VLAN database to set up the mapping between the VPLS and VLANs.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vlan** *vlan-id*
4. **interface vlan** *vlan-id*
5. **end**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
Step 3	<b>vlan <i>vlan-id</i></b> <b>Example:</b> Device(config)# <b>vlan 2129</b>	Configures a specific virtual LAN (VLAN).
Step 4	<b>interface vlan <i>vlan-id</i></b> <b>Example:</b> Device(config-vlan)# <b>interface vlan 2129</b>	Configures an interface on the VLAN.
Step 5	<b>end</b> <b>Example:</b> Device(config)# <b>end</b>	Returns to privileged EXEC mode.

## Configuring MPLS in the PE

To configure MPLS in the PE, you must provide the required MPLS parameters.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ip**
4. **mpls label protocol ldp**
5. **end**
6. **mpls ldp logging neighbor-changes**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>	Enables privileged EXEC mode.

	Command or Action	Purpose
	<b>Example:</b>  Device> <b>enable</b>	<ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b>  Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>mpls ip</b>  <b>Example:</b>  Device(config)# <b>mpls ip</b>	Configures MPLS hop-by-hop forwarding.
<b>Step 4</b>	<b>mpls label protocol ldp</b>  <b>Example:</b>  Device(config-vlan)# <b>mpls label protocol ldp</b>	Specifies the default Label Distribution Protocol for a platform.
<b>Step 5</b>	<b>end</b>  <b>Example:</b>  Device(config)# <b>end</b>	Returns to privileged EXEC mode.
<b>Step 6</b>	<b>mpls ldp logging neighbor-changes</b>  <b>Example:</b>  Device(config)# <b>mpls ldp logging neighbor-changes</b>	(Optional) Determines logging neighbor changes.

## Configuring VFI in the PE

The virtual switch instance (VFI) specifies the VPN ID of a VPLS domain, the addresses of other PE devices in this domain, and the type of tunnel signaling and encapsulation mechanism for each peer (This is where you create the VFI and associated VCs.). Configure a VFI as follows:

### SUMMARY STEPS

- enable**
- configure terminal**
- l2 vfi vfi-name manual**
- vpn id vpn-id**
- neighbor router-id {encapsulation mpls}**
- end**



## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
Step 3	<b>l2 vfi vfi-name manual</b> <b>Example:</b> Device(config)# <b>l2 vfi 2129 manual</b>	Enables the Layer 2 VFI manual configuration mode.
Step 4	<b>vpn id vpn-id</b> <b>Example:</b> Device(config-vfi)# <b>vpn id 2129</b>	Configures a VPN ID for a VPLS domain. The emulated VCs bound to this Layer 2 VRF use this VPN ID for signaling. <b>Note</b> <i>vpn-id</i> is the same as <i>vlan-id</i> .
Step 5	<b>neighbor router-id {encapsulation mpls}</b> <b>Example:</b> Device(config-vfi)# <b>neighbor remote-router-id encapsulation mpls</b>	Specifies the remote peering router ID and the tunnel encapsulation type or the pseudo-wire property to be used to set up the emulated VC.
Step 6	<b>end</b> <b>Example:</b> Device(config)# <b>end</b>	Returns to privileged EXEC mode.

## Associating the Attachment Circuit with the VFI at the PE

After defining the VFI, you must bind it to one or more attachment circuits.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface vlan vlan-id**
4. **no ip address**
5. **xconnect vfi vfi-name**

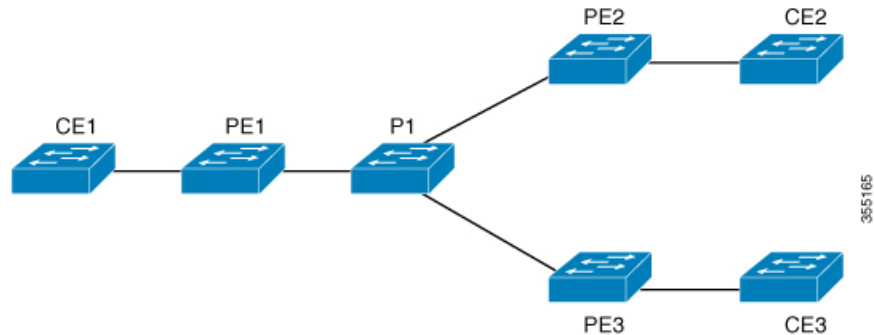
## 6. end

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>interface vlan <i>vlan-id</i></b> <b>Example:</b> Device(config)# <b>interface vlan 2129</b>	Creates or accesses a dynamic switched virtual interface (SVI). <b>Note</b> <i>vlan-id</i> is the same as <i>vpn-id</i> .
<b>Step 4</b>	<b>no ip address</b> <b>Example:</b> Device(config-if)# <b>no ip address</b>	Disables IP processing. (You configure a Layer 3 interface for the VLAN if you configure an IP address.)
<b>Step 5</b>	<b>xconnect vfi <i>vfi-name</i></b> <b>Example:</b> Device(config-if)# <b>xconnect vfi 2129</b>	Specifies the Layer 2 VFI that you are binding to the VLAN port.
<b>Step 6</b>	<b>end</b> <b>Example:</b> Device(config)# <b>end</b>	Returns to privileged EXEC mode.

# Configuration Examples for VPLS

Figure 2: VPLS Topology



PE1 Configuration	PE2 Configuration
<pre> pseudowire-class vpls2129  encapsulation mpls  !  l2 vfi 2129 manual   vpn id 2129   neighbor 44.254.44.44 pw-class vpls2129  !  neighbor 188.98.89.98 pw-class vpls2129  !  interface TenGigabitEthernet1/0/24   switchport trunk allowed vlan 2129   switchport mode trunk  !  interface Vlan2129   no ip address   xconnect vfi 2129  !           </pre>	<pre> pseudowire-class vpls2129  encapsulation mpls  no control-word  !  l2 vfi 2129 manual   vpn id 2129   neighbor 1.1.1.72 pw-class vpls2129   neighbor 188.98.89.98 pw-class vpls2129  !  interface TenGigabitEthernet1/0/47   switchport trunk allowed vlan 2129   switchport mode trunk  end  !  interface Vlan2129   no ip address   xconnect vfi 2129  !           </pre>

The **show mpls 12transport vc detail** command provides information the virtual circuits.

```

Local interface: VFI 2129 vfi up
  Interworking type is Ethernet
  Destination address: 44.254.44.44, VC ID: 2129, VC status: up
  Output interface: Gi1/0/9, imposed label stack {18 17}
  Preferred path: not configured
  Default path: active
  Next hop: 177.77.177.2
  Create time: 19:09:33, last status change time: 09:24:14
  Last label FSM state change time: 09:24:14
  Signaling protocol: LDP, peer 44.254.44.44:0 up
  Targeted Hello: 1.1.1.72 (LDP Id) -> 44.254.44.44, LDP is UP
  Graceful restart: configured and enabled
  Non stop routing: not configured and not enabled
  Status TLV support (local/remote) : enabled/supported
  LDP route watch : enabled
    
```

```

Label/status state machine      : established, LruRru
Last local dataplane status rcvd: No fault
Last BFD dataplane status rcvd: Not sent
Last BFD peer monitor status rcvd: No fault
Last local AC circuit status rcvd: No fault
Last local AC circuit status sent: No fault
Last local PW i/f circ status rcvd: No fault
Last local LDP TLV status sent: No fault
Last remote LDP TLV status rcvd: No fault
Last remote LDP ADJ status rcvd: No fault
MPLS VC labels: local 512, remote 17
Group ID: local n/a, remote 0
MTU: local 1500, remote 1500
Remote interface description:
Sequencing: receive disabled, send disabled
Control Word: Off
SSO Descriptor: 44.254.44.44/2129, local label: 512
Dataplane:
SSM segment/switch IDs: 20498/20492 (used), PWID: 2
VC statistics:
transit packet totals: receive 0, send 0
transit byte totals: receive 0, send 0
transit packet drops: receive 0, seq error 0, send 0

```

The **show l2vpn atom vc** shows that ATM over MPLS is configured on a VC.

```

pseudowire100005 is up, VC status is up PW type: Ethernet
Create time: 19:25:56, last status change time: 09:40:37
Last label FSM state change time: 09:40:37
Destination address: 44.254.44.44 VC ID: 2129
Output interface: Gi1/0/9, imposed label stack {18 17}
Preferred path: not configured
Default path: active
Next hop: 177.77.177.2
Member of vfi service 2129
Bridge-Domain id: 2129
Service id: 0x32000003
Signaling protocol: LDP, peer 44.254.44.44:0 up
Targeted Hello: 1.1.1.72(LDP Id) -> 44.254.44.44, LDP is UP
Graceful restart: configured and enabled
Non stop routing: not configured and not enabled
PWid FEC (128), VC ID: 2129
Status TLV support (local/remote)      : enabled/supported
LDP route watch                        : enabled
Label/status state machine              : established, LruRru
Local dataplane status received         : No fault
BFD dataplane status received          : Not sent
BFD peer monitor status received       : No fault
Status received from access circuit     : No fault
Status sent to access circuit           : No fault
Status received from pseudowire i/f    : No fault

```

```

Status sent to network peer          : No fault
  Status received from network peer   : No fault
  Adjacency status of remote peer     : No fault
Sequencing: receive disabled, send disabled
Bindings
  Parameter      Local              Remote
  -----
Label           512                 17
Group ID        n/a                 0
Interface

MTU             1500                 1500
Control word    off                  off
PW type         Ethernet            Ethernet
VCCV CV type    0x02                 0x02
                LSPV [2]             LSPV [2]

VCCV CC type    0x06                 0x06
                RA [2], TTL [3]      RA [2], TTL [3]
Status TLV      enabled              supported
SSO Descriptor: 44.254.44.44/2129, local label: 512
Dataplane:
  SSM segment/switch IDs: 20498/20492 (used), PWID: 2
Rx Counters
  0 input transit packets, 0 bytes
  0 drops, 0 seq err
Tx Counters
  0 output transit packets, 0 bytes
  0 drops

```

## Configuring VPLS BGP-based Autodiscovery

### Information About VPLS BGP-Based Autodiscovery

#### VPLS BGP Based Autodiscovery

VPLS Autodiscovery enables each Virtual Private LAN Service (VPLS) provider edge (PE) device to discover other PE devices that are part of the same VPLS domain. VPLS Autodiscovery also tracks PE devices when they are added to or removed from a VPLS domain. As a result, with VPLS Autodiscovery enabled, you no longer need to manually configure a VPLS domain and maintain the configuration when a PE device is added or deleted. VPLS Autodiscovery uses the Border Gateway Protocol (BGP) to discover VPLS members and set up and tear down pseudowires in a VPLS domain.

BGP uses the Layer 2 VPN (L2VPN) Routing Information Base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 virtual forwarding instance (VFI) is configured. The prefix and path information is stored in the L2VPN database, which allows BGP to make decisions about the best path. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, this endpoint information is used to configure a pseudowire mesh to support L2VPN-based services.

The BGP autodiscovery mechanism facilitates the configuration of L2VPN services, which are an integral part of the VPLS feature. VPLS enables flexibility in deploying services by connecting geographically dispersed sites as a large LAN over high-speed Ethernet in a robust and scalable IP Multiprotocol Label Switching (MPLS) network.

For scale information related to this feature, see [Cisco Catalyst 9500 Series Switches Data Sheet](#).

## Enabling VPLS BGP-based Autodiscovery

Perform this task to enable Virtual Private LAN Service (VPLS) PE devices to discover other PE devices that are part of the same VPLS domain.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 vfi *vfi-name* autodiscovery**
4. **vpn id *vpn-id***
5. **end**

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
Step 3	<b>l2 vfi <i>vfi-name</i> autodiscovery</b> <b>Example:</b> Device(config)# <b>l2 vfi 2128 autodiscovery</b>	Enables VPLS Autodiscovery on a PE device and enters L2 VFI configuration mode.
Step 4	<b>vpn id <i>vpn-id</i></b> <b>Example:</b> Device(config-vfi)# <b>vpn id 2128</b>	Configures a VPN ID for the VPLS domain.
Step 5	<b>end</b> <b>Example:</b>	Returns to privileged EXEC mode.

	Command or Action	Purpose
	Device(config)# <b>end</b>	

## Configuring BGP to Enable VPLS Autodiscovery

The Border Gateway Protocol (BGP) Layer 2 VPN (L2VPN) address family supports a separate L2VPN Routing Information Base (RIB) that contains endpoint provisioning information for Virtual Private LAN Service (VPLS) Autodiscovery. BGP learns the endpoint provisioning information from the L2VPN database, which is updated each time a Layer 2 virtual forwarding instance (VFI) is configured. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to configure a pseudowire mesh to support L2VPN-based services.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **no bgp default ipv4-unicast**
5. **bgp log-neighbor-changes**
6. **neighbor remote-as** { *ip-address* | *peer-group-name* } **remote-as** *autonomous-system-number*
7. **neighbor** { *ip-address* | *peer-group-name* } **update-source** *interface-type interface-number*
8. Repeat Steps 6 and 7 to configure other BGP neighbors.
9. **address-family l2vpn** [*vpls*]
10. **neighbor** { *ip-address* | *peer-group-name* } **activate**
11. **neighbor** { *ip-address* | *peer-group-name* } **send-community** { **both** | **standard** | **extended** }
12. Repeat Steps 10 and 11 to activate other BGP neighbors under an L2VPN address family.
13. **exit-address-family**
14. **end**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.

	Command or Action	Purpose
<b>Step 3</b>	<b>router bgp</b> <i>autonomous-system-number</i> <b>Example:</b> Device(config)# <b>router bgp</b> 1000	Enters router configuration mode for the specified routing process.
<b>Step 4</b>	<b>no bgp default ipv4-unicast</b> <b>Example:</b> Device(config-router)# <b>no bgp default ipv4-unicast</b>	Disables the IPv4 unicast address family for the BGP routing process.  <b>Note</b> Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured using the neighbor remote-as router configuration command unless you configure the no bgp default ipv4-unicast router configuration command before configuring the neighbor remote-as command. Existing neighbor configurations are not affected.
<b>Step 5</b>	<b>bgp log-neighbor-changes</b> <b>Example:</b> Device(config-router)# <b>bgp log-neighbor-changes</b>	Enables logging of BGP neighbor resets.
<b>Step 6</b>	<b>neighbor remote-as</b> { <i>ip-address</i>   <i>peer-group-name</i> } <b>remote-as</b> <i>autonomous-system-number</i> <b>Example:</b> Device(config-router)# <b>neighbor 44.254.44.44</b> <b>remote-as 1000</b>	Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device. <ul style="list-style-type: none"> <li>• If the autonomous-system-number argument matches the autonomous system number specified in the router bgp command, the neighbor is an internal neighbor.</li> <li>• If the autonomous-system-number argument does not match the autonomous system number specified in the router bgp command, the neighbor is an external neighbor.</li> </ul>
<b>Step 7</b>	<b>neighbor</b> { <i>ip-address</i>   <i>peer-group-name</i> } <b>update-source</b> <i>interface-type interface-number</i> <b>Example:</b> Device(config-router)# <b>neighbor 44.254.44.44</b> <b>update-source Loopback300</b>	(Optional) Configures a device to select a specific source or interface to receive routing table updates.
<b>Step 8</b>	Repeat Steps 6 and 7 to configure other BGP neighbors.	Exits interface configuration mode.
<b>Step 9</b>	<b>address-family l2vpn [vpls]</b> <b>Example:</b>	Specifies the L2VPN address family and enters address family configuration mode.



	Command or Action	Purpose
	Device(config-router)# <b>address-family l2vpn vpls</b>	The optional <b>vpls</b> keyword specifies that the VPLS endpoint provisioning information is to be distributed to BGP peers.
<b>Step 10</b>	<b>neighbor { ip-address   peer-group-name } activate</b> <b>Example:</b> Device(config-router-af)# <b>neighbor 44.254.44.44 activate</b>	Enables the exchange of information with a BGP neighbor.
<b>Step 11</b>	<b>neighbor { ip-address   peer-group-name } send-community { both   standard   extended }</b> <b>Example:</b> Device(config-router-af)# <b>neighbor 44.254.44.44 send-community both</b>	Specifies that a communities attribute should be sent to a BGP neighbor.
<b>Step 12</b>	Repeat Steps 10 and 11 to activate other BGP neighbors under an L2VPN address family.	
<b>Step 13</b>	<b>exit-address-family</b> <b>Example:</b> Device(config-router-af)# <b>exit-address-family</b>	Exits address family configuration mode and returns to router configuration mode.
<b>Step 14</b>	<b>end</b> <b>Example:</b> Device(config-router-af)# <b>end</b>	Exits router configuration mode and returns to privileged EXEC mode.

## Configuration Examples for VPLS BGP-AD

### PE Configuration

```

router bgp 1000
  bgp log-neighbor-changes
  bgp graceful-restart
  neighbor 44.254.44.44 remote-as 1000
  neighbor 44.254.44.44 update-source Loopback300
!
  address-family l2vpn vpls
    neighbor 44.254.44.44 activate
    neighbor 44.254.44.44 send-community both
  exit-address-family
!
l2 vfi 2128 autodiscovery
  vpn id 2128
interface Vlan2128
  no ip address
  xconnect vfi 2128
!

```

The following is a sample output of **show platform software fed sw 1 matm macTable vlan 2000** command :

VLAN	MAC	Type	Seq#	macHandle	siHandle
	diHandle	*a_time	*e_time	ports	
2000	2852.6134.05c8	0X8002	0	0xffbba312c8	0xffbb9ef938
	0x5154	0	0	Vlan2000	
2000	0000.0078.9012	0X1	32627	0xffbb665ec8	0xffbb60b198
	0xffbb653f98	300	278448	Port-channel11	
2000	2852.6134.0000	0X1	32651	0xffba15e1a8	0xff454c2328
	0xffbb653f98	300	63	Port-channel11	
2000	0000.0012.3456	0X2000001	32655	0xffba15c508	0xff44f9ec98
	0x0	300	1	2000:33.33.33.33	

Total Mac number of addresses:: 4

\*a\_time=aging\_time(secs) \*e\_time=total\_elapsed\_time(secs)

Type:

MAT_DYNAMIC_ADDR	0x1	MAT_STATIC_ADDR	0x2
MAT_CPU_ADDR	0x4	MAT_DISCARD_ADDR	0x8
MAT_ALL_VLANS	0x10	MAT_NO_FORWARD	0x20
MAT_IPMULT_ADDR	0x40	MAT_RESYNC	0x80
MAT_DO_NOT_AGE	0x100	MAT_SECURE_ADDR	0x200
MAT_NO_PORT	0x400	MAT_DROP_ADDR	0x800
MAT_DUP_ADDR	0x1000	MAT_NULL_DESTINATION	0x2000
MAT_DOT1X_ADDR	0x4000	MAT_ROUTER_ADDR	0x8000
MAT_WIRELESS_ADDR	0x10000	MAT_SECURE_CFG_ADDR	0x20000
MAT_OPQ_DATA_PRESENT	0x40000	MAT_WIRED_TUNNEL_ADDR	0x80000
MAT_DLR_ADDR	0x100000	MAT_MRP_ADDR	0x200000
MAT_MSRRP_ADDR	0x400000	MAT_LISP_LOCAL_ADDR	0x800000
MAT_LISP_REMOTE_ADDR	0x1000000	MAT_VPLS_ADDR	0x2000000

The following is a sample output of **show bgp l2vpn vpls all** command :

```
BGP table version is 6, local router ID is 222.5.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
  r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
  x best-external, a additional-path, c RIB-compressed,
  t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 1000:2128
*> 1000:2128:1.1.1.72/96
      0.0.0.0          32768 ?
*>i 1000:2128:44.254.44.44/96
      44.254.44.44          0    100    0 ?
```

