

# Configuring Ethernet Virtual Connections on the Cisco ASR 1000 Series Router

Ethernet virtual circuit (EVC) infrastructure is a Layer 2 platform-independent bridging architecture that supports Ethernet services. This document describes the infrastructure and the features it supports on the Cisco ASR 1000 Series Aggregation Services Router.


- Restrictions for Configuring EVCs on the Cisco ASR 1000 Series Router, on page 1
- Information About Configuring EVCs on the Cisco ASR 1000 Series Router, on page 1
- How to Configure EVCs on the Cisco ASR 1000 Series Router, on page 8
- Configuration Examples for EVCs on the Cisco ASR 1000 Series Router, on page 11
- Additional References, on page 11
- Feature Information for Configuring EVCs on the Cisco ASR 1000 Series Router, on page 12


# Restrictions for Configuring EVCs on the Cisco ASR 1000 Series Router

- Bridge domain configuration is supported only as part of the EVC service instance configuration.
- The following features are not supported:
  - Service instance (Ethernet flow point [EFP]) group support
  - EVC cross-connect and connect forwarding services
  - Ethernet service protection (Ethernet Operations, Administration, and Maintenance [EOAM], connectivity fault management [CFM], Ethernet Local Management Interface [E-LMI]) on EVCs
  - IPv6 access control lists (ACLs) are not supported.

# Information About Configuring EVCs on the Cisco ASR 1000 Series Router

The following topics are described in this section and provide background information for configuring EVCs on the Cisco ASR 1000 Series Router:

In Cisco IOS XE Release 3.2S and later releases, the following features are supported in the EVC infrastructure:

In Cisco IOS XE Release 3.3S, Layer 3 and Layer 4 protocol support was added. This support is described in the "Layer 3 and Layer 4 ACL Support".

# EVCs

An EVC is defined by the Metro-Ethernet Forum (MEF) as an association between two or more user network interfaces that identifies a point-to-point or multipoint-to-multipoint path within the service provider network. An EVC is a conceptual service pipe within the service provider network. A bridge domain is a local broadcast domain that is VLAN-ID-agnostic. An Ethernet flow point (EFP) service instance is a logical interface that connects a bridge domain to a physical port.

An EVC broadcast domain is determined by a bridge domain and the EFPs that are connected to it. You can connect multiple EFPs to the same bridge domain on the same physical interface, and each EFP can have its own matching criteria and rewrite operation. An incoming frame is matched against EFP matching criteria on the interface, learned on the matching EFP, and forwarded to one or more EFPs in the bridge domain. If there are no matching EFPs, the frame is dropped.

You can use EFPs to configure VLAN translation. For example, if there are two EFPs egressing the same interface, each EFP can have a different VLAN rewrite operation, which is more flexible than the traditional switch port VLAN translation model.

# Service Instances and Associated EFPs

Configuring a service instance on a Layer 2 port creates a pseudoport or EFP on which you configure EVC features. Each service instance has a unique number per interface, but you can use the same number on different interfaces because service instances on different ports are not related.

An EFP classifies frames from the same physical port to one of the multiple service instances associated with that port, based on user-defined criteria. Each EFP can be associated with different forwarding actions and behavior.

When an EFP is created, the initial state is UP. The state changes to DOWN under the following circumstances:

- The EFP is explicitly shut down by a user.
- The main interface to which the EFP is associated is down or removed.
- If the EFP belongs to a bridge domain, the bridge domain is down.
- The EFP is forced down as an error-prevention measure of certain features.

Use the **service instance ethernet** interface configuration command to create an EFP on a Layer 2 interface and to enter service instance configuration mode. Service instance configuration mode is used to configure all management and control data plane attributes and parameters that apply to the service instance on a per-interface basis. The service instance number is the EFP identifier.

After the device enters service instance configuration mode, you can configure these options:

- default--Sets a command to its defaults
- description--Adds a service instance-specific description
- encapsulation--Configures Ethernet frame match criteria
- exit--Exits from service instance configuration mode

- no--Negates a command or sets its defaults
- shutdown--Takes the service instance out of service

# Encapsulation (Flexible Service Mapping)

Encapsulation defines the matching criteria that map a VLAN, a range of VLANs, class of service (CoS) bits, Ethertype, or a combination of these to a service instance. VLAN tags and CoS can be a single value, a range, or a list. Ethertype can be a single type or a list of types.

Different types of encapsulations are default, dot1ad, dot1q, priority-tagged, and untagged. On the Cisco ASR 1000 Series Router, priority-tagged frames are always single-tagged. Valid Ethertypes (type) are ipv4, ipv6, pppoe-all, pppoe-discovery, and pppoe-session.

Encapsulation classification options also include:

- inner tag CoS
- inner tag VLAN
- outer tag CoS
- outer tag VLAN
- outer tag Ethertype (VLAN type)--VLAN type is always matched. If you do not specify an alternative, the default is 0x8100 for dot1q and 0x88a8 for dot1ad.
- payload Ethertype--Any Ethertype tag after the VLAN tag

When you configure an encapsulation method, you enable flexible service mapping, which allows you to map an incoming packet to an EFP based on the configured encapsulation.

The default behavior for flexible service mapping based on outer 802.1q and 802.1ad VLAN tag values is nonexact, meaning that when the EFP encapsulation configuration does not explicitly specify an inner (second) VLAN tag matching criterion, the software maps both single-tagged and double-tagged frames to the EFP as long as the frames fulfill the criteria of outer VLAN tag values. The command-line interface (CLI) does allow you to specify exact mapping with the **exact** keyword. If this keyword is specified, the EFP is designated as single-tagged-frame-only and double-tagged frames are not classified to that EFP.

Using the CLI **encapsulation** command in service-instance configuration mode, you can set encapsulation criteria. You must configure one encapsulation command per EFP (service instance). After you have configured an encapsulation method, these commands are available in service instance configuration mode:

- **bridge-domain** --Configures a bridge domain.
- **rewrite** --Configures Ethernet rewrite criteria.

The table below shows the supported encapsulation types.

*Table 1: Supported Encapsulation Types*

| Command | Description |
|---|---|
| **encapsulation dot1q** *vlan-id* [**,** *vlan-id* [**-** *vlan-id*]] | Defines the matching criteria to be used to map 802.1q frames ingressing on an interface to the appropriate EFP. The options are a single VLAN, a range of VLANs, or lists of VLANs or VLAN ranges. VLAN IDs are 1 to 4094.<br>• Enter a single VLAN ID for an exact match of the outermost tag.<br>• Enter a VLAN range for a ranged outermost match. |
| **encapsulation dot1q** *vlan-id* **second-dot1q** *vlan-id* [**,** *vlan-id* [**-** *vlan-id*]] | Double-tagged 802.1q encapsulation. Matching criteria to be used to map QinQ frames ingressing on an interface to the appropriate EFP. The outer tag is unique and the inner tag can be a single VLAN, a range of VLANs or lists of VLANs or VLAN ranges.<br>• Enter a single VLAN ID in each instance for an exact match of the outermost two tags.<br>• Enter a VLAN range for second-dot1q for an exact outermost tag and a range for a second tag. |
| **encapsulation dot1q** {**any** \| *vlan-id* [**,** *vlan-id*[**-** *vlan-id*]]} **etype** *ethertype* | Ethertype encapsulation is the payload encapsulation type after VLAN encapsulation.<br>Ethertype encapsulation matches any or an exact outermost VLAN or VLAN range and a payload ethertype.<br>Valid values for *ethertype* are **ipv4**, **ipv6**, **pppoe-discovery**, **pppoe-session**, or **pppoe-all**. |
| **encapsulation dot1q** *vlan-id* **cos** *cos-value* **second-dot1q** *vlan-id* **cos** *cos-value* | CoS value encapsulation defines match criteria after including the CoS for the S-Tag and the C-Tag. The CoS value is a single digit between 1 and 7 for S-Tag and C-Tag.<br>You cannot configure CoS encapsulation with the **encapsulation untagged** command, but you can configure it with the **encapsulation priority-tagged** command. The result is an exact outermost VLAN and CoS match and second tag. You can also use VLAN ranges. |
| **encapsulation dot1q any** | Matches any packet with one or more VLANs. |
| **encapsulation dot1q vlan-type** | Specifies the value of the VLAN protocol type, which is the tag protocol identifier (TPID) of an 802.1q VLAN tag. If there is more than one tag, this command refers to the outermost tag. By default the TPID is assumed to be 0x8100. Use this command to set the TPID to other supported alternatives: 0x88A8, 0x9100, 0x9200. |
| **encapsulation dot1ad** | Defines the matching criteria to be used to map 802.1d frames ingressing on an interface to the appropriate EFP. |

| Command | Description |
|---|---|
| **encapsulation untagged** | Matching criteria to be used to map native Ethernet frames (without a dot1q tag) entering an interface to the appropriate EFP.<br>Only one EFP per port can have untagged encapsulation. However, a port that hosts EFP matching untagged traffic can also host other EFPs that match tagged frames. |
| **encapsulation default** | Configures the default EFP on an interface, acting as a catch-all encapsulation for all packets without a configured encapsulation. All packets are seen as native. If you enter the **rewrite** command with encapsulation default, the command is rejected.<br>Only one default EFP per interface can be configured. If you try to configure more than one default EFP, the command is rejected. |
| **encapsulation priority-tagged** | Specifies priority-tagged frames. A priority-tagged packet has VLAN ID 0 and a CoS value of 0 to 7. |

If a packet entering or leaving a port does not match any of the encapsulations on that port, the packet is dropped, resulting in filtering on both ingress and egress. The encapsulation must match the packet on the wire to determine filtering criteria. On the wire refers to packets ingressing the router before any rewrites and to packets egressing the router after all rewrites.

# Layer 3 and Layer 4 ACL Support

Beginning in Cisco IOS XE Release 3.3S, support was added for configuring IPv4 Layer 3 and Layer 4 ACLs on EFPs. Configuring an ACL on an EFP is the same as configuring an ACL on other types of interfaces; for example, Ethernet or asynchronous transfer mode (ATM). One exception is that ACLs are not supported for packets prefixed with a Multiprotocol Label Switching (MPLS) header, including when an MPLS packet contains either Layer 3 or Layer 4 headers of supported protocols.

An ACL configured on a main interface containing EFPs does not affect traffic through the EFPs.

To configure an IPv4 Layer 3 and Layer 4 ACL on an EFP, use the **ip access-group** command. An ACL configuration is shown in the "Configuring an ACL on an EFP".

# Advanced Frame Manipulation

The Advanced Frame Manipulation feature allows you to specify the VLAN tag manipulation needed on both the incoming and outgoing frames of an EFP. These manipulations include PUSH, POP, and TRANSLATION of one or both VLAN tags.

The PUSH, POP, and TRANSLATION manipulations are as follows:

- PUSH Operations
  - Add one VLAN tag
  - Add two VLAN tags
- POP Operations
  - Remove the outermost VLAN tag
  - Remove the two outermost VLAN tags

- TRANSLATION Operations
  - 1:1 VLAN Translation
  - 1:2 VLAN Translation
  - 2:1 VLAN Translation
  - 2:2 VLAN Translation

When a VLAN tag exists and a new one is added, the CoS field of the new tag is set to the same value as the CoS field of the existing VLAN tag; otherwise, the CoS field is set to a default of 0. Using QoS marking configuration commands, you can change the CoS marking.

### EFPs and Layer 2 Protocols

On the Cisco ASR 1000 Series Router, EFPs treat the protocol data units (PDUs) of Layer 2 protocols as data frames. PDUs are forwarded as data frames.

Layer 2 protocols include Cisco Discovery Protocol, Dynamic Trunking Protocol (DTP), Link Aggregation Control Protocol (LACP), Link Layer Discovery Protocol (LLDP), Multiple Spanning Tree Protocol (MSTP), Port Aggregation Protocol (PAgP), Unidirectional Link Detection (UDLD), and VLAN Trunk Protocol (VTP).

## Egress Frame Filtering

Egress frame filtering is performed to ensure that frames exiting an EFP contain a Layer 2 header that matches the encapsulation characteristics associated with the EFP. This filtering is done primarily to prevent unintended frame leaks and is always enabled on EFPs.

## Bridge Domains

A bridge domain defines a broadcast domain internal to a platform and allows the decoupling of a broadcast domain from a VLAN. This decoupling enables per-port VLAN significance, thus removing the scalability limitations associated with a single per-device VLAN ID space. You can configure a maximum of 4096 EFPs per bridge domain.

A bridge domain interface (BDI) is used to support frame forwarding in a bridge domain at Layer 3. The BDI is a virtual interface that supports Layer 3 features. Each bridge domain can have only one BDI configuration.

If the destination MAC address in a frame received from one of the EFPs participating in a bridge domain matches the BDI MAC address, the frame is routed; otherwise, the frame is bridged. When the egress interface for a routed packet is the BDI interface, the frame is bridged using the destination MAC address.

Frames with broadcast and well-known multicast MAC addresses are also forwarded to the BDI.

The following sections describe support for bridge domains:

EFP, bridge domain, and BDI support based on the Cisco ASR 1000 Series Router forwarding processors are shown in the table in "EFP Bridge Domain and BDI Support Based on the Cisco ASR 1000 Series Router Forwarding Processors".

### Ethernet MAC Address Learning

MAC address learning is always enabled and cannot be disabled.

## Flooding of Layer 2 Frames for Unknown MAC Multicast and Broadcast Addresses

A Layer 2 frame with an unknown unicast or broadcast destination MAC address is flooded to all the EFPs in the bridge domain except to the originating EFP. A frame with a multicast MAC address is flooded to all the EFPs in the bridge domain. If the destination MAC address is a multicast MAC address, the frame is treated like a broadcast frame and sent to all the EFPs in the bridge domain.

When a frame with either a multicast or broadcast MAC address is flooded and a BDI is associated with the bridge domain, the frame is also flooded to the BDI.

Replication of frames involves recycling the frame several times. This recycling may have a negative effect on forwarding performance and reduce the packet forwarding rate for all features.

## Layer 2 Destination MAC Address-Based Forwarding

When bridging is configured, a unicast frame received from an EFP is forwarded based on the destination Layer 2 MAC address. If the destination address is known, the frame is forwarded only to the EFP associated with the destination address.

Because bridge and EFP configurations are interrelated, bridging is supported only on EFPs. To support multiple bridge domains, MAC address entries are associated with the bridge domain of the EFP. Only unicast MAC addresses need to be dynamically learned.

EVC infrastructure does not modify frame contents. Each bridge domain can learn 1000 MAC addresses per second. The Layer 2 frame forwarding rate is 8 million packets per second (MPPS) if flooding is not involved.

## MAC Address Aging

The dynamically learned MAC address entries in the MAC table are periodically aged out and entries that are inactive for longer than the configured time period are removed from the table. The supported range of aging-time values, in seconds, is 30 to 600 with a granularity of 1. The default is 5 minutes.

The **aging-time** parameter can be configured per bridge domain and is a relative value. The value is the aging time relative to the time a frame was received with that MAC address.

## MAC Address Move

As stations (systems connected to the Cisco ASR 1000 Series Router through the EFP interface) move from one network to another, the interface associated with a MAC address changes.

## MAC Address Table

The MAC address table is used to forward frames based on Layer 2 destination MAC addresses. The table consists of static MAC addresses downloaded from the route processor (RP) and the MAC addresses dynamically learned by the data path.

While the MAC Learning feature is enabled, an entry is added to the MAC table when a new unique MAC address is learned on the data path and an entry is deleted from the table when it is aged out.

## Split Horizon Group

The split-horizon feature allows service instances in a bridge domain to join groups. Service instances in the same bridge domain and split-horizon group cannot pass data to each other but can forward data to other service instances that are in the same bridge domain and not in the same split-horizon group.

A service instance cannot join more than one split-horizon group. A service instance does not have to be in a split-horizon group. When a service instance does not belong to a group, it can send and receive data from all ports within the bridge domain.

One or more EFPs in a bridge domain may be configured for the same split horizon group, but when a frame is replicated to EFPs, that frame cannot be replicated to EFPs that are within the same split horizon group as the input interface. This restriction applies to MAC address frames that are either known or unknown unicast, broadcast, and multicast frames.

Two split horizon groups per bridge domain are supported on the Cisco ASR 1000 Series Router. You can configure a split horizon group using the **bridge-domain** CLI command with the **split-horizon** and **group** keywords. The group ID can be either 0 or 1.

All members of the bridge-domain that are configured with the same group ID are part of the same split-horizon group. EFPs that are not configured with an explicit group ID do not belong to any group.

## EFP Bridge Domain and BDI Support Based on the Cisco ASR 1000 Series Router Forwarding Processors

The table below shows EFP, bridge domain, and BDI support based on the Cisco ASR 1000 Series Router forwarding processors.

***Table 2: EFP, Bridge Domain, and BDI Support on the Cisco ASR 1000 Series Router Forwarding Processors***

| Description | ASR1000-ESP5, ASR 1001, ASR 1002-F (ESP2.5) | ASR1000-ESP10, ASR1000-ESP10-N, ASR1000-ESP20, | ASR1000-ESP40 |
|---|---|---|---|
| Maximum EFPs per router | 8192 | 16384 | 24576 |
| Maximum EFPs per bridge domain | 4000 | 4000 | 4000 |
| Maximum EFPs per interface | 8000 | 8000 | 8000 |
| Maximum bridge domains per router | 4096 | 4096 | 4096 |
| Maximum BDIs per router | 4096 | 4096 | 4096 |
| Maximum MAC table entries per router | 65536 | 65536 | 65536 |
| Maximum MAC table entries per bridge domain | 16384 | 16384 | 16384 |
| Maximum split horizon groups per bridge domain | 2 | 2 | 2 |

# How to Configure EVCs on the Cisco ASR 1000 Series Router

## Configuring an EFP and a Bridge Domain on the Cisco ASR 1000 Series Router

Configuring a service instance on a Layer 2 port creates an EFP on which you can configure EVC features. Perform this task to configure an EFP.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service instance** *id* **ethernet**
5. **encapsulation** *encapsulation-type vlan-id*
6. **rewrite ingress tag translate 1-to-1 dot1q** *vlan-id* **symmetric**
7. **bridge-domain** *bridge-id*
8. **end**

## DETAILED STEPS

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface** *type number*<br>**Example:**<br>`Router(config)# interface gigabitethernet 0/1/1` | Enters interface configuration mode.<br>• The example shows how to configure Gigabit Ethernet interface 0/1/1 and enter interface configuration mode. |
| **Step 4** | **service instance** *id* **ethernet**<br>**Example:**<br>`Router(config-if)# service instance 1 ethernet` | Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.<br>• The example shows how to configure Ethernet service instance 1. |
| **Step 5** | **encapsulation** *encapsulation-type vlan-id*<br>**Example:**<br>`Router(config-if-srv)# encapsulation dot1q 1` | Defines the encapsulation type.<br>• The example shows how to define dot1q as the encapsulation type. |
| **Step 6** | **rewrite ingress tag translate 1-to-1 dot1q** *vlan-id* **symmetric**<br>**Example:**<br>`Router(config-if-srv)# rewrite ingress tag translate 1-to-1 dot1q 1 symmetric` | (Optional) Specifies the encapsulation adjustment to be performed on a frame ingressing a service instance.<br>• The example shows how to specify translating a single tag defined by the **encapsulation** command to a single tag defined in the **rewrite ingress tag** command with reciprocal adjustment to be done in the egress direction. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 7** | **bridge-domain** *bridge-id*<br>**Example:**<br>`Router(config-if-srv)# bridge-domain 1` | Configures the bridge domain.<br>• The example shows how to configure bridge domain 1. |
| **Step 8** | **end**<br>**Example:**<br>`Router(config-if-srv)# end` | Returns to privileged EXEC mode. |

# Configuring an ACL on an EFP

Perform this task to configure an ACL on an EFP.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip access-group** *access-list-number* | *access-list-name*} {**in** | **out**}
5. **end**

### DETAILED STEPS

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface** *type number*<br>**Example:**<br>`Router(config)# interface gigabitethernet 0/1/1` | Enters interface configuration mode.<br>• The example shows how to configure Gigabit Ethernet interface 0/1/1 and enter interface configuration mode. |
| **Step 4** | **ip access-group** *access-list-number* \| *access-list-name*} {**in** \| **out**}<br>**Example:** | Applies an IP access list or object group access control list (OGACL) to an interface or a service policy map. |

| | Command or Action | Purpose |
|---|---|---|
| | `Router(config-if)# ip access-group acl55 in` | • The example shows how to configure an ACL named acl55 for inbound packets. |
| **Step 5** | **end**<br>**Example:**<br>`Router(config-if)# end` | Returns to privileged EXEC mode. |

# Configuration Examples for EVCs on the Cisco ASR 1000 Series Router

## Example Configuring EFPs on a Gigabit Ethernet Interface

```
interface GigabitEthernet0/0/1
 no ip address
 negotiation auto
 service instance 1 ethernet
  encapsulation dot1q 201
  rewrite ingress tag translate 1-to-1 dot1q 300 symmetric
  bridge-domain 1
 !
 service instance 2 ethernet
  encapsulation default
  bridge-domain 1
 !
 service instance 3 ethernet
  encapsulation priority-tagged
  bridge-domain 2
 !
```

# Additional References

### Related Documents

| Related Topic | Document Title |
|---|---|
| IEEE CFM | "Configuring IEEE Standard-Compliant Ethernet CFM in a Service Provider Network" |
| Using OAM | "Using Ethernet Operations, Administration, and Maintenance" |
| IEEE CFM and Y.1731 commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples | *Cisco IOS Carrier Ethernet Command Reference* |

### Standards

| Standard | Title |
|---|---|
| IEEE 802.1ag | *802.1ag - Connectivity Fault Management* |
| IEEE 802.3ah | *Ethernet in the First Mile* |
| ITU-T | *ITU-T Y.1731 OAM Mechanisms for Ethernet-Based Networks* |

### 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 Configuring EVCs on the Cisco ASR 1000 Series Router

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 3: Feature Information for Configuring EVCs on the Cisco ASR 1000 Series Router***

| Feature Name | Releases | Feature Information |
|---|---|---|
| ASR1000 EVC Infrastructure | Cisco IOS XE Release 3.2S Cisco IOS XE Release 3.3S | EVC infrastructure is a Layer 2 platform-independent bridging architecture that supports Ethernet services.<br>In Cisco IOS XE Release 3.2S, this feature was introduced on the Cisco ASR 1000 Series Router.<br>The following commands are introduced or modified:**rewrite egress tag**, **rewrite ingress tag**, and **shutdown** (bdomain). |

| Feature Name | Releases | Feature Information |
|---|---|---|
| ASR1000 BD Infrastructure | Cisco IOS XE Release 3.2S | Bridge domain infrastructure is a Layer 2 platform-independent architecture that enables bridging.<br>In Cisco IOS XE Release 3.2S this feature was introduced on the Cisco ASR 1000 Series Router. The following sections provide information about support for this feature:<br>The following commands are introduced or modified:**bridge-domain** (service instance), **mac aging-time**. |
| ACL and QoS Enhancements to EVC Infrastructure in Cisco IOS XE Software | Cisco IOS XE Release 3.3S | Support for configuring Layer 3 and Layer 4 ACLs on EFPs was added in Cisco IOS XE Release 3.3S.<br>The following commands are introduced or modified:**ip access-group**. |