Cisco Catalyst 8500 and 8500L Series Edge Platforms Software Configuration Guide

Cisco Catalyst 8500 Edge Platform
Updated February 6, 2026

© 2026 Cisco Systems, Inc. All rights reserved.

Full Cisco Trademarks with Software License

Full Cisco Trademarks with Software License

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

Topics included

Full Cisco Trademarks with Software License....ii

1 Overview of Catalyst 8500 and 8500L Series Edge Platforms....7

Overview....8
Breakout support....8
Breakout support configuration....9
Bay configuration C8500-12X4QC....10
Bay configuration examples....12
Bay configuration C8500-12X....14
Bay configuration C8500-20X6C....15

2 Software Package and Architecture....17

Software package and architecture....18
Consolidated packages....18
Individual software subpackages within a consolidated package....18
Provisioning files....18
Upgrade field programmable hardware devices....19
Processes overview....19
Cisco IOS XE software....21
Access Cisco IOS XE software....21
Console connections....21
Access the device console using SSH....22
Remote CLI access methods....23
USB serial console ports....24
Keyboard shortcuts....24
History buffer....25
CLI command forms....32
Save configuration changes....33
Configuration files....33
Filtering output from the show and more commands....34
Save configuration changes before powering off....34
Cisco software images....34
CLI session management....35
Deploy IOS-XE and SD-WAN....37
Autonomous or controller mode....37
Switch between controller and autonomous modes....37
PnP discovery process....37
Software upgrade....38

Perform a software upgrade....38

3 Factory Reset....39

Factory reset....40

About factory reset....40

Software and hardware support for factory reset....42

Prerequisites for performing factory reset....42

Restrictions for performing a factory reset....42

When to perform factory reset....43

How to perform a factory reset....43

What happens after a factory reset....44

4 Security-Enhanced Linux....45

Security-Enhanced Linux....46

Prerequisites....46

Restrictions....46

Configure Security-Enhanced Linux in EXEC Mode....46

Configure Security-Enhanced Linux in CONFIG Mode....47

SYSLOG message reference....47

Verify Security-Enhanced Linux enablement....48

Troubleshoot Security-Enhanced Linux....48

5 High Availability....51

High Availability....52

Interchassis high availability....52

Configure Interchassis High Availability....53

Bidirectional Forwarding Detection....53

Configure Bidirectional Forwarding....54

Verify Interchassis High Availability....54

Verify BFD Offload....62

6 Management Ethernet Interface....67

Management Ethernet Interface....68

Gigabit Ethernet management interface....68

IP Address in ROMmon and the management Ethernet port....68

GigabitEthernet Management interface VRF....68

Common Ethernet management tasks....69

7 Bridge Domain Interfaces....73

Information about bridge domain interface....74

Restrictions for bridge domain interfaces....74

Ethernet virtual circuit overview....75

Bridge domain interface encapsulation....75
Assign a MAC address....75
Support for IP protocols....76
IP forward support....76
Packet forwarding....76
Link states of a bridge domain and a bridge domain interface....77
Bridge domain interface statistics....78
Create or delete a bridge domain interface....78
Bridge domain virtual IP interface....78
How to configure a bridge domain interface....79
Display and verify bridge domain interface configuration....81
Configure bridge domain virtual IP interface....82
Associate VIF interface with a bridge domain....83
Verify bridge domain virtual IP interface....83
Example configuration bridge domain virtual IP interface....83

8 Diagnostic Utilities....85
Packet drops....86
Packet drops....86
Packet drop diagnostics....86
View packet drop information....86
Commands for viewing packet drop information....88
Packet drop warning thresholds....89
Configure packet drop warning threshold....89
View packet drop warning thresholds....91
Packet trace....92
Packet trace....92
Packet trace feature levels....92
Best practice for packet trace configuration....97
Packet trace display commands....98
Commands for clearing packet trace data....98
Configure packet trace....98

9 SFP Module and Forward Error Correction....105
Configure SFP+....106
Configure SFP+....106
Configure Forward Error Correction....107

10 Cisco ThousandEyes Enterprise Agent Application....109
Cisco ThousandEyes Enterprise agent application....110
Supported platforms and system requirements....110
Workflow to install and run the Cisco ThousandEyes application....111
Modify the agent parameters....116

Uninstall the application....116
Troubleshoot the Cisco ThousandEyes application....117

1 Overview of Catalyst 8500 and 8500L Series Edge Platforms

Topics:

- Overview
- Breakout support
- Bay configuration C8500-12X4QC

Describes the hardware capabilities and port configurations for various edge platform models. It explains how to implement breakout support and manage bay configurations to optimize interface density and traffic speeds.

Overview

Introduces the Catalyst 8500 and 8500L Series Edge platforms and their specific hardware models. Learn about the performance capabilities and supported features for each router to determine the appropriate configuration for your network environment.

Cisco Catalyst 8500 and 8500L Series Edge platforms significantly increases services performance, router throughput, and router scale at lower costs.

This document covers configuration details of these models:

- Catalyst 8500 Platforms (C8500-12X4QC, C8500-12X and C8500-20X6C)
- Catalyst 8500L Platform (C8500L-8S4X)

Table 1: Catalyst 8500 and Edge 8500L Platforms

Features	C8500-12X4QC	C8500-12X	C8500-20X6C	C8500L-8S4X
Support for In-Service Software Upgrade (ISSU)	Not supported	Not supported	Not supported	Not supported
Data plane processing	QFP 3.0	QFP 3.0	QFP 3.0	Software-based
Support for Unified Threat Defense(UTD)	Not supported	Not supported	Not Supported	Support exists
Support for Fast Reroute(FRR)	Not supported	Not supported	Support exists	Not supported
Dynamic allocation of cores	Not supported	Not supported	Not supported	Not supported

Note

- Local SPAN is not supported on any Cisco Catalyst 8500 and 8500L Series Edge platforms.
- The combination of overload and match-host within a single NAT pool is not supported.

Breakout support

Explains how to split high-speed native ports into multiple independent 10G logical ports. Use this feature to increase port density and provide flexibility for various network connectivity requirements.

Breakout support for a port helps to split a higher density port to multiple independent and logical ports. Starting from Cisco IOS XE 17.4, breakout support is introduced in Bay 2 of C8500-12X4QC that supports breakout capable 40G native ports. The breakout support is of 4X10G and uses a 3-tuple approach.

Note

Breakout support is only supported on C8500-12X4QC and not on C8500-20X6C.

This table explains the interface names when breakout is configured:

Table 2: Interface names when breakout support is configured for C8500-12X4QC

Interface names	Description
Te0/2/0, Te0/2/1, Te0/2/2, Te0/2/3, Te0/2/4, Te0/2/5, Te0/2/6, Te0/2/7, Te0/2/8, Te0/2/9, Te0/2/10, Te0/2/11	All three 40 G native ports working in 10G breakout mode
Fo0/2/0,Fo0/2/4, Te0/2/8, Te0/2/9, Te0/2/10, Te0/2/11	1st native port in 40G mode 2nd native port in 40G mode 3rd native port in 10G breakout mode
Fo0/2/0, Te0/2/4, Te0/2/5, Te0/2/6, Te0/2/7 Fo0/2/8	1st native port in 40G mode 2nd native port 10G breakout mode 3rd native port in 40G mode
Te0/2/0, Te0/2/1, Te0/2/2, Te0/2/3, Fo0/2/4, Fo0/2/8	1st native port in 10G breakout mode 2nd native port in 40G mode 3rd native port in 40G mode
1st native port in 10G breakout mode 2nd native port in 40G mode 3rd native port in 40G mode	1st native port in 40G mode 2nd native port in 10G breakout mode 3rd native port in 10G breakout mode
Te0/2/0, Te0/2/1, Te0/2/2, Te0/2/3, Te0/2/4, Te0/2/5, Te0/2/6, Te0/2/7, Fo0/2/8	1st native port in 10G breakout mode 2nd native port in 10G breakout mode 3rd native port in 40G mode
Te0/2/0, Te0/2/1, Te0/2/2, Te0/2/3, Fo0/2/4, Te0/2/8, Te0/2/9, Te0/2/10, Te0/2/11	1st native port in 10G breakout mode 2nd native port in 40G mode 3rd native port in 10G breakout mode

Breakout support configuration

This topic outlines commands and examples for configuring breakout support.

> **Note**
>
> Before using the breakout capability, ensure that Bay 2 is configured in 40G mode

```
Router(config)#hw-module subslot 0/2 breakout 10G port ?

all                 configure all native ports in breakout mode
native_port_0       configure native port 0 in breakout mode
native_port_4       configure native port 4 in breakout mode
native_port_8       configure native port 8 in breakout mode
```

Sample commands to configure breakout support

When native_port 0 and 8 are in 10G breakout and native_port 4 is running in 40G mode

```
hw-module subslot 0/2 breakout 10g port native_port_0
hw-module subslot 0/2 breakout 10g port native_port_8
```

When all three native 40G ports have same breakout config

```
hw-module subslot 0/2 breakout 10g port all
hw-module subslot 0/2 breakout none port all
```

When you want to remove breakout configuration from all ports

```
hw-module subslot 0/2 breakout none port all
```

Bay configuration C8500-12X4QC

Provides details and examples for configuring hardware bays on specific Catalyst 8500 models. Use these procedures to change interface modes and optimize traffic speeds across different built-in adapters.

On C8500-12X4QC there are three built-in EPAs that are configurable.

This table describes the port details:

Table 3: Bay configuration for C8500-12X4QC

Bay Number	EPA	Port Configuration	Interface numbers
Bay 0 8xSFP+	1/10G EPA	Eight 1/10G interfaces - TE0 - TE7 Disabled when 100G port in used in Bay 1	0/0/0 0/0/1 0/0/2 0/0/3 0/0/4 0/0/5 0/0/6 0/0/7 0/0/8

Bay Number	EPA	Port Configuration	Interface numbers
Bay 1 4xSFP+/1xQSFP	1/10/40/100G EPA	Four 1/10G interfaces active - TE0 - TE3 (interfaces 0/1/0 ... 0/1/3) The bay can be used in these modes: • Four 1/10G interfaces • One 40G interface active • One 100G interface. This utilizes the eight 1/10G ports of Bay 0	0/1/0 0/1/1 0/1/2 0/1/3
Bay 2 3xQSFP	40/100G EPA	Three 40G interfaces (0/1/0 to 0/1/2) One 100G interface (0/0/0) (0/0/0)	0/0/0 0/1/0 0/1/1 0/1/2

Note

The speed of a 10G interface can be 1G or 10G based on the SFP transceiver plugged into to the port. Even when the speed changes the interface name is still indicated as TenGigabitEthernet.

By default , C8500-12X4QC operates Bay 1 in 10G mode and Bay 2 in 40G mode. The Bay 1 mode can be changed from 10G to 40G to 100G and vice versa. But if Bay 1 is set to 100G, all ports of Bay 0 move to **admin down** state and the ports are no longer functional.

The Bay 2 mode can be changed from 40G to 100G and vice versa. The mode change on Bay 2 does not impact traffic on Bay 1.

Use the **show platform** and **show ip interface** commands to view the bay and interface details:

```
Router#show platform
Chassis type: C8500-12X4QC

Slot      Type                State                 Insert time (ago)
--------- ------------------- --------------------- -----------------
0         C8500-12X4QC        ok                    1w3d
 0/0      BUILTIN-8x1/10G     ok                    1w3d
 0/1      BUILTIN-100/40/4x10Gok                    00:04:53
 0/2      BUILTIN-100G/3X40G  ok                    00:08:16
R0        C8500-12X4QC        ok                    1w3d
 R0/0                         ok, active            1w3d
 R0/1                         ok, standby           1w3d
F0        C8500-12X4QC        ok, active            1w3d
P0        AIR-AC-750W-R       ok                    1w3d
P1        AIR-AC-750W-R       ps, fail              1w3d
P2        C8500-FAN-1R        ok                    1w3d
```

```
Slot      CPLD Version        Firmware Version
--------- ------------------- ----------------------------------------
0         19020715            12.2(20181120:104547) [user-gd_secur...
R0        19020715            12.2(20181120:104547) [user-gd_secur...
F0        19020715            12.2(20181120:104547) [user-gd_secur...
```

```
Router#show ip interface
Te0/0/0                unassigned      YES NVRAM  down                    down
Te0/0/1                unassigned      YES NVRAM  down                    down
Te0/0/2                unassigned      YES NVRAM  down                    down
Te0/0/3                unassigned      YES NVRAM  down                    down
Te0/0/4                unassigned      YES NVRAM  down                    down
Te0/0/5                unassigned      YES NVRAM  down                    down
Te0/0/6                unassigned      YES NVRAM  down                    down
Te0/0/7                unassigned      YES NVRAM  down                    down
Te0/1/0                unassigned      YES NVRAM  down                    down
Te0/1/1                unassigned      YES NVRAM  down                    down
Te0/1/2                unassigned      YES NVRAM  down                    down
Te0/1/3                unassigned      YES NVRAM  down                    down
Fo0/2/0                unassigned      YES unset  down                    down
Fo0/2/4                unassigned      YES unset  down                    down
Fo0/2/8                unassigned      YES unset  down                    down
GigabitEthernet0       10.104.33.213   YES NVRAM  up                      up
Router#
```

Bay configuration examples

These examples show how mode can be changed on C8500-12X4QC to achieve different traffic speeds:

This example shows how to change to 40G mode on Bay 1 of C8500-12X4QC:

```
Router(config)# hw-module subslot 0/1 mode 40G
Present configuration of this subslot will be erased and will not be restored.
CLI will not be available until mode change is complete and EPA returns to OK
 state.
Do you want to proceed? [confirm]

The "[no] negotiation auto" command will have no effect with this interface
The "[no] negotiation auto" command will have no effect with this interface
The "[no] negotiation auto" command will have no effect with this interface
The "[no] negotiation auto" command will have no effect with this interface
*Oct 29 17:58:10.020 IST: BUILTIN-100/40/4x10G[0/1] : config for spa port 0
would be lost
*Oct 29 17:58:10.028 IST: %SYS-5-CONFIG_P: Configured programmatically by
process Exec from console as console
*Oct 29 17:58:10.028 IST: %SYS-5-CONFIG_P: Configured programmatically by
process Exec from console as console
*Oct 29 17:58:10.028 IST: BUILTIN-100/40/4x10G[0/1] : TenGigabitEthernet0/1/0
 moved to default config
*Oct 29 17:58:10.028 IST: BUILTIN-100/40/4x10G[0/1] : config for spa port 1
would be lost
*Oct 29 17:58:10.035 IST: %SYS-5-CONFIG_P: Configured programmatically by
process Exec from console as console
*Oct 29 17:58:10.036 IST: %SYS-5-CONFIG_P: Configured programmatically by
process Exec from console as
:
:
*Oct 29 17:58:11.060 IST: BUILTIN-100/40/4x10G[0/1] : Old mode cleanup done!
*Oct 29 17:58:11.061 IST: %SPA_OIR-6-OFFLINECARD: SPA (BUILTIN-100/40/4x10G)
offline in subslot 0/1
*Oct 29 17:58:16.297 IST: BUILTIN-100/40/4x10G[0/1] : Number of ports 1
```

```
*Oct 29 17:58:16.298 IST: BUILTIN-100/40/4x10G[0/1] : XCVR namestring create:
 Maximum number of XCVR = 1
```

This example shows how to change to 40G mode to 100G on Bay 1 of C8500-12X4QC:

```
Router(config)# hw-module subslot 0/1 mode 100G
Changing mode of subslot 0/1 to 100G will cause EPA in subslot 0/0 to go offline
Present configuration of this subslot will be erased and will not be restored.
CLI will not be available until mode change is complete and EPA returns to OK
 state.
Do you want to proceed? [confirm]

*Oct 29 18:09:01.360 IST: BUILTIN-100/40/4x10G[0/1] : config for spa port 0
would be lost
*Oct 29 18:09:01.368 IST: %SYS-5-CONFIG_P: Configured programmatically by
process Exec from console as console
*Oct 29 18:09:01.368 IST: %SYS-5-CONFIG_P: Configured programmatically by
process Exec from console as console
The "[no] negotiation auto" command will have no effect with this interface
The "[no] negotiation auto" command will have no effect with this interface
The "[no] negotiation auto" command will have no effect with this interface
The "[no] negotiation auto" command will have no effect with this interface
The "[no] negotiation auto" command will have no effect with this interface
The "[no] negotiation auto" command will have no effect with this interface
The "[no] negotiation auto" command will have no effect with this interface
The "[no] negotiation auto" command will have no effect with this interface
*Oct 29 18:09:01.368 IST: BUILTIN-100/40/4x10G[0/1] : FortyGigabitEthernet0/1/0
 moved to default config
*Oct 29 18:09:02.368 IST: BUILTIN-8x1/10G[0/0] : config for spa port 0 would
be lost
:
:
*Oct 29 18:09:03.446 IST: %SPA_OIR-6-OFFLINECARD: SPA (BUILTIN-100/40/4x10G)
offline in subslot 0/1
*Oct 29 18:09:08.790 IST: BUILTIN-100/40/4x10G[0/1] : Number of ports 1
*Oct 29 18:09:08.792 IST: BUILTIN-100/40/4x10G[0/1] : XCVR namestring create:
 Maximum number of XCVR = 1
Router(config)#
*Oct 29 18:09:15.552 IST: %SPA_OIR-6-ONLINECARD: SPA (BUILTIN-100/40/4x10G)
online in subslot 0/1
```

This example shows how to change to 10G mode from 100G on Bay 1 of C8500-12X4QC:

```
Router(config)# hw-module subslot 0/1 mode 10G
Present configuration of this subslot will be erased and will not be restored.
CLI will not be available until mode change is complete and EPA returns to OK
 state.
Do you want to proceed? [confirm]
*Oct 29 18:14:36.484 IST: %PLATFORM_SCC-1-AUTHENTICATION_FAIL: Chassis
authentication failed

*Oct 29 18:14:38.219 IST: BUILTIN-100/40/4x10G[0/1] : config for spa port 0
would be lost
*Oct 29 18:14:38.227 IST: %SYS-5-CONFIG_P: Configured programmatically by
process Exec from console as console
*Oct 29 18:14:38.227 IST: %SYS-5-CONFIG_P: Configured programmatically by
process Exec from console as console
*Oct 29 18:14:38.227 IST: BUILTIN-100/40/4x10G[0/1] : HundredGigE0/1/0 moved
to default config
*Oct 29 18:14:39.228 IST: BUILTIN-100/40/4x10G[0/1] : Received mode change
request from 100G to 10G! system_configured TRUE
*Oct 29 18:14:39.230 IST: %IOSXE_OIR-6-SOFT_RELOADSPA: SPA(BUILTIN-100/40/4x10G)
```

```
 reloaded on subslot 0/1
*Oct 29 18:14:39.230 IST: BUILTIN-100/40/4x10G[0/1] : EPA moving from 100G
mode to 10G mode
*Oct 29 18:14:39.230 IST: BUILTIN-100/40/4x10G[0/1] : config for spa port 0
would be lost
*Oct 29 18:14:39.231 IST: BUILTIN-100/40/4x10G[0/1] : Old mode cleanup done!
*Oct 29 18:14:39.232 IST: %SPA_OIR-6-OFFLINECARD: SPA (BUILTIN-100/40/4x10G)
offline in subslot 0/1
*Oct 29 18:14:44.472 IST: BUILTIN-100/40/4x10G[0/1] : Number of ports 4
*Oct 29 18:14:44.475 IST: BUILTIN-100/40/4x10G[0/1] : XCVR namestring create:
 Maximum number of XCVR = 4
*Oct 29 18:15:03.336 IST: %SPA_OIR-6-ONLINECARD: SPA (BUILTIN-100/40/4x10G)
online in subslot 0/1
```

This example shows how to change to 100G mode from 100G on Bay 2 of C8500-12X4QC:

```
Router(config)# hw-module subslot 0/2 mode 100G
Present configuration of this subslot will be erased and will not be restored.
CLI will not be available until mode change is complete and EPA returns to OK
 state.
Do you want to proceed? [confirm]

*Oct 29 18:17:03.394 IST: BUILTIN-100G/3X40G[0/2] : config for spa port 0 would
 be lost
*Oct 29 18:17:03.401 IST: %SYS-5-CONFIG_P: Configured programmatically by
process Exec from console as console
*Oct 29 18:17:03.401 IST: %SYS-5-CONFIG_P: Configured programmatically by
process Exec from console as console
*Oct 29 18:17:03.401 IST: BUILTIN-100G/3X40G[0/2] : FortyGigabitEthernet0/2/0
 moved to default config
*Oct 29 18:17:03.401 IST: BUILTIN-100G/3X40G[0/2] : config for spa port 1 would
 be lost
*Oct 29 18:17:03.406 IST: BUILTIN-100G/3X40G[0/2] : Breakout XCVR type QSFP
4X10G AC7M (546) is not allowed as XCVR port Fortyrnet0/2/0 is not configured
 in breakout
*Oct 29 18:17:03.406 IST: %IOSXE_EPA-3-XCVR_PROHIBIT: Transceiver is prohibited
 to come online for interface FortyGigabitEther
*Oct 29 18:17:03.407 IST: BUILTIN-100G/3X40G[0/2] : XCVR prohibited on port
FortyGigabitEthernet0/2/0, epa_name=BUILTIN-100G/3=FortyGigabitEthernet0/2/0,
 xcvr_speed=40000000, admin_state=UNSHUT xcvr_type=546

*Oct 29 18:17:03.409 IST: %SYS-5-CONFIG_P: Configured programmatically by
process Exec from console as console
*Oct 29 18:17:03.409 IST: %SYS-5-CONFIG_P: Configured programmatically by
process Exec from console as console
*Oct 29 18:17:03.409 IST: BUILTIN-100G/3X40G[0/2] : FortyGigabitEthernet0/2/4
 moved to default config
*Oct 29 18:17:03.409 IST: BUILTIN-100G/3X40G[0/2] : config for spa port 2 would
 be lost
:
:
*Oct 29 18:17:09.686 IST: BUILTIN-100G/3X40G[0/2] : XCVR namestring create:
Maximum number of XCVR = 1
Router(config)#
Router(config)#
*Oct 29 18:17:16.017 IST: %SPA_OIR-6-ONLINECARD: SPA (BUILTIN-100G/3X40G)
online in subslot 0/2
```

Bay configuration C8500-12X

This topic covers details of bay configuration for C8500-12X that has one built-in EPA.

On C8500-12X4 there is one built-in EPA that supports ports TE0 - TE11 for SFP/SFP+ transceivers.

The speed of a 10G interface can be 1G or 10G based on the SFP transceiver plugged into to the port. Even when the speed changes the interface name is still indicated as TenGigabitEthernet.

Bay configuration C8500-20X6C

This topic covers details of bay configuration on C8500-20X6C that has two built-in EPAs.

On C8500-20X6C there are two built-in EPAs that are configurable.

Table 4: Bay configuration for C8500-20X6C

Bay Number	EPA	Port Configuration	Interface numbers
Bay 0 20xSFP+	1/10G EPA	Twenty 1G interfaces Twenty 10G interfaces Twenty 1/10G interfaces	0/0/0, 0/0/1, 0/0/2, 0/0/3, 0/0/4, 0/0/5, 0/0/6, 0/0/7, 0/0/8,0/0/9, 0/0/10, 0/0/11, 0/0/12,0/0/13, 0/0/14, 0/0/15, 0/0/16,0/0/17, 0/0/18, 0/0/19
Bay 1 6xQSFP+	40/100G EPA	Six 40/100G interfaces active The bay can be used in these modes: • Six 40G interfaces Six 100G interfaces • Six 40/100G interfaces	0/1/0, 0/1/1, 0/1/2, 0/1/3, 0/1/4, 0/1/5

2 Software Package and Architecture

Topics:

- Software package and architecture
- Cisco IOS XE software
- Deploy IOS-XE and SD-WAN
- Software upgrade

Details the operating system's modular design, file systems, and boot processes. It guides users through CLI access methods, command modes, and procedures for upgrading software or switching between autonomous and controller modes.

Software package and architecture

Explains the modular design of Cisco IOS XE, including consolidated packages and individual subpackages. Learn how these components interact to improve system resiliency and resource allocation.

Consolidated packages

A consolidated package is a single image composed of individual software subpackage files that

- enables the Cisco Catalyst 8500 Series Edge Platforms to boot and operate using a single, bootable package file,
- includes a provisioning file within the consolidated package which can be used to boot the router with the complete consolidated package.
- uses the provisioning file for booting when subpackages are extracted or optional subpackages are selected, and
- simplifies software management by combining all necessary components into one file.

Individual software subpackages within a consolidated package

Individual subpackages are not available for separate download from Cisco.com. Users must download a consolidated package and extract the needed subpackages from it by using the command-line interface. If the router operates with individual subpackages instead of a complete consolidated package, it must be booted using a provisioning file. Every consolidated package contains a provisioning file, which is extracted along with the individual subpackages during the extraction process.

- **RPBase**: Provides the operating system software for the Route Processor.
- **RPControl**: Controls the control plane processes that interface between the IOS process and the rest of the platform.
- **RPAccess**: Exports processing of restricted components, such as Secure Socket Layer (SSL), Secure Shell (SSH), and other security features.
- **RPIOS**: Provides the Cisco IOS kernel, which is where IOS features are stored and run. Each consolidated package has a different RPIOS.
- **ESPBase**: Provides the ESP operating system and control processes, and the ESP software.

Provisioning files

Provisioning files manage the boot process when the Cisco Catalyst 8500 Series Edge Platforms is configured to run using individual subpackages or optional subpackages (such as the package for the Cisco WebEx Node Cisco Catalyst 8500 Series Edge Platforms Series).

When individual subpackages are being used to run the Cisco Catalyst 8500 Series Edge Platforms, the router has to be configured to boot the provisioning file. The provisioning file manages the bootup of each individual subpackage and the Cisco Catalyst 8500 Series Edge Platform assumes normal operation.

Provisioning files are extracted automatically when individual subpackage files are extracted from a consolidated package.

Provisioning files are not necessary for running the router using the complete consolidated package; if you want to run the router using the complete consolidated package, simply boot the router using the consolidated package file.

You must use the provisioning files to manage the boot process if you need to install optional subpackages. The provisioning file and individual subpackage files must be kept in the same directory. The provisioning filename can be renamed; the individual subpackage filenames cannot be renamed. After placing the provisioning file and the individual subpackage files in a directory and booting the router, it is highly advisable not to rename, delete, or alter any of these files. Renaming, deleting, or altering the files can lead to unpredictable router problems and behaviors.

Upgrade field programmable hardware devices

Starting in Cisco IOS XE Release 17.3.2, a hardware programmable package file used to upgrade field programmable hardware devices is released as needed . A package file is provided for the field programmable device to customers in cases where a field upgrade is required.

If the Cisco Catalyst 8500 Series Edge Platforms contains an incompatible version of the hardware programmable firmware, then that firmware may need to be upgraded.

Generally an upgrade is only necessary in cases where a system message indicates one of the field programmable devices on the Cisco Catalyst 8500 Series Edge Platforms needs an upgrade or a Cisco technical support representative suggests an upgrade.

Processes overview

Cisco IOS XE has numerous components that run entirely as separate processes on the Cisco Catalyst 8500 Series Edge Platforms. This modular architecture increases network resiliency by distributing operating responsibility among separate processes rather than relying on Cisco IOS software for all operations.

IOS as a process

In almost all previous Cisco router platforms, an overwhelming majority of the internal software processes are run using Cisco IOS memory.

The Cisco Catalyst 8500 Series Edge Platforms introduce a distributed software architecture that moves many operating system responsibilities out of the IOS process. In this architecture, IOS, which previously was responsible for almost all of the internal software processes, now runs as one of many Linux processes while allowing other Linux processes to share responsibility for running the router. This architecture allows for better allocation of memory so the router can run more efficiently.

Dual IOS processes

The Cisco Catalyst 8500 Series Edge Platforms introduces a dual IOS process model that allows for increased high availability at all times. In the event of an active IOS failure, the second IOS process immediately becomes the active IOS process with little to no service disruption

On Cisco Catalyst 8500 Series Edge Platforms configured with dual Route Processors, the second IOS process runs on the standby Route Processor.

The state of these dual IOS processes can be checked by entering the **show platform** command.

File systems

The table provides a list of file systems that can be seen on the Cisco Catalyst 8500 Series Edge Platforms.

Table 5: File Systems

File System	Description
bootflash:	The boot flash memory file system on the active RP.
cns:	The Cisco Networking Services file directory.
harddisk:	The hard disk file system on the active RP.
nvram:	Router NVRAM. You can copy the startup configuration to NVRAM or from NVRAM.
obfl:	The file system for Onboard Failure Logging files.

File System	Description
system:	The system memory file system, which includes the running configuration.
tar:	The archive file system.
tmpsys:	The temporary system files file system.
usb[0-1]:	The Universal Serial Bus (USB) flash drive file systems on the active RP.

Note

If you run into a file system not listed in the above table, enter the ? help option or see the copy command reference for additional information on that file system.

Autogenerated file directories and files

Any autogenerated file on the bootflash: directory should remain unchanged unless you receive explicit instructions from customer support. Changing these files may impact system performance in unpredictable ways.

The table provides a list and descriptions of autogenerated files on the Cisco Catalyst 8500 Series Edge Platforms.

Table 6: Autogenerated Files

File or Directory	Description
crashinfo files	A crashinfo file may appear in the bootflash: or harddisk: file system. These files provide descriptive information about a crash. They may be useful for tuning or troubleshooting purposes, but they are not part of router operations and can be erased without impacting router functioning
core directory	The storage area for.core files. If this directory is erased, it will automatically regenerate itself at bootup. The .core files in this directory can be erased without impacting any router functionality, but the directory itself should not be erased.
lost+found directory	This directory is created on bootup if a system check is performed. Its appearance is completely normal and does not indicate any issues with the router.
tracelogs directory	The storage area for trace files. Trace files are useful for troubleshooting, but they are not part of router operations and can be erased without impacting the router's performance.

Crashinfo, core, and trace files can be deleted. However, do not remove the core and tracelog directories, which are an automatic part of the harddisk: file system.

Cisco IOS XE software

Covers the fundamentals of the Cisco IOS XE software, including console and SSH access methods, CLI navigation (shortcuts, history, and command modes), file system management, and best practices for CLI session management.

Access Cisco IOS XE software

Use the console port to access the command line interface directly or when using Telnet.

Console connections

The CON port is an EIA or TIA-232 asynchronous, serial connection with no-flow control and an RJ-45 connector. The CON port is located on the front panel of the chassis.

Connect to the console port

This section explains how to connect to a console port.

Use this procedure when you need direct access to device configuration via the console port.

Procedure

1. Configure your terminal emulation software with these settings.
   - 9600 bits per second (bps),
   - eight data bits,
   - no parity, and
   - no flow control
2. Connect to the CON port using the RJ-45-to-RJ-45 cable and either the RJ-45-to-DB-25 DTE adapter or the RJ-45-to-DB-9 DTE adapter labeled Terminal.

Your terminal displays the device's console output, allowing direct access to device configuration.

Access the console interface

Use the console interface to directly manage and configure the router.

Follow these steps to access the console interface:

Procedure

1. Enter the following command.

   ```
   Router> enable
   ```

2. If the enable password has not been configured, proceed to Step 3. Otherwise, at the password prompt, enter your system password.

   ```
   Password: enablepass
   ```

   When your password is accepted, the privileged EXEC mode prompt is displayed.

   ```
   Router#
   ```

You now have access to the CLI in privileged EXEC mode. Enter the commands to complete your tasks.

3. If you enter the **setup** command, refer to *Using Cisco Setup Command Facility* .
4. To exit the console session, enter the **quit** command.

```
Router# quit
```

You successfully access and exit the privileged EXEC mode in the console interface.

Access the device console using SSH

This section describes how to access the console using SSH.

Secure Shell is a protocol that provides a secure remote access connection to network devices. To enable SSH support on the device, access the device console using SSH as described in these steps.

Before you begin

Ensure the device is reachable via IP and supports SSH.

Install SSH client software on your management workstation.

Procedure

---

1. Configure the device host name.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

Here, *host name* is the device host name or IP address.

2. Configure the DNS domain of the device.

```
Router(config)# ip domain name cisco.com
```

3. Generate an SSH key to be used with SSH.

```
Router(config)# crypto key generate rsa
The name for the keys will be: Router.xxx.cisco.com Choose the size of the
key modulus in the range
of 360 to 4096 for your General Purpose Keys. Choosing a key modulus greater
 than 512 may take a few
minutes.
How many bits in the modulus [512]: 1024 % Generating 1024 bit RSA keys, keys
 will be non-exportable
[OK] (elapsed time was 0 seconds)
Router(config)#
```

4. By default, the vtys? transport is Telnet. In this case, Telnet is disabled, and only SSH is supported.

```
Router(config)# line vty 0 4
xxx_lab(config-line)# transport input ssh
```

5. Create a username for SSH authentication and enable login authentication for your device.

```
Router(config)# username jsmith privilege 15 secret 0 p@ss3456
Router(config)# line vty 0 4
Router(config-line)# login local
```

6. Verify remote connection to the device using SSH.

You will establish a secure SSH session to the device console.

Remote CLI access methods

This section provides procedures to access the CLI from a remote console using Telnet.

- Enable console access to the device via Telnet
- Access a console interface using Telnet

Enable console access to the device via Telnet

This section describes the necessary steps and configuration requirements for enabling remote access to a device console using Telnet.

Before you begin

Ensure the device is configured to support remote access using Telnet over a TCP or IP network.

Configure the device's virtual terminal lines with the **line vty** command.

Set the vty lines to require user login and specify a password to secure remote access. For details about the **line vty** command, refer the Cisco IOS Terminal Services Command Reference document.

Use this task to set up remote console access for network management.

Procedure

---

1. To add a line password to the vty, specify a password with the **password** command when you configure the **login** command. If you are using authentication, authorization, and accounting (AAA), configure the **login authentication** command on the vty lines.
2. To ensure login is not disabled on the lines when using AAA authentication, configure the list with the **login authentication** command and also define the login list with the **aaa authentication login** command.

   For more information about AAA services, see the Cisco IOS XE Security Configuration Guide: Secure Connectivity and the Cisco IOS Security Command Reference documents. For more information about the **login line-configuration** command, see the Cisco IOS Terminal Services Command Reference document.
3. Ensure the device has a configured hostname or an IP address before attempting Telnet access.

   For more information about the requirements for connecting to the device using Telnet, information about customizing your Telnet services, and using Telnet key sequences, see the Cisco IOS Configuration Fundamentals Configuration Guide.

The device is now configured to allow remote console access via Telnet using password or AAA authentication.

Access a console interface using Telnet

This section describes how to access a console on your device using Telnet.

Use this procedure when you need to manage a device from your terminal or PC via Telnet.

Follow these steps to access the console interface:

Before you begin

Ensure Telnet is enabled on the device.

Procedure

From your terminal or PC, enter one of these commands:

- **connect host** *port keyword*]
- **telnet host** *port keyword*]

Here, *host* refers to the device hostname or IP address, *port* is a decimal port number (23 is the default), and *keyword* is a supported keyword. For more information about these commands, see the Cisco IOS Terminal Services Command Reference document.

> **Note**
>
> When you use an access server, specify a valid port number, such as **telnet 198.51.100.2 2004**, in addition to the hostname or IP address.

This example shows how to use the **telnet** command to connect to a device named **router**:

```
unix_host% telnet router
Trying 198.51.100.2...
Connected to 198.51.100.2.
Escape character is '^]'.
unix_host% connect
```

You are connected to the console interface of the remote device.

USB serial console ports

The router provides an additional mechanism for configuring the system: a type B miniport USB serial console that supports remote administration of the router using a type B USB-compliant cable. Refer to the 'Connecting to a Console Terminal or Modem' section for detailed instructions.

- Hardware Installation Guide for Cisco Catalyst 8300 Series Edge Platform
- Hardware Installation Guide for Cisco Catalyst 8200 Series Edge Platforms

Keyboard shortcuts

This section explains how you can use different keyboard shortcuts.

A keyboard shortcut is a key combination or sequence that

- allows commands to be entered without regard to case sensitivity,
- enables the use of abbreviated commands and parameters, and

- requires that abbreviations contain enough unique letters to distinguish them from any other available commands or parameters.

This table lists the keyboard shortcuts for entering and editing commands.

Table 7: Keyboard shortcuts

Key Name	Purpose
Ctrl-B or the **Left** Arrow key[1]	Move the cursor back one character.
Ctrl-F or the **Right Arrow key**[1]	Move the cursor forward one character.
Ctrl-A	Move the cursor to the beginning of the command line.
Ctrl-E	Move the cursor to the end of the command line.
Esc B	Move the cursor back one word.
Esc F	Move the cursor forward one word.

History buffer

This section describes the history substitution commands.

The history buffer is a CLI feature that:

- stores the last 20 commands you entered,
- enables history substitution so you can access previous commands without retyping them, and
- uses special abbreviated commands to quickly recall and reuse stored entries.

This table lists the history substitution commands.

Table 8: History substitution commands

Command	Purpose
Ctrl-P or the **Up Arrow** key[1]	Recalls commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall successively older commands.
Ctrl-N or the **Down Arrow** key[1]	Returns to more recent commands in the history buffer after recalling commands with **Ctrl-P** or the **Up Arrow** key.
Router# show history	While you are in EXEC mode, lists the last few commands you entered.

[1] The arrow keys function only on ANSI-compatible terminals such as VT100s.

Command modes

This is a short description on command modes.

A command mode is a CLI access level that

- determines which IOS XE commands are available,
- secures and isolates different configuration functions, and

- defines the privileges assigned to each user or process within the operating system.

Cisco IOS XE provides the same command modes as traditional Cisco IOS and supports these modes only in autonomous mode. You access Cisco IOS XE software through the CLI, which divides commands into several modes. The commands available to you always depend on your current mode. When you enter a question mark (**?**) at the CLI prompt, you can see a list of commands available in that mode.

When you log in to the CLI, you are in user EXEC mode. User EXEC mode contains only a limited subset of commands. To have access to all commands, you must enter privileged EXEC mode, normally by using a password. From privileged EXEC mode, you can issue any EXEC command–user or privileged mode–or you can enter global configuration mode. Most EXEC commands are one-time commands. For example, **show** commands show important status information, and **clear** commands clear counters or interfaces. The EXEC commands are not saved when the software reboots.

Configuration modes allow you to make changes to the running configuration. If you later save the running configuration to the startup configuration, these changed commands are stored when the software is rebooted. To enter specific configuration modes, you must start at global configuration mode. From global configuration mode, you can enter interface configuration mode and a variety of other modes, such as protocol-specific modes.

ROM monitor mode is a separate mode used when the Cisco IOS XE software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode.

This table describes how to access and exit various common command modes of the Cisco IOS XE software. It also shows examples of the prompts displayed for each mode.

Table 9: Accessing and exiting command modes

Command Mode	Access Method	Prompt	Exit Method
User EXEC	Log in.	`Router>`	Enter the **logout** command.
Privileged EXEC	From user EXEC mode, enter the **enable** command.	`Router#`	To return to user EXEC mode, use the **disable** command.
Global configuration	From privileged EXEC mode, enter the **configure terminal** command.	`Router(config)#`	To return to privileged EXEC mode from global configuration mode, enter the **exit** or **end** command.
Interface configuration	From global configuration mode, specify an interface using an **interface** command.	`Router(config if)#`	To return to global configuration mode, use the ui command. To return to privileged EXEC mode, enter the **end** command.

Command Mode	Access Method	Prompt	Exit Method
Diagnostic	The device boots up or accesses diagnostic mode in the following scenarios: • In some cases, diagnostic mode will be reached when the Cisco IOS process or processes fail. In most scenarios, however, the device will reload. • A user-configured access policy is configured using the **transport-map** command that directs a user into diagnostic mode. • A break signal (**Ctrl-C**, **Ctrl-Shift-6**, or the **send break** command) is entered and the device is configured to go to diagnostic mode when the break signal is received.	`Router(diag)#`	If failure of the Cisco IOS process is the reason for entering diagnostic mode, the Cisco IOS problem must be resolved and the device rebooted to get out of diagnostic mode. If the device is in diagnostic mode because of a transport-map configuration, access the device through another port or by using a method that is configured to connect to the Cisco IOS CLI.
ROM monitor	From privileged EXEC mode, enter the **reload** EXEC command. Press the **Break** key during the first 60 seconds while the system is booting.	`rommon#>`	To exit ROM monitor mode, manually boot a valid image or perform a reset with autoboot set so that a valid image is loaded.

Diagnostic mode

Use diagnostic mode to troubleshoot device issues and access key commands when the main operating system is unavailable..

A diagnostic mode is a device operation state that

- offers a comprehensive user interface for troubleshooting, surpassing the limited access methods of previous devices
- enables diagnosis and troubleshooting of Cisco IOS problems even when the Cisco IOS process is not functioning properly, and
- provides diagnostic commands that are also accessible in privileged EXEC mode when the device is operating normally.

The device enters diagnostic mode in several scenarios. If the IOS process fails, your device might boot into diagnostic mode automatically, or it may reset first based on device configuration. Additionally, if a user-configured access policy uses the **transport-map** command to direct access into diagnostic mode, the device follows this policy. The device also enters diagnostic mode when it receives a break signal (such as **Ctrl-C** or **Ctrl-Shift-6**) during access, provided it is configured to do so in response to the break

Additional reference information

In diagnostic mode, you have have access to a subset of commands from user EXEC mode. These commands allow users to:

- Inspect various states on the device, including the IOS state.
- Replace or roll back the configuration.
- Provide methods of restarting the IOS or other processes.
- Reboot hardware, such as the entire device, a module, or possibly other hardware components.
- Transfer files to or from the device using remote access methods such as FTP, TFTP, and SCP.

CLI help commands

Use these CLI commands to access help or list available options and arguments.

Entering a question mark (**?**) at the CLI prompt displays a list of commands available for each command mode. To get help that is specific to a command mode, a command, a keyword, or an argument, use one of the following commands.

Table 10: Help Commands

Command	Comment
help	Provides a brief description of the help system in any command mode.
abbreviated-command-entry?	Provides a list of commands that begin with a particular character string. **Note** There is no space between the command and the question mark.
abbreviated-command-entry <Tab>	Completes a partial command name.
?	Lists all the commands that are available for a particular command mode.
command ?	Lists the keywords or arguments that you must enter next on the command line. **Note** There is a space between the command and the question mark.

Command help options and symbols

The Cisco IOS XE software provides command-line help so you can enter commands accurately.

- Entering a question mark (**?**) at the CLI prompt or after part of a command displays a list of available keywords and arguments. For example, if you are in global configuration mode and want to see all the keywords and arguments for the **arap** command, you should type **arap ?**.
- Command help output shows optional or required arguments, along with brief descriptions.
- The **<cr>** symbol indicates that pressing **Enter** will complete the command.
- If **<cr>** appears at the end of help output, all previous options are optional. If **<cr>** is not displayed, further arguments or keywords are required.
- The **<cr>** symbol refers to the carriage return key, which is labeled **Enter** on most modern keyboards. On older keyboards, the carriage return key is the **Return** key.

Table 11: Finding command options

Command	Comment
`Router> enable` `Password: <password>` `Router#`	Enter the **enable** command and password to access privileged EXEC commands. You are in privileged EXEC mode when the prompt changes from '>' to '#', for example, for example, `Router>` to `Router#`
`Router> configure terminal` `Enter configuration commands, one per line.` `End with CNTL/Z.` `Router(config)#`	Enter the **configure terminal** privileged EXEC command to enter global configuration mode. You are in global configuration mode when the prompt changes to `Router(config)#`
`Router(config)# interface GigabitEthernet ?` `<0-1> GigabitEthernet interface number` `Router(config)# interface GigabitEthernet 0/?` `<0-5> Port Adapter number` `Router (config)# interface GigabitEthernet 0/0/?` `<0-63> GigabitEthernet interface number` `Router (config)# interface GigabitEthernet0/0/1?` `. <0-5>` `Router(config-if)#`	Enter interface configuration mode by specifying the interface that you want to configure, using the **interface GigabitEthernet** global configuration command. Enter **?** to display what you must enter next on the command line. When the **<cr>** symbol is displayed, you can press **Enter** to complete the command. You are in interface configuration mode when the prompt changes to `Router(config-if)#`.
	Enter **?** to display a list of all the interface configuration commands available for the interface. This example shows only some of the available interface configuration commands.

Command	Comment
`Router(config-if)# ?` `Interface configuration commands:` `  .` `  .` `  .` `  ip  Interface Internet Protocol` `  config commands` `  keepalive  Enable keepalive` `  lan-name  LAN Name command` `  llc2  LLC2 Interface Subcommands` `  logging  Configure logging for interface` `  mls  mls router sub/interface commands` `  mpoa  MPOA interface configuration commands` `  mtu  Set the interface MTU` `  no  Negate a command or set its defaults` `  ntp  Configure NTP` `  .` `  .` `  .` `Router(config-if)#`	
`Router(config-if)# ip ?` `Interface IP configuration subcommands:` `  access-group  Specify access control for packets` `  accounting  Enable IP accounting on this interface` `  address  Set the IP address of an interface` `  authentication  authentication subcommands` `  cgmp  Enable/disable CGMP` `  dvmrp  DVMRP interface commands` `  hello-interval  Configures IP-EIGRP hello interval` `  hold-time  Configures IP-EIGRP hold time` `  .` `  .` `  .` `Router(config-if)# ip`	Enter the command that you want to configure for the interface. This example uses the **ip** command. Enter **?** to display what you must enter next on the command line. This example shows only some of the available interface IP configuration commands.
`Router(config-if)# ip address ?` `  A.B.C.D  IP address` `  negotiated  IP Address negotiated over PPP` `Router(config-if)# ip address`	

Command	Comment
	Enter the command that you want to configure for the interface. This example uses the **ip address** command. Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP address or the **negotiated** keyword. A carriage return (<cr>) is not displayed. Therefore, you must enter additional keywords or arguments to complete the command.
`Router(config-if)# ip address 198.51.100.5 ?` `  A.B.C.D  IP subnet mask` `Router(config-if)# ip address 198.51.100.5`	Enter the keyword or argument that you want to use. This example uses the 198.51.100.5 IP address. Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP subnet mask. <cr> is not displayed. Therefore, you must enter additional keywords or arguments to complete the command.
`Router(config-if)# ip address 198.51.100.5 255.255.255.0 ?` `  secondary  Make this IP address a secondary address` `  <cr>` `Router(config-if)# ip address 198.51.100.5 255.255.255.0`	Enter the IP subnet mask. This example uses the 255.255.255.0 IP subnet mask. Enter **?** to display what you must enter next on the command line. In this example, you can enter the **secondary** keyword, or you can press **Enter**. **<cr>** is displayed. Press **Enter** to complete the command, or enter another keyword.
`Router(config-if)# ip address 198.51.100.5 255.255.255.0` `Router(config-if)#`	Press **Enter** to complete the command.

CLI command forms

A CLI command form is a configuration command variation that:

- typically has a **no** form to disable a function and a standard form to enable or re-enable a function,
- often includes a default form that resets the command to its default setting using the **default** keyword, and
- is fully documented in Cisco IOS Software command reference publications, which explain the syntax and functions of the standard, **no**, and **default** forms.

Additional reference information

The Cisco IOS software command reference publications document the syntax and effects of both the **no** and **default** forms for configuration commands. To see available **default** commands on your system, enter **default ?** in the appropriate command mode.

> **Note**
>
> To disable IP routing (which is enabled by default), use the **no ip routing** command.To re-enable IP routing, use the **ip routing** command.
>
> To reset a command to its default setting, use the default command-name syntax.

Save configuration changes

To ensure your configuration changes are retained after a software reload or power outage, you must save the running configuration to the startup configuration. This task writes your current settings to NVRAM, making them persistent across device reboots.

Before you begin

Make sure you have the necessary privileges to execute configuration commands in the CLI.

Follow these steps to save configuration changes:

Procedure

Use the **copy running-config startup-config** command to save your configuration changes to the startup configuration so that the changes will not be lost if the software reloads or a power outage occurs.

Example

```
Router# copy running-config startup-config
Building configuration...
```

It may take a few minutes to save the configuration. After the configuration has been saved, the following output is displayed.

```
[OK]
Router#
```

This task saves the configuration to the NVRAM.

Your configuration changes are saved to NVRAM and retained across device reboots or power loss.

Configuration files

This is a short description.

A configuration file is a data file that stores and manages the operational settings of a Cisco router. Key details about configuration file storage and maintenance include:

- The startup configuration file resides in the NVRAM: file system.
- The running configuration files are stored in the system: file system.
- Maintains a consistent storage arrangement across different Cisco router platforms.

Additional reference information

Users should routinely back up the startup configuration file on any Cisco router. Copy the startup configuration file from NVRAM to another file system on the router and to a network server to back it up. TA backup makes it easy for you to recover the startup configuration file if the file in NVRAM becomes unusable.

The **copy** command can be used to back up startup configuration files.

To learn more about managing configuration files, refer the "Managing Configuration Files" section in the Cisco IOS XE Configuration Fundamentals Configuration Guide.

Filtering output from the show and more commands

You can search and filter the output of **show** and **more** commands. This functionality is useful if you need to review large amounts of output or if you want to exclude information that is not relevant.

To use this functionality, enter a **show** or **more** command followed by the pipe character (|); one of the keywords **begin**, **include**, or **exclude**, and a regular expression on which you want to search or filter (the expression is case sensitive).

show *command* **append begin exclude include redirect section tee** *regular-expression*

The output matches certain lines of information in the configuration file.

Example

In this example, a modifier of the **show interface** command (**include protocol**) is used to provide only the output lines in which the expression **protocol** is displayed:

```
Router# show interface | include protocol
GigabitEthernet0/0/0 is administratively down, line protocol is down
     0 unknown protocol drops
GigabitEthernet0/0/1 is administratively down, line protocol is down
     0 unknown protocol drops
GigabitEthernet0/0/2 is administratively down, line protocol is down
     0 unknown protocol drops
GigabitEthernet0/0/3 is administratively down, line protocol is down
     0 unknown protocol drops
GigabitEthernet0 is up, line protocol is up
     0 unknown protocol drops
Loopback0 is up, line protocol is up
     0 unknown protocol drops
```

Save configuration changes before powering off

This section outlines how to preserve configuration changes before shutdown.

Ensure you save any configuration changes you want to keep by using the **copy running-config startup-config** command before powering off the device.

If you power off the device by switching the power supply to Off without saving, you will lose any changes made to the running configuration since the last save to NVRAM. When you power the device back on, it loads the configuration stored in NVRAM. Only saved settings are retained after startup.

Cisco software images

This is a short description.

The Cisco IOS XE Software is packaged in feature sets consisting of software images that provide the following benefits:

- contains a group of feature sets that are available for a specific platform depends on which Cisco software images are included in a release.

- is distributed in multiple feature sets for various deployment needs, and
- supports designated Cisco platforms.

Additional reference information

To identify which software images are available for a particular platform, or whether a feature is present in a given image, use the Cisco Feature Navigator or see the Release Notes for Cisco IOS XE.

Cisco feature navigators

This is a short description.

A Cisco feature navigator is an online tool that

- provides details about platform support for Cisco devices,
- enables you to identify which Cisco IOS XE software images support specific features, and
- facilitates software image and feature set searches without requiring a Cisco.com account.

Additional reference information

To find information about platform support details and software image support, refer to the Cisco Feature Navigator.

Software advisor

This short description summarizes the Software advisor tool.

The Software advisor tool is a Cisco resource that:

- allows you to check if a feature is supported in a specific Cisco IOS XE release,
- helps you locate the software documentation for a given feature, and
- enables you to verify the minimum software requirements of Cisco IOS XE software based on the hardware installed on your device.

Cisco provides the Software Advisor tool, which is available in the Tools and Resources section.

Release notes

This is a short description on release notes.

The Release Notes document for Cisco Catalyst 8000 Series Edge Platforms is a release-specific resource that:

- provides memory recommendations for the platform,
- lists open and resolved severity 1 and 2 caveats, and
- focuses on the current release without offering cumulative feature information from previous releases

For cumulative feature information, refer to the Cisco Feature Navigator at http://www.cisco.com/go/cfn/.

CLI session management

An inactivity timeout is configurable and can be enforced. Session locking provides protection from two users overwriting changes that each other has made. To prevent an internal process from using all the available capacity, some spare capacity is reserved for CLI session access. For example, this allows a user to remotely access the router.

Best practice for CLI session management

Follow these best practices to ensure secure and effective CLI session management:

- Configure inactivity timeouts to automatically close idle CLI sessions and enhance security.
- Lock your CLI session to prevent others from making configuration changes at the same time.
- Reserve enough system capacity for CLI session access so that you and other administrators can always access the system, even under high load.

Configure the CLI session timeout

This section describes how to configure the CLI session timeout.

Procedure

1. **`configure terminal`**

   Enters global configuration mode

2. **`line console 0`**
3. **`session-timeout`** *`minutes`*

   The value of *`minutes`* sets the amount of time that the CLI waits before timing out. Setting the CLI session timeout increases the security of a CLI session. Set a value of 0 for *`minutes`* to disable session timeout.

4. **`show line console 0`**
   Verifies the value to which the session timeout has been set, which is shown as the value for `"Idle Session"`.

The device automatically ends the CLI session after the specified period of inactivity, increasing security by preventing unauthorized access to unattended sessions.

Lock a CLI session

This section describes how to lock your CLI session.

Use this task to lock your CLI session when stepping away, requiring a temporary password for re-entry.

Before you begin

To configure a temporary password on a CLI session, first configure the line using the **lockable** command, and then use the **lock** command in EXEC mode. When the line is configured as **lockable**, you can use the **lock** command to assign a temporary password.

Procedure

1. `Router#` **`configure terminal`**

   Enters global configuration mode.

2. Enter the line where you want to be able to use the **lock** command.
   `Router(config)#` **`line console 0`**
3. `Router(config)#` **`lockable`**

   Enables the line to be locked.

4. `Router(config)#` **`exit`**

5. `Router# lock`
   The system prompts you for a password, which you must enter twice.

```
Password: <password>
Again: <password>
Locked
```

Your CLI session is locked. Accessing the session again requires the temporary password you set.

Deploy IOS-XE and SD-WAN

Explains how to manage device operation modes using a single software image. Learn how to switch between autonomous and controller modes to enable SD-WAN functionality or utilize the Plug and Play discovery process for automated mode selection. You can use the universalk9 image to deploy both Cisco IOS XE SD-WAN and Cisco IOS XE on Cisco IOS XE devices. This helps in seamless upgrades of both the SD-WAN and non SD-WAN features and deployments.

Autonomous or controller mode

Access the Cisco IOS XE and Cisco IOS XE SD-WAN functionality through Autonomous and Controller execution modes, respectively. The Autonomous mode is the default mode for the routers and includes the Cisco IOS XE functionality. To access Cisco IOS XE SD-WAN functionality switch to the Controller mode.

For more information, see Cisco Catalyst SD-WAN Getting Started Guide.

Switch between controller and autonomous modes

The default mode of the device is autonomous mode. Use the **controller-mode** command in Privileged EXEC mode to switch between controller and autonomous modes.

The **controller-mode enable** command switches the device to controller mode.

The **controller-mode disable** command switches the device to autonomous mode.

For information see Cisco SD-WAN Getting Started Guide

PnP discovery process

You can use the existing Plug and Play Workflow to determine the mode of the device.

The PnP-based discovery process determines the mode in which the device operates, based on the controller discovery and initiates a mode change, if required. This discovery is based on the controller profile attached to the device UID in the smart account/virtual account. The mode change results in a reboot of the device. Once reboot is complete, the device performs appropriate discovery process.

Plug and Play (PnP) deployment include these discovery process scenarios:

Boot up mode	Discovery process	Mode change
Automomous	Plug and Play Connect Discovery or on-premise plug and play server discovery	No Mode change
Controller	Plug and Play Connect Discovery or on-premise plug and play server discovery	Mode change to autonomous mode

Software upgrade

Use this procedure to update the ROMMON and IOS XE images simultaneously. This process ensures the device supports new hardware features and maintains critical software dependencies across the platform.

Perform a software upgrade

Use this procedure when both ROMMON and IOS need to be updated, such as when introducing new hardware features or addressing critical software dependencies.

Procedure

To upgrade the ROMMON and IOS at the same time, perform the steps given below:

1. Copy the XE image to the router and configure the boot system to point to the new image.
2. Copy the ROMMON package to the router and perform the ROMMON upgrade.
3. Reload the router and verify that it boots to the IOS prompt on the new XE image.
4. Verify that the new ROMMON image was successfully installed using a show platform.

The router operates with the new ROMMON and XE images.

3 Factory Reset

Topics:

- Factory reset

Outlines the process for restoring a device to its original functional state by clearing configurations and personal data. It specifies the commands required for secure resets and explains which data is retained or erased during the procedure.

Factory reset

Explains the process of clearing configurations and personal data to restore the device to a functional default state. Use this feature when decommissioning, repurposing, or recovering a compromised router.

About factory reset

This section explains what factory reset is and what data are retained and deleted after a factory reset.

Factory Reset is a process of clearing the current running and start-up configuration information on a device, and resetting the device to an earlier, fully-functional state.

The factory reset process uses the **factory-reset all** command to take backup of existing configuration, and then reset the router to an earlier, fully functional state. The duration of the factory reset process is dependent on the storage size of the router. It can vary between 30 minutes on a C8500 consolidated platform, and up to 3 hours on a high availability setup.

From Cisco IOS XE Bengaluru 17.6 release and later, you can use the **factory-reset all secure** command to reset the router and securely clear the files stored in the bootflash memory.

Table 12: Data erased or retained during Factory Reset

Command Name	Data Erased	Data Retained
factory-reset all secure	Non-volatile random-access memory (NVRAM) data	Data from remote field-replaceable units (FRUs).
	OBFL (Onboard Failure Logging) logs	Value of configuration register **Important** From Cisco IOS XE 17.14.1a, the value of the configuration register can be erased using the **factory-reset all secure** command on C8500L-8S4X.
	Licenses	Contents of USB
	User data, startup, and running configuration	Credentials (Secure Unique Device Identifier [SUDI] certificates, public key infrastructure (PKI) keys, and FIPS-related keys)
	ROMMON variables	
	All writeable file systems and personal data. **Note** If the current boot image is a remote image or stored on a USB, NIM-SSD, or such, ensure that you take a backup of the image before performing factory reset.	
factory-reset keep-licensing-info	• License Boot level configuration • Throughput level configuration • Smart license transport type • Smart license URL data	

Command Name	Data Erased	Data Retained
		• Real User Monitoring (RUM) Reports (open/unacknowledged license usage report) • Usage reporting details (last ACK received, next ACK scheduled, last/next report push) • Unique Device Identification (UDI) trust codes • Customer policy received from CSSM • SLAC, SLR authorization codes return codes • Factory installed purchase information

After the factory reset process is complete, the router reboots to ROMMON mode. If you have the zero-touch provisioning (ZTP) capability setup, after the router completes the factory reset procedure, the router reboots with ZTP configuration.

Software and hardware support for factory reset

This feature is supported on all Cisco Catalyst 8500 and 8500L Series Edge Platforms. Factory Reset process is supported on standalone routers as well as on routers configured for high availability.

Prerequisites for performing factory reset

Ensure that these prerequisites are met before performing a Factory Reset.

- Ensure that all the software images, configurations and personal data is backed up before performing factory reset.
- Ensure that there is uninterrupted power supply when factory reset is in progress.
- The factory reset process takes a backup of the boot image if the system is booted from an image stored locally (bootflash or hard disk). If the current boot image is a remote image or stored on an USB, NIM-SSD or such, ensure that you take a backup of the image before performing factory reset.
- The **factory-reset all secure** command erases all files, including the boot image, even if the image is stored locally. If the current boot image is a remote image or stored on a USB, NIM-SSD, or such, ensure that you take a backup of the image before performing secure factory reset.
- Ensure that ISSU/ISSD (In- Service Software Upgrade or Downgrade) is not in progress before performing factory reset.

Restrictions for performing a factory reset

These are some of the restrictions to know before you perform a Factory Reset.

- Any software patches that are installed on the router are not restored after the factory reset operation.
- If the factory reset command is issued through a Virtual Teletype (VTY) session, the session is not restored after the completion of the factory reset process.

When to perform factory reset

This topics outlines some of the scenarios when you can perform a Factory Reset.

- Return Material Authorization (RMA): If a router is returned back to Cisco for RMA, it is important that all sensitive information is removed.
- Router is compromised: If the router data is compromised due to a malicious attack, the router must be reset to factory configuration and then reconfigured once again for further use.
- Repurposing: The router needs to be moved to a new topology or market from the existing site to a different site.

How to perform a factory reset

Procedure

1. Log in to a Cisco Catalyst 8500 or 8500L device.

   Important

   If the current boot image is a remote image or is stored in a USB or a NIM-SSD, ensure that you take a backup of the image before starting the factory reset process.

2. This step is divided into two parts (a and b). If you need to retain the licensing information while performing the **factory-reset** command, follow step 2. a. If you do not need to retain the licensing information and want all the data to be erased, perform step 2. b.

   a) Execute **factory-reset keep-licensing-info** command to retain the licensing data.

   The system displays the following message when you use the **factory-reset keep-licensing-info** command:

```
Router# factory-reset keep-licensing-info
```

```
The factory reset operation is irreversible for Keeping license usage.
Are you sure? [confirm]
This operation may take 20 minutes or more. Please do not power cycle.

Dec 1 20:58:38.205: %PMAN-5-EXITACTION: R0/0: pvp: Process manager is
exiting: process exit with
reload chassis code
/bootflash failed to mount
Dec 01 20:59:44.264: Factory reset operation completed.
Initializing Hardware ...

Current image running: Boot ROM1

Last reset cause: LocalSoft

ISR4331/K9 platform with 4194304 Kbytes of main memory
rommon 1
```

   b) Execute the **factory-reset all secure 3-pass** command to securely erase all data.

The system displays the following message when you use the **factory-reset all secure 3-pass** command:

```
Router# factory-reset all secure 3-pass
```

```
The factory reset operation is irreversible for securely reset all.
Are you sure? [confirm]
This operation may take hours. Please do not power cycle.

*Jun 19 00:53:33.385: %SYS-5-RELOAD: Reload requested by Exec.
Reload Reason: Factory Reset.Jun 19 00:53:42.856: %PMAN-5-EXITACTION:

Enabling factory reset for this reload cycle
  Jun 19 00:54:06.914: Factory reset secure operation. Write 0s.
Please do not power cycle.

  Jun 19 02:40:29.770: Factory reset secure operation completed.
Initializing Hardware ....
```

3. Enter **confirm** to proceed with the factory reset.

> **Note**
>
> The duration of the factory reset process depends on the storage size of the router. It can extend between 30 minutes and up to 3 hours on a high availability setup. If you want to quit the factory reset process, press the **Escape** key.

What happens after a factory reset

After the factory reset is successfully completed, the router boots up. However, before the factory reset process started, if the configuration register was set to manually boot from ROMMON, the router stops at ROMMON.

After you configure Smart Licensing, execute the **#show license status** command, to check whether Smart Licensing is enabled for your instance.

> **Note**
>
> If you had Specific License Reservation enabled before you performed the factory reset, use the same license and enter the same license key that you received from the smart agent.

4 Security-Enhanced Linux

Topics:

- Security-Enhanced Linux

Introduces the mandatory access control architecture used to safeguard system integrity and prevent unauthorized resource access. It describes how to configure different security modes and troubleshoot policy violations using system logs.

Security-Enhanced Linux

Explains the mandatory access control architecture that protects system integrity on Cisco IOS XE platforms. Learn how this mechanism prevents tampering and limits damage from potentially malicious or flawed applications.

Security-Enhanced Linux (SELinux) is a solution that incorporates a strong, flexible Mandatory Access Control (MAC) architecture into Cisco IOS-XE platforms through a combination of a Linux kernel security module and system utilities.

SELinux provides an enhanced mechanism to enforce the separation of information based on confidentiality and integrity requirements. This approach helps prevent tampering and bypassing of application security mechanisms while limiting damage from malicious or flawed applications.

Security-Enhanced Linux Modes

SELinux can operate either in **Permissive mode** or **Enforcing mode** when enabled on a system.

- **Permissive mode** : SELinux does not enforce the policy and only generates system logs for any denials caused by policy violations. Operations are logged for resource access policy violations but not denied.
- **Enforcing mode** : SELinux policy is enabled and enforced. It denies resource access based on the access policy rules, and generates system logs.

  SELinux is enabled in **Enforcing mode** by default on supported Cisco IOS XE platforms. In the **Enforcing mode**, any system resource access that does not have the necessary allow policy is treated as a violation, and the operation is denied. The violating operation fails when a denial occurs, and system logs are generated. In **Enforcing mode**, the solution works in access-violation prevention mode.

Prerequisites

There are no specific prerequisites for this feature.

Restrictions

There are no specific restrictions for this feature.

Configure Security-Enhanced Linux in EXEC Mode

Configuring SELinux in EXEC mode denies resource access based on the access policy rules, and generates system logs. Use this example to configure SELinux in EXEC mode:

Procedure

---

set platform software selinux {default | enforcing | permissive}

Configuring SELinux in EXEC mode denies resource access based on the access policy rules, and generates system logs.

Example

```
Device# set platform software selinux ?

default     Set SELinux mode to default
enforcing   Set SELinux mode to enforcing
permissive  Set SELinux mode to permissive
```

Configure Security-Enhanced Linux in CONFIG Mode

Configuring SELinux in CONFIG mode means setting the policy to either **enforcing** or **permissive** state. Use this example to configure SELinux in CONFIG mode:

Use this example to configure SELinux in configuration mode:

Procedure

---

platform security selinux {enforcing | permissive}

Configuring SELinux in CONFIG mode means setting the policy to either **enforcing** or **permissive** state.

Example

```
Device(config)# platform security selinux

enforcing   Set SELinux policy to Enforcing mode
permissive  Set SELinux policy to Permissive mode

Device(config)# platform security selinux permissive

Device(config)#
*Oct 20 21:52:45.155: %IOSXE-1-PLATFORM: R0/0:
SELINUX_MODE_PROG: Platform Selinux confinement mode downgraded to permissive!

Device(config)#
```

SYSLOG message reference

This section covers details about syslog messages for SELinux events.

Facility-Severity-Mnemonic	%SELINUX-1-VIOLATION
Severity-Meaning	Alert Level Log
Message	N/A
Message Explanation	Resource access was made by the process for which a resource access policy does not exist. The operation was flagged, and resource access was denied. A system log was generated with information that process resource access has been denied.
Component	SELINUX

Facility-Severity-Mnemonic	%SELINUX-1-VIOLATION
Recommended Action	Contact Cisco TAC with the following relevant information as attachments: • The exact message as it appears on the console or in the system • Output of the **show tech-support** command (text file) • Archive of Btrace files from the box using the following command: **request platform software trace archive target <URL>** • Output of the **show platform software selinux** command

This example shows sample syslog messages:

Example 1:

```
*Nov 14 00:09:04.943: %SELINUX-1-VIOLATION: R0/0: audispd: type=AVC
msg=audit(1699927057.934:129): avc: denied { getattr } for pid=5899 comm="ls"

path="/root/test" dev="rootfs" ino=25839
scontext=system_u:system_r:polaris_iosd_t:s0
tcontext=system_u:object_r:admin_home_t:s0 tclass=file permissive=0
```

Example 2:

```
*Nov 14 00:09:04.947: %SELINUX-1-VIOLATION: R0/0: audispd: t type=AVC
msg=audit(1699927198.486:130): avc: denied { write } for pid=6012 comm="echo"

path="/root/test" dev="rootfs" ino=25839
scontext=system_u:system_r:polaris_iosd_t:s0
tcontext=system_u:object_r:admin_home_t:s0 tclass=file permissive= 0
```

Verify Security-Enhanced Linux enablement

Use the **show platform software selinux** command to view the SELinux configuration mode:

```
Device# show platform software selinux
========================================
IOS-XE SELINUX STATUS
========================================
SElinux Status :    Enabled
Current Mode :      Enforcing
Config file Mode :  Enforcing
```

Troubleshoot Security-Enhanced Linux

If there is an instance of an SELinux violation on your device or network, please reach out to Cisco TAC with these details:

- The message exactly as it appears on the console or in the system log. For example:

```
device#request platform software trace archive target
    flash:selinux_btrace_logs
```

- Output of the **show tech-support** command (text file)
- Archive of Btrace files from the box using this command:

  request platform software trace archive target <URL>
- Output of the **show platform software selinux** command

5 High Availability

Topics:

- High Availability

Explains network resiliency solutions, focusing on interchassis redundancy and rapid failure detection. It provides configuration steps for backup pairs and bidirectional forwarding detection to ensure continuous service during hardware or link disruptions.

High Availability

Explains network resiliency solutions, focusing on interchassis redundancy and rapid failure detection. It provides configuration steps for backup pairs and bidirectional forwarding detection to ensure continuous service during hardware or link disruptions.

The Cisco high availability technology is a network resiliency solution that

- enables rapid recovery from disruptions,
- ensures fault transparency to users and network applications, and
- maximizes network uptime through integrated hardware and software design.

These mechanisms that help maintain continuous network operations even during failure events or maintenance.

Interchassis high availability

This is a short description.

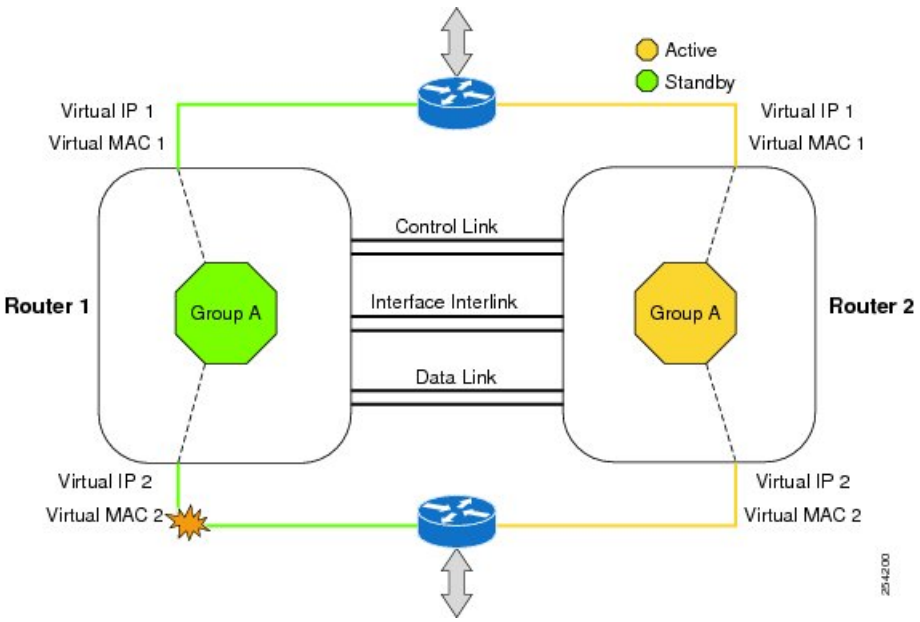
The Interchassis High Availability feature is also known as the box-to-box redundancy feature is a system redundancy solution that

- enables two devices to form a backup pair for high-availability operation,
- automatically monitors device status to detect failures, and
- seamlessly transfers call signaling and media forwarding to a standby device during failover.

Redundancy groups

Groups of redundant interfaces are known as redundancy groups. The figure depicts the active-standby device scenario. It shows how the redundancy group is configured for a pair of devices that have a single outgoing interface.

Figure 1: Redundancy group configuration



The devices are connected by a configurable control link and data synchronization link. The control link is used to communicate the status of the devices. The data synchronization link transfers stateful information to synchronize the

database for the calls and media flows. Each pair of redundant interfaces is configured with the same unique ID number, also known as the RII.

Prerequisites

Before configuring interchassis high availability, ensure that you meet all these requirements:

- The active device and the standby device must run on the identical version of the Cisco IOS XE software.
- The active device and the standby device must be connected through an L2 connection for the control path.
- Either the Network Time Protocol (NTP) must be configured or the clock must be set identical on both devices to allow timestamps and call timers to match.
- Virtual Routing and Forwarding (VRF) must be defined in the same order on both active and standby devices for an accurate synchronization of data.
- The latency times must be minimal on all control and data links to prevent timeouts.
- Physically redundant links, such as Gigabit EtherChannel, must be used for the control and data paths.

Limitations

- The failover time for a box-to-box application is higher for a non-box-to-box application.
- LAN and MESH scenarios are not supported.
- VRFs are not supported and cannot be configured under ZBFW High Availability data and control interfaces.
- The maximum number of virtual MACs supported by the Front Panel Gigabit Ethernet (FPGE) interfaces depends on the platform. For information about the FPGE interfaces, see the Hardware Installation Guide for Cisco Catalyst 8300 Edge Platform.
- When the configuration is replicated to the standby device, it is not committed to the startup configuration; it is in the running configuration. A user must run the **write memory** command to commit the changes that have been synchronized from the active device, on the standby device.

Configure Interchassis High Availability

For information on configuring Interchassis High Availability on the device, see the IP Addressing: Configuring NAT for IP Address Conservation.

Bidirectional Forwarding Detection

Bidirectional Forwarding Detection (BFD) is a protocol that

- rapidly detects failures in the forwarding path between two routers,
- provides fast-forwarding path-failure detection times for all media type, routing protocol, or topology, and
- provides consistent and predictable failure detection rates for streamlined network convergence.

In addition to fast-forwarding path-failure detection, BFD provides a consistent failure detection method for network administrators. Because a network administrator can use BFD to detect forwarding path failures at a uniform rate rather than variable rates for different routing protocol hello mechanisms, network profiling and planning is easier, and reconvergence time is consistent and predictable. For more information on BFD, see the IP Routing: BFD Configuration Guide.

Bidirectional Forwarding Detection Offload

This is a short description.

The Bidirectional Forwarding Detection Offload feature allows the offload of BFD session management to the forwarding engine, resulting in faster failure detection. By sending rapid failure detection packets to routing protocols, BFD offload helps recalculating the routing table and reduces network convergence time.

Limitation

These are the limitations when configuring tha BFD offload.

- Only BFD version 1 is supported.
- When configured, only offloaded BFD sessions are supported;, BFD session on RP are not supported.
- Only Asynchronous mode or no echo mode of BFD is supported.
- 511 asynchronous BFD sessions are supported.
- BFD hardware offload is supported for IPv4 sessions with non-echo mode only.
- BFD offload is supported only on port-channel interfaces.
- BFD offload is supported only for the Ethernet interface.
- BFD offload is not supported for IPv6 BFD sessions.
- BFD offload is not supported for BFD with TE/FRR.

Configure Bidirectional Forwarding

For information on configuring BFD on your device, see the IP Routing BFD Configuration Guide.

For BFD commands, see the Cisco IOS IP Routing: Protocol-Independent Command Reference document.

Configure BFD Offload

BFD offload functionality is enabled by default. You can configure BFD hardware offload on the route processor. For more information, see the IP Routing BFD Configuration Guide.

Verify Interchassis High Availability

You can verify the Interchassis High Availability configuration using these **show** commands. This section provides some examples to verify the configuration.

Use these **show** commands to verify the Interchassis High Availability.

- **show redundancy application group [group-id | all]**
- **show redundancy application transport {client | group [group-id]}**
- **show redundancy application control-interface group [group-id]**
- **show redundancy application faults group [group-id]**
- **show redundancy application protocol {protocol-id | group [group-id]}**
- **show redundancy application if-mgr group [group-id]**
- **show redundancy application data-interface group [group-id]**

This example shows the redundancy application groups configured on the device:

```
Router# show redundancy application group
Group ID    Group Name                    State
--------    ----------                    -----
```

```
1           Generic-Redundancy-1            STANDBY
2           Generic-Redundancy2             ACTIVE
```

This example shows the details of redundancy application group 1:

```
Router# show redundancy application group 1
Group ID:1
Group Name:Generic-Redundancy-1

Administrative State: No Shutdown
Aggregate operational state : Up
My Role: STANDBY
Peer Role: ACTIVE
Peer Presence: Yes
Peer Comm: Yes
Peer Progression Started: Yes

RF Domain: btob-one
RF state: STANDBY HOT
Peer RF state: ACTIVE
```

This example shows the details of redundancy application group 2:

```
Router# show redundancy application group 2
Group ID:2
Group Name:Generic-Redundancy2

Administrative State: No Shutdown
Aggregate operational state : Up
My Role: ACTIVE
Peer Role: STANDBY
Peer Presence: Yes
Peer Comm: Yes
Peer Progression Started: Yes

RF Domain: btob-two
RF state: ACTIVE
Peer RF state: STANDBY HOT
```

This example shows details of the redundancy application transport client:

```
Router# show redundancy application transport client
Client          Conn#  Priority   Interface   L3        L4
( 0)RF            0      1          CTRL       IPV4      SCTP

( 1)MCP_HA        1      1          DATA       IPV4      UDP_REL

( 4)AR            0      1          ASYM       IPV4      UDP

( 5)CF            0      1          DATA       IPV4      SCTP
```

This example shows configuration details for the redundancy application transport group:

```
Router# show redundancy application transport group
Transport Information for RG (1)
Client = RF
TI   conn_id my_ip           my_port  peer_ip         peer_por intf    L3
  L4
0    0       192.0.2.8       59000    192.0.2.4       59000    CTRL    IPV4
  SCTP
Client = MCP_HA
```

```
TI   conn_id my_ip           my_port  peer_ip         peer_por intf    L3
  L4
1    1       10.10.2.10      53000    10.10.6.9       53000    DATA    IPV4
  UDP_REL
Client = AR
TI   conn_id my_ip           my_port  peer_ip         peer_por intf    L3
  L4
2    0       192.0.2.3       0       192.0.2.3        0        NONE_IN NONE_L3
 NONE_L4
Client = CF
TI   conn_id my_ip           my_port  peer_ip         peer_por intf    L3
  L4
3    0       10.10.2.10      59001    10.10.6.9       59001    DATA    IPV4
  SCTP
Transport Information for RG (2)
Client = RF
TI   conn_id my_ip           my_port  peer_ip         peer_por intf    L3
  L4
8    0       192.0.2.8       59004    192.0.2.2       59004    CTRL    IPV4
  SCTP
Client = MCP_HA
TI   conn_id my_ip           my_port  peer_ip         peer_por intf    L3
  L4
9    1       10.10.2.10      53002    10.10.6.9       53002    DATA    IPV4
  UDP_REL
Client = AR
TI   conn_id my_ip           my_port  peer_ip         peer_por intf    L3
  L4
10   0       192.0.2.3       0        192.0.2.3       0        NONE_IN NONE_L3
 NONE_L4
Client = CF
TI   conn_id my_ip           my_port  peer_ip         peer_por intf    L3
  L4
11   0       10.10.2.10      59005    10.10.6.9       59005    DATA    IPV4
  SCTP
```

This example shows the configuration details of redundancy application transport group 1:

```
Router# show redundancy application transport group 1
Transport Information for RG (1)
Client = RF
TI   conn_id my_ip           my_port  peer_ip         peer_por intf    L3
  L4
0    0       192.0.2.8       59000    192.0.2.4       59000    CTRL    IPV4
  SCTP
Client = MCP_HA
TI   conn_id my_ip           my_port  peer_ip         peer_por intf    L3
  L4
1    1       10.10.2.10      53000    10.10.2.10      53000    DATA    IPV4
  UDP_REL
Client = AR
TI   conn_id my_ip           my_port  peer_ip         peer_por intf    L3
  L4
2    0       192.0.2.3       0        192.0.2.3       0        NONE_IN NONE_L3
 NONE_L4
Client = CF
TI   conn_id my_ip           my_port  peer_ip         peer_por intf    L3
  L4
3    0       10.10.2.10      59001    10.10.2.10      59001    DATA    IPV4
  SCTP
```

This example shows configuration details of redundancy application transport group 2:

```
Router# show redundancy application transport group 2
Transport Information for RG (2)
Client = RF
TI   conn_id my_ip           my_port  peer_ip         peer_por intf    L3
  L4
8    0       192.0.2.8       59004    192.0.2.4       59004    CTRL    IPV4
  SCTP
Client = MCP_HA
TI   conn_id my_ip           my_port  peer_ip         peer_por intf    L3
  L4
9    1       10.10.2.10      53002    10.10.2.10      53002    DATA    IPV4
  UDP_REL
Client = AR
TI   conn_id my_ip           my_port  peer_ip         peer_por intf    L3
  L4
10   0       192.0.2.3       0        192.0.2.3       0        NONE_IN NONE_L3
 NONE_L4
Client = CF
TI   conn_id my_ip           my_port  peer_ip         peer_por intf    L3
  L4
11   0       10.10.2.10      59005    10.10.2.10      59005    DATA    IPV4
  SCTP
```

This example shows configuration details of the redundancy application control-interface group:

```
Router# show redundancy application control-interface group
The control interface for rg[1] is GigabitEthernet0/0/0
Interface is Control interface associated with the following protocols: 2 1
BFD Enabled
Interface Neighbors:
Peer: 192.0.2.4 Active RGs: 1 Standby RGs: 2 BFD handle: 0

The control interface for rg[2] is GigabitEthernet0/0/0
Interface is Control interface associated with the following protocols: 2 1
BFD Enabled
Interface Neighbors:
Peer: 192.0.2.4 Active RGs: 1 Standby RGs: 2 BFD handle: 0
```

This example shows configuration details of the redundancy application control-interface group 1:

```
Router# show redundancy application control-interface group 1
The control interface for rg[1] is GigabitEthernet0/0/0
Interface is Control interface associated with the following protocols: 2 1
BFD Enabled
Interface Neighbors:
Peer: 192.0.2.4 Active RGs: 1 Standby RGs: 2 BFD handle: 0
```

This example shows configuration details of the redundancy application control-interface group 2:

```
Router# show redundancy application control-interface group 2
The control interface for rg[2] is GigabitEthernet0/0/0
Interface is Control interface associated with the following protocols: 2 1
BFD Enabled
Interface Neighbors:
Peer: 192.0.2.4 Active RGs: 1 Standby RGs: 2 BFD handle: 0
```

This example shows configuration details of the redundancy application faults group:

```
Router# show redundancy application faults group
Faults states Group 1 info:
Runtime priority: [50]
RG Faults RG State: Up.
Total # of switchovers due to faults: 0
Total # of down/up state changes due to faults: 2
Faults states Group 2 info:
Runtime priority: [135]
RG Faults RG State: Up.
Total # of switchovers due to faults: 0
Total # of down/up state changes due to faults: 2
```

This example shows configuration details specific to redundancy application faults group 1:

```
Router# show redundancy application faults group 1
Faults states Group 1 info:
Runtime priority: [50]
RG Faults RG State: Up.
Total # of switchovers due to faults: 0
Total # of down/up state changes due to faults: 2
```

This example shows configuration details specific to redundancy application faults group 2:

```
Router# show redundancy application faults group 2
Faults states Group 2 info:
Runtime priority: [135]
RG Faults RG State: Up.
Total # of switchovers due to faults: 0
Total # of down/up state changes due to faults: 2
```

This example shows configuration details for the redundancy application protocol group:

```
Router# show redundancy application protocol group
RG Protocol RG 1
------------------
Role: Standby
Negotiation: Enabled
Priority: 50
Protocol state: Standby-hot
Ctrl Intf(s) state: Up
Active Peer: address 192.0.4.2, priority 150, intf Gi0/0/0
Standby Peer: Local
Log counters:
role change to active: 0
role change to standby: 1
disable events: rg down state 1, rg shut 0
ctrl intf events: up 2, down 1, admin_down 1
reload events: local request 0, peer request 0

RG Media Context for RG 1
--------------------------
Ctx State: Standby
Protocol ID: 1
Media type: Default
Control Interface: GigabitEthernet0/0/0
Current Hello timer: 3000
Configured Hello timer: 3000, Hold timer: 10000
Peer Hello timer: 3000, Peer Hold timer: 10000
Stats:
```

```
Pkts 117, Bytes 7254, HA Seq 0, Seq Number 117, Pkt Loss 0
Authentication not configured
Authentication Failure: 0
Reload Peer: TX 0, RX 0
Resign: TX 0, RX 0
Active Peer: Present. Hold Timer: 10000
Pkts 115, Bytes 3910, HA Seq 0, Seq Number 1453975, Pkt Loss 0



RG Protocol RG 2
------------------
Role: Active
Negotiation: Enabled
Priority: 135
Protocol state: Active
Ctrl Intf(s) state: Up
Active Peer: Local
Standby Peer: address 192.0.4.2, priority 130, intf Gi0/0/0
Log counters:
role change to active: 1
role change to standby: 1
disable events: rg down state 1, rg shut 0
ctrl intf events: up 2, down 1, admin_down 1
reload events: local request 0, peer request 0

RG Media Context for RG 2
--------------------------
Ctx State: Active
Protocol ID: 2
Media type: Default
Control Interface: GigabitEthernet0/0/0
Current Hello timer: 3000
Configured Hello timer: 3000, Hold timer: 10000
Peer Hello timer: 3000, Peer Hold timer: 10000
Stats:
Pkts 118, Bytes 7316, HA Seq 0, Seq Number 118, Pkt Loss 0
Authentication not configured
Authentication Failure: 0
Reload Peer: TX 0, RX 0
Resign: TX 0, RX 1
Standby Peer: Present. Hold Timer: 10000
Pkts 102, Bytes 3468, HA Seq 0, Seq Number 1453977, Pkt Loss 0
```

This example shows configuration details for the redundancy application protocol group 1:

```
Router# show redundancy application protocol group 1
RG Protocol RG 1
------------------
Role: Standby
Negotiation: Enabled
Priority: 50
Protocol state: Standby-hot
Ctrl Intf(s) state: Up
Active Peer: address 192.0.4.2, priority 150, intf Gi0/0/0
Standby Peer: Local
Log counters:
role change to active: 0
role change to standby: 1
disable events: rg down state 1, rg shut 0
ctrl intf events: up 2, down 1, admin_down 1
reload events: local request 0, peer request 0
```

```
RG Media Context for RG 1
--------------------------
Ctx State: Standby
Protocol ID: 1
Media type: Default
Control Interface: GigabitEthernet0/0/0
Current Hello timer: 3000
Configured Hello timer: 3000, Hold timer: 10000
Peer Hello timer: 3000, Peer Hold timer: 10000
Stats:
Pkts 120, Bytes 7440, HA Seq 0, Seq Number 120, Pkt Loss 0
Authentication not configured
Authentication Failure: 0
Reload Peer: TX 0, RX 0
Resign: TX 0, RX 0
Active Peer: Present. Hold Timer: 10000
Pkts 118, Bytes 4012, HA Seq 0, Seq Number 1453978, Pkt Loss 0
```

This example shows configuration details for the redundancy application protocol group 2:

```
Router# show redundancy application protocol group 2
RG Protocol RG 2
------------------
Role: Active
Negotiation: Enabled
Priority: 135
Protocol state: Active
Ctrl Intf(s) state: Up
Active Peer: Local
Standby Peer: address 192.0.4.2, priority 130, intf Gi0/0/0
Log counters:
role change to active: 1
role change to standby: 1
disable events: rg down state 1, rg shut 0
ctrl intf events: up 2, down 1, admin_down 1
reload events: local request 0, peer request 0

RG Media Context for RG 2
--------------------------
Ctx State: Active
Protocol ID: 2
Media type: Default
Control Interface: GigabitEthernet0/0/0
Current Hello timer: 3000
Configured Hello timer: 3000, Hold timer: 10000
Peer Hello timer: 3000, Peer Hold timer: 10000
Stats:
Pkts 123, Bytes 7626, HA Seq 0, Seq Number 123, Pkt Loss 0
Authentication not configured
Authentication Failure: 0
Reload Peer: TX 0, RX 0
Resign: TX 0, RX 1
Standby Peer: Present. Hold Timer: 10000
Pkts 107, Bytes 3638, HA Seq 0, Seq Number 1453982, Pkt Loss 0
```

This example shows configuration details for the redundancy application protocol 1:

```
Router# show redundancy application protocol 1
Protocol id: 1, name: rg-protocol-1
BFD: ENABLE
Hello timer in msecs: 3000
```

```
Hold timer in msecs: 10000
OVLD-1#show redundancy application protocol 2
Protocol id: 2, name: rg-protocol-2
BFD: ENABLE
Hello timer in msecs: 3000
Hold timer in msecs: 10000
```

This example shows configuration details for redundancy application interface manager group:

```
Router# show redundancy application if-mgr group
 RG ID: 1
 ==========

 interface      GigabitEthernet0/0/3.152
 ---------------------------------------
 VMAC           0007.b421.4e21
 VIP            203.0.113.1
 Shut           shut
 Decrement      10

 interface      GigabitEthernet0/0/2.152
 ---------------------------------------
 VMAC           0007.b421.5209
 VIP            203.0.113.4
 Shut           shut
 Decrement      10


 RG ID: 2
 ==========

 interface      GigabitEthernet0/0/3.166
 ---------------------------------------
 VMAC           0007.b422.14d6
 VIP            203.0.113.6
 Shut           no shut
 Decrement      10

 interface      GigabitEthernet0/0/2.166
 ---------------------------------------
 VMAC           0007.b422.0d06
 VIP            203.0.113.9
 Shut           no shut
 Decrement      10
```

These examples show configuration details for redundancy application interface manager group 1 and group 2:

```
Router# show redundancy application if-mgr group 1

 RG ID: 1
 ==========

 interface      GigabitEthernet0/0/3.152
 ---------------------------------------
 VMAC           0007.b421.4e21
 VIP            203.0.113.3
 Shut           shut
 Decrement      10

 interface      GigabitEthernet0/0/2.152
 ---------------------------------------
 VMAC           0007.b421.5209
```

```
 VIP            203.0.113.2
 Shut           shut
 Decrement      10

Router# show redundancy application if-mgr group 2
 RG ID: 2
 ==========

 interface      GigabitEthernet0/0/3.166
 ---------------------------------------
 VMAC           0007.b422.14d6
 VIP            203.0.113.5
 Shut           no shut
 Decrement      10

 interface      GigabitEthernet0/0/2.166
 ---------------------------------------
 VMAC           0007.b422.0d06
 VIP            203.0.113.7
 Shut           no shut
 Decrement      10
```

This example shows configuration details for redundancy application data-interface group:

```
Router# show redundancy application data-interface group
The data interface for rg[1] is GigabitEthernet0/0/1
The data interface for rg[2] is GigabitEthernet0/0/1
```

These examples show configuration details specific to redundancy application data-interface group 1 and group 2:

```
Router# show redundancy application data-interface group 1
The data interface for rg[1] is GigabitEthernet0/0/1

Router # show redundancy application data-interface group 2
The data interface for rg[2] is GigabitEthernet0/0/1
```

Verify BFD Offload

Use these commands to verify and monitor BFD offload feature on your device.

- **show bfd neighbors [details]**
- **debug bfd [packet | event]**
- **debug bfd event**

The **show bfd neighbors** command displays the BFD adjacency database:

```
Router# show bfd neighbor

IPv4 Sessions
NeighAddr                              LD/RD         RH/RS     State     Int
192.0.2.1                             362/1277       Up        Up
Gi0/0/1.2
192.0.2.5                             445/1278       Up        Up
Gi0/0/1.3
192.0.2.3                            1093/961        Up        Up
Gi0/0/1.4
192.0.2.2                            1244/946        Up        Up
Gi0/0/1.5
```

```
192.0.2.6                           1094/937        Up        Up
Gi0/0/1.6
192.0.2.7                           1097/1260       Up        Up
Gi0/0/1.7
192.0.2.4                           1098/929        Up        Up
Gi0/0/1.8
192.0.2.9                           1111/928        Up        Up
Gi0/0/1.9
192.0.2.8                           1100/1254       Up        Up
Gi0/0/1.10
```

The **debug bfd neighbor detail** command displays the debugging information related to BFD packets:

```
Router# show bfd neighbor detail

IPv4 Sessions
NeighAddr                              LD/RD         RH/RS     State     Int
192.0.2.1                             362/1277       Up        Up
Gi0/0/1.2
Session state is UP and not using echo function.
Session Host: Hardware
OurAddr: 192.0.2.2
Handle: 33
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 50000, Multiplier: 3
Received MinRxInt: 50000, Received Multiplier: 3
Holddown (hits): 0(0), Hello (hits): 50(0)
Rx Count: 3465, Rx Interval (ms) min/max/avg: 42/51/46
Tx Count: 3466, Tx Interval (ms) min/max/avg: 39/52/46
Elapsed time watermarks: 0 0 (last: 0)
Registered protocols: CEF EIGRP
Uptime: 00:02:50
Last packet: Version: 1                  - Diagnostic: 0
             State bit: Up               - Demand bit: 0
             Poll bit: 0                 - Final bit: 0
             C bit: 1
             Multiplier: 3               - Length: 24
             My Discr.: 1277             - Your Discr.: 362
             Min tx interval: 50000      - Min rx interval: 50000
             Min Echo interval: 0
```

The **show bfd summary** command displays the BFD summary:

```
Router# show bfd summary

                    Session          Up          Down

Total                   400         400             0
```

The **show bfd drops** command displays the number of packets dropped in BFD:

```
Router# show bfd drops
BFD Drop Statistics
                     IPV4     IPV6    IPV4-M    IPV6-M    MPLS_PW   MPLS_TP_LSP
Invalid TTL             0        0         0         0          0         0
BFD Not Configured      0        0         0         0          0         0
No BFD Adjacency        33       0         0         0          0         0
Invalid Header Bits     0        0         0         0          0         0
Invalid Discriminator   1        0         0         0          0         0
Session AdminDown       94       0         0         0          0         0
Authen invalid BFD ver  0        0         0         0          0         0
```

```
Authen invalid len      0         0       0         0           0       0
Authen invalid seq      0         0       0         0           0       0
Authen failed           0         0       0         0           0       0
```

The **debug bfd packet** command displays debugging information about BFD control packets.

```
Router# debug bfd packet
*Nov 12 23:08:27.982: BFD-DEBUG Packet: Rx IP:192.0.2.1 ld/rd:1941/0 diag:0(No
 Diagnostic) Down C cnt:4 ttl:254 (0)
*Nov 12 23:08:27.982: BFD-DEBUG Packet: Tx IP:192.0.2.1 ld/rd:983/1941
diag:3(Neighbor Signaled Session Down) Init  C cnt:44 (0)
*Nov 12 23:08:28.007: BFD-DEBUG Packet: Rx IP:192.0.2.1 ld/rd:1941/983 diag:0(No
 Diagnostic) Up PC cnt:4 ttl:254 (0)
*Nov 12 23:08:28.007: BFD-DEBUG Packet: Tx IP:192.0.2.1 ld/rd:983/1941 diag:0(No
 Diagnostic) Up F C cnt:0 (0)
*Nov 12 23:08:28.311: BFD-DEBUG Packet: Rx IP:192.0.2.1 ld/rd:1941/983 diag:0(No
 Diagnostic) Up FC cnt:0 ttl:254 (0)
*Nov 12 23:08:28.311: BFD-DEBUG Packet: Tx IP:192.0.2.1 ld/rd:983/1941 diag:0(No
 Diagnostic) Up  C cnt:0 (0)
*Nov 12 23:08:28.311: BFD-DEBUG Packet: Rx IP:192.0.2.3 ld/rd:1907/0 diag:0(No
 Diagnostic) Down C cnt:3 ttl:254 (0)
*Nov 12 23:08:28.311: BFD-DEBUG Packet: Tx IP:192.0.2.3 ld/rd:993/1907
diag:3(Neighbor Signaled Session Down) Init  C cnt:43 (0)
*Nov 12 23:08:28.311: BFD-DEBUG Packet: Rx IP:192.0.2.1 ld/rd:1941/983 diag:0(No
 Diagnostic) Up C cnt:0 ttl:254 (0)
*Nov 12 23:08:28.626: BFD-DEBUG Packet: Rx IP:192.0.2.3 ld/rd:1907/993 diag:0(No
 Diagnostic) Up PC cnt:3 ttl:254 (0)
*Nov 12 23:08:28.626: BFD-DEBUG Packet: Tx IP:192.0.2.3 ld/rd:993/1907 diag:0(No
 Diagnostic) Up F C cnt:0 (0)
*Nov 12 23:08:28.645: BFD-DEBUG Packet: Rx IP:192.0.2.3 ld/rd:1907/993 diag:0(No
 Diagnostic) Up C cnt:0 ttl:254 (0)
*Nov 12 23:08:28.700: BFD-DEBUG Packet: Rx IP:192.0.2.3 ld/rd:1907/993 diag:0(No
 Diagnostic) Up FC cnt:0 ttl:254 (0)
*Nov 12 23:08:28.700: BFD-DEBUG Packet: Tx IP:192.0.2.3 ld/rd:993/1907 diag:0(No
 Diagnostic) Up  C cnt:0 (0)
*Nov 12 23:08:28.993: BFD-DEBUG Packet: Rx IP:192.0.2.3 ld/rd:1907/993 diag:0(No
 Diagnostic) Up C cnt:0 ttl:254 (0)
```

The **debug bfd event** displays debugging information about BFD state transitions:

```
Router# deb bfd event

*Nov 12 23:11:29.503: BFD-DEBUG Event: notify client(EIGRP) IP:192.0.2.6,
ld:1401, handle:77, event:DOWN adminDown, (0)
*Nov 12 23:11:29.503: BFD-DEBUG Event: notify client(CEF) IP:192.0.2.6, ld:1401,
 handle:77, event:DOWN adminDown, (0)
*Nov 12 23:11:29.503: BFD-DEBUG Event: notify client(EIGRP) IP:192.0.2.10,
ld:1400, handle:39, event:DOWN adminDown, (0)
*Nov 12 23:11:29.503: BFD-DEBUG Event: notify client(CEF) IP:192.0.2.10,
ld:1400, handle:39, event:DOWN adminDown, (0)
*Nov 12 23:11:29.503: BFD-DEBUG Event: notify client(EIGRP) IP:192.0.2.8,
ld:1399, handle:25, event:DOWN adminDown, (0)
*Nov 12 23:11:29.503: BFD-DEBUG Event: notify client(CEF) IP:192.0.2.8, ld:1399,
 handle:25, event:DOWN adminDown, (0)
*Nov 12 23:11:29.503: BFD-DEBUG Event: notify client(EIGRP) IP:192.0.2.5,
ld:1403, handle:173, event:DOWN adminDown, (0)
*Nov 12 23:11:29.503: BFD-DEBUG Event: notify client(CEF) IP:192.0.2.6, ld:1403,
 handle:173, event:DOWN adminDown, (0)
*Nov 12 23:11:29.503: BFD-DEBUG Event: notify client(EIGRP) IP:192.0.2.4,
ld:1402, handle:95, event:DOWN adminDown, (0)
*Nov 12 23:11:29.503: BFD-DEBUG Event: notify client(CEF) IP:192.0.2.4, ld:1402,
```

```
 handle:95, event:DOWN adminDown, (0)
*Nov 12 23:11:30.639: BFD-HW-API: Handle 1404: Timers: Tx timer 1000000 Detect
 timer 0
*Nov 12 23:11:30.639: BFD-HW-API: Handle 1404: Flags: Poll 0 Final 0
*Nov 12 23:11:30.639: BFD-HW-API: Handle 1404: Buffer: 0x23480318 0x0000057C
0x00000000 0x000F4240 0x000F4240 0x00000000 size 24
*Nov 12 23:11:30.641: BFD-HW-API: Handle 1405: Timers: Tx timer 1000000 Detect
 timer 0
*Nov 12 23:11:30.641: BFD-HW-API: Handle 1405: Flags: Poll 0 Final 0
*Nov 12 23:11:30.641: BFD-HW-API: Handle 1405: Buffer: 0x23480318 0x0000057D
0x00000000 0x000F4240 0x000F4240 0x00000000 size 24
*Nov 12 23:11:30.649: BFD-DEBUG Packet: Rx IP:192.0.2.6 ld/rd:1601/1404
diag:7(Administratively Down) AdminDown C cnt:0 ttl:254 (0)
*Nov 12 23:11:30.650: BFD-DEBUG Event: V1 FSM ld:1404 handle:207 event:RX
ADMINDOWN state:UP (0)
*Nov 12 23:11:30.650: BFD-DEBUG Event: resetting timestamps ld:1404 handle:207
 (0)
*Nov 12 23:11:30.650: BFD-DEBUG Event: notify client(CEF) IP:192.0.2.1, ld:1404,
 handle:207, event:DOWN adminDown, (0)
*Nov 12 23:11:30.650: BFD-DEBUG Packet: Tx IP:192.0.2.1 ld/rd:1404/0
diag:3(Neighbor Signaled Session Down) Down C cnt:0 (0)
*Nov 12 23:11:30.650: BFD-DEBUG Packet: Rx IP:192.0.2.1 ld/rd:1620/1405
diag:7(Administratively Down) AdminDown C cnt:0 ttl:254 (0)
*Nov 12 23:11:30.650: BFD-DEBUG Event: V1 FSM ld:1405 handle:209 event:RX
ADMINDOWN state:UP (0)
*Nov 12 23:11:30.650: BFD-DEBUG Event: resetting timestamps ld:1405 handle:209
 (0)
*Nov 12 23:11:30.650: BFD-DEBUG Event: notify client(CEF) IP:192.0.2.1, ld:1405,
 handle:209, event:DOWN adminDown, (0)
*Nov 12 23:11:30.650: BFD-DEBUG Packet: Tx IP:192.0.2.7 ld/rd:1405/0
diag:3(Neighbor Signaled Session Down) Down C cnt:0 (0)
*Nov 12 23:11:30.650: BFD-DEBUG Event: notify client(EIGRP) IP:192.0.2.7,
ld:1404, handle:207, event:DOWN adminDown, (0)
*Nov 12 23:11:30.650: BFD-DEBUG Event: notify client(CEF) IP:192.0.2.7, ld:1404,
 handle:207, event:DOWN adminDown, (0)
*Nov 12 23:11:30.650: BFD-DEBUG Event: notify client(EIGRP) IP:192.0.2.7,
ld:1405, handle:209, event:DOWN adminDown, (0)
*Nov 12 23:11:30.650: BFD-DEBUG Event: notify client(CEF) IP:192.0.2.7, ld:1405,
 handle:209, event:DOWN adminDown, (0)
*Nov 12 23:11:31.035: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 192.0.2.8
```

6 Management Ethernet Interface

Topics:

- Management Ethernet Interface

Describes the dedicated interface used for out-of-band device management and administrative tasks. It explains the unique virtual routing and forwarding instance assigned to this port and details common configuration tasks like setting IP addresses and default routes.

Management Ethernet Interface

Describes the dedicated interface used for out-of-band device management and administrative tasks. It explains the unique virtual routing and forwarding instance assigned to this port and details common configuration tasks like setting IP addresses and default routes.

The Cisco 8500 Series Catalyst Edge Platform have one Gigabit Ethernet Management Ethernet interface.

Gigabit Ethernet management interface

A management interface is a router interface that

- allows users to perform management tasks on the router,
- typically does not, and often cannot, forward network traffic,
- provides access to the router, often via Telnet and SSH for most management operations, and
- is especially useful before the router has begun routing or when the SPA interfaces are inactive during troubleshooting scenarios.

These aspects of the Management Ethernet interface should be noted:

- IPv4, IPv6, and ARP are the only routed protocols supported for the interface.
- The Ethernet Management Interface cannot be used as a Lawful Intercept MD source interface.
- The Management Ethernet interface is part of its own VRF. This is discussed in more detail in the Gigabit Ethernet Management Interface VRF.

IP Address in ROMmon and the management Ethernet port

On the Cisco 8500 Series Catalyst Edge Platform, IP addresses can be configured in ROMmon (the **IP_ADDRESS= and IP_SUBNET_MASK=** commands) and through the use of the IOS command-line interface (the **ip address** command in interface configuration mode).

Assuming the IOS process has not begun running on the Cisco 8500 Series Catalyst Edge Platform, the IP address that was set in ROMmon acts as the IP address of the Management Ethernet interface.

In cases where the IOS process is running and has taken control of the Management Ethernet interface, the IP address specified when configuring the Gigabit Ethernet 0 interface in the IOS CLI becomes the IP address of the Management Ethernet interface. The ROMmon-defined IP address is only used as the interface address when the IOS process is inactive.

For this reason, the IP addresses specified in ROMmon and in the IOS CLI can be identical and the Management Ethernet interface will function properly in single RP configurations.

GigabitEthernet Management interface VRF

The GigabitEthernet Management interface is automatically part of its own VRF. This VRF, which is named “Mgmt-intf,” is automatically configured on the Cisco 8500 Series Catalyst Edge Platform and is dedicated to the Management Ethernet interface; no other interfaces can join this VRF. Therefore, this VRF does not participate in the MPLS VPN VRF or any other network-wide VRF. The Mgmt-intf VRF supports loopback interface.

Placing the management Ethernet interface in its own VRF results in these effects on the Management Ethernet interface:

- Many features must be configured or used inside the VRF, therefore, the CLI may differ for certain Management Ethernet functions on the Cisco 8500 Series Catalyst Edge Platform compared to the Management Ethernet interfaces on other routers.

- This configuration prevents transit traffic from traversing the router. Because all built-in ports and the Management Ethernet interface are automatically assigned to different VRFs, no transit traffic can enter the Management Ethernet interface and leave a built-in port, nor travel in the opposite direction.
- The interface security is improved. Because the Mgmt-intf VRF has its own routing table, only routes that are explicitly entered by a user can be added to the routing table of the Management Ethernet interface.

The Management Ethernet interface VRF supports both the IPv4 and IPv6 address families.

Common Ethernet management tasks

You can complete most router tasks by accessing the router through the Management Ethernet interface.

This section covers tasks that might be common or somewhat tricky on the Cisco 8500 Series Catalyst Edge Platform. It is not intended as a comprehensive list of all tasks that can be done using the Management Ethernet interface.

This section describes common processes.

View the VRF configuration

The VRF configuration for the Management Ethernet interface is viewable using the **show running-config vrf** command.

This example shows the default VRF configuration:

```
Router#show running-config vrf
Building configuration...
Current configuration : 351 bytes
vrf definition Mgmt-intf
!
address-family ipv4
exit-address-family
!
address-family ipv6
exit-address-family
!

(some output removed for brevity)
```

View detailed VRF information for the management ethernet VRF

To see detailed information about the Management Ethernet VRF, enter the **show vrf detail Mgmt-intf** command:

```
Router#show vrf detail Mgmt-intf
```

Set a default route in the management Ethernet interface VRF

To set a default route in the Management Ethernet Interface VRF, enter this command

ip route vrf Mgmt-intf 0.0.0.0 0.0.0.0 *next-hop-IP-address*

Set the management Ethernet IP address

The IP address of the Management Ethernet port is set like the IP address on any other interface.

Below are two simple examples of configuring an IPv4 adress and an IPv6 address on the Management Ethernet interface.

IPv4 Example

```
Router(config)# interface GigabitEthernet 0
Router(config-if)# ip address
A.B.C.D A.B.C.D
```

IPv6 Example

```
Router(config)# interface GigabitEthernet 0
Router(config-if)# ipv6 address X:X:X:X::X
```

Copy using TFTP or FTP

To copy a file using TFTP through the Management Ethernet interface, the **ip tftp source-interface GigabitEthernet 0** command must be entered before entering the **copy tftp** command because the **copy tftp** command has no option of specifying a VRF name.

Similarly, to copy a file using FTP through the Management Ethernet interface, the **ip ftp source-interface GigabitEthernet 0** command must be entered before entering the **copy ftp** command because the **copy ftp** command has no option of specifying a VRF name.

TFTP example

```
Router(config)# ip tftp source-interface gigabitethernet 0
```

FTP example

```
Router(config)# ip ftp source-interface gigabitethernet 0
```

NTP server

To allow the software clock to be synchronized by a Network Time Protocol (NTP) time server over the Management Ethernet interface, enter **the ntp server vrf Mgmt-intf** command and specify the IP address of the device providing the update.

This CLI provides an example of this procedure.

```
Router(config)# ntp server vrf Mgmt-intf 172.17.1.1
```

SYSLOG server

To specify the Management Ethernet interface as the source IP or IPv6 address for logging purposes, enter the **logging host <ip-address> vrf Mgmt-intf** command.

This CLI provides an example of this procedure.

```
Router(config)# logging host <ip-address> vrf Mgmt-intf
```

SNMP-Related services

To specify the Management Ethernet interface as the source of all SNMP trap messages, enter the **snmp-server source-interface traps gigabitEthernet 0** command.

This CLI provides an example of this procedure:

```
Router(config)# snmp-server source-interface traps gigabitEthernet 0
```

Domain name assignment

The IP domain name assignment for the Management Ethernet interface is done through the VRF.

To define the default domain name as the Management Ethernet VRF interface, enter the **ip domain-name vrf Mgmt-intf domain** command.

```
Router(config)# ip domain-name vrf Mgmt-intf cisco.com
```

DNS service

To specify the Management Ethernet interface VRF as a name server, enter the **ip name-server vrf Mgmt-intf** *IPv4-or-IPv6-address* command.

```
Router(config)#ip name-server vrf Mgmt-intf
IPv4-or-IPv6-address
```

RADIUS or TACACS+ server

To group the Management VRF as part of a AAA server group, enter the **ip vrf forward Mgmt-intf** command when configuring the AAA server group.

The same concept is true for configuring a TACACS+ server group. To group the Management VRF as part of a TACACS+ server group, enter the **ip vrf forwarding Mgmt-intf** command when configuring the TACACS+ server group.

RADIUS server group configuration

```
Router(config)#aaa group server radius hello
Router(config-sg-radius)#ip vrf forwarding Mgmt-intf
```

TACACS+ server group example

```
Router(config)#aaa group server tacacs+ hello
Router(config-sg-tacacs+)#ip vrf forwarding Mgmt-intf
```

VTY lines with ACL

To ensure an access control list (ACL) is attached to vty lines that are and are not using VRF, use the vrf-also option when attaching the ACL to the vty lines.

```
Router(config)# line vty 0 4
Router(config-line)# access-class 90 in vrf-also
```

7 Bridge Domain Interfaces

Topics:

- Information about bridge domain interface
- Configure bridge domain virtual IP interface

Provides instructions for configuring logical interfaces that connect Layer 2 bridged networks with Layer 3 routed environments. It covers Ethernet virtual circuits, MAC address assignment, and the encapsulation methods required for efficient packet forwarding.

Information about bridge domain interface

Introduces bridge domain interface architecture, detailing restrictions, Ethernet virtual circuits, encapsulation, MAC address assignment, IP protocol support, forwarding mechanisms between Layer 2 and Layer 3, link states, interface statistics, and providing step-by-step configuration and verification instructions including virtual IP interface overview.

A bridge domain interface is a logical interface that

- allows bidirectional flow of traffic between a Layer 2 bridged network and a Layer 3 routed network,
- is identified by the same index as its associated bridge domain,
- represents a Layer 2 broadcast domain through each bridge domain, and
- can be associated only once per bridge domain, allowing just one bridge domain interface per domain.

Bridge domain interfaces support these features:

- IPv4 Multicast
- QoS marking and policing. Shaping and queuing are not supported
- IPv4 VRF
- IPv6 unicast forwarding
- Dynamic routing such as BGP, OSPF, EIGRP, RIP, IS-IS, and STATIC
- Hot Standby Router Protocol (HSRP)
- Virtual Router Redundancy Protocol (VRRP) from IOS XE 3.8.0 onwards.
- IP termination
- Layer 3 VPN termination
- Address Resolution Protocol (ARP), G-ARP, and P-ARP handling
- MAC address assignment

Restrictions for bridge domain interfaces

These are the restrictions pertaining to bridge domain interfaces.

These are restrictions for bridge domain interfaces

- Only 4096 bridge domain interfaces are supported per system.
- For a bridge domain interface, the maximum transmission unit (MTU) size can be configured between 1500 and 9216 bytes.

Unsupported Features

Bridge domain interfaces do not support these features:

- PPP over Ethernet (PPPoE)
- Bidirectional Forwarding Detection (BFD) protocol
- QoS
- Network-Based Application Recognition (NBAR) or Advanced Video Coding (AVC)

Ethernet virtual circuit overview

An Ethernet Virtual Circuit (EVC) is an end-to-end representation of a single instance of a Layer 2 service that

- is offered by a provider and defines the service parameters,
- consists of one or more Layer 2 interfaces called service instances within the Cisco EVC Framework,
- instantiates a service instance on a specific router port, and
- associates each service instance with a bridge domain based on configuration.

An incoming frame can be classified as a service instance based on these criteria:

- Single 802.1Q VLAN tag, priority-tagged, or 802.1ad VLAN tag
- Both QinQ (inner and outer) VLAN tags, or both 802.1ad S-VLAN and C-VLAN tags
- Outer 802.1p CoS bits, inner 802.1p CoS bits, or both
- Payload Ethernet type (IPv4, IPv6, PPPoE-all, PPPoE-discovery, and PPPoE-session)

Service instance also supports alternative mapping criteria:

- Untagged–Maps to all frames lacking an 802.1Q or 802.1ad header
- Default–Maps to all frames

Bridge domain interface encapsulation

Security Group classification includes both Source and Destination Group, which is specified by source SGT and DGT. SGT Based PBR feature provides the PBR route-map match clause for SGT/DGT based packet classification. SGT Based PBR feature supports configuration of an unlimited number of tags. Configure the tags according to the memory available on the platform.

An EVC provides the ability to employ different encapsulations on each Ethernet flow point (EFP) present in a bridge domain. A BDI egress point may not be aware of the encapsulation of an egress packet. The packet may have exited from one or more EFPs with different encapsulations.

Encapsulation in a bridge domain is a configuration method that dictates how traffic is tagged and untagged within the domain, depending on the encapsulation settings of its Ethernet Flow Points (EFPs).

- If all EFPs have different encapsulations, the Bridge Domain Interface (BDI) must be untagged (using the no 802.1Q tag), and you must configure traffic encapsulation (popped or pushed) and rewriting at each EFP.
- If all EFPs have the same encapsulation, configure the encapsulations directly on the BDI using the encapsulation command.
- Enabling encapsulation at the BDI when EFPs have the same encapsulation ensures effective pushing or popping of tags. This eliminates the need to configure the rewrite command at the EFPs.

Assign a MAC address

Important information about assigning a MAC Address includes:

- All bridge domain interfaces on Cisco Catalyst 8500 Series Edge Platforms share one MAC address. The first interface within a bridge domain receives a MAC address, and each additional bridge domain interface in the same domain uses that address.
- You can configure a static MAC address on a bridge domain interface using the **mac-address** command.

Support for IP protocols

Bridge domain interfaces allow the Cisco 8500 Series Catalyst Edge Platform to function as a Layer 3 endpoint on a Layer 2 bridge domain for these IP-related protocols:

- ARP
- DHCP
- HTTP
- ICMP
- NTP
- RARP
- SNMP
- TCP
- Telnet
- TFTP
- UDP

IP forward support

Bridge domain interface supports these IP forwarding features:

- IPv4 input and output access control lists (ACL)
- IPv4 input and output QoS policies. The operations supported for the input and output service policies on a bridge domain interface are:
  - Classification
  - Marking
  - Policing
- IPv4 L3 VRFs

Packet forwarding

A bridge domain interface provides bridging and forwarding services between the Layer 2 and Layer 3 network infrastructure.

Layer 2 to Layer 3

During a packet flow from a Layer 2 network to a Layer 3 network, the packet or a copy of the packet is forwarded to the bridge domain interface in these cases.

- If the destination MAC address matches the bridge domain interface MAC address, the packet is forwarded to the bridge domain interface.
- If the destination MAC address is a multicast address, a copy of the packet is forwarded to the bridge domain interface.

> **Note**
>
> MAC address learning cannot not be performed on the bridge domain interface.

Layer 3 to Layer 2

When a packet arrives at a Layer 3 physical interface of a router, a route lookup action is performed. If the route lookup points to a bridge domain interface, the bridge domain interface adds layer 2 encapsulation and forwards the frame to the correct bridge domain.

During a Layer 2 lookup on a bridge domain to which the bridge domain interface belongs, the bridge domain forwards the packets to the correct service instance based on the destination MAC address.

Link states of a bridge domain and a bridge domain interface

Bridge domain interface acts as a routable IOS interface on Layer 3 and as a port on a bridge domain. Both bridge domain interfaces and bridge domains operate with individual administrative states.

Shutting down a bridge domain interface stops the Layer 3 data service, but does not override or impact the state of the associated bridge domain.

Shutting down a bridge domain has several impacts on its associated components

- It stops Layer 2 forwarding across all associated members, including service instances and bridge domain interfaces.
- The operational state of a bridge domain is influenced by its associated service instances.
- A bridge domain interface can only become operational if at least one of its associated service instances is active.

Because a bridge domain interface is an internal interface, the operational state of bridge domain interface does not affect the bridge domain operational state.

Bridge domain interface-initial state

The initial administrative state of a BDI depends on how the BDI is created. When you create a BDI at boot time in the startup configuration, the default administrative state for the BDI is up. It will remain in this state unless the startup configuration includes the shutdown command.

This behavior is consistent with all the other interfaces. When you create a BDI dynamically at command prompt, the default administrative state is down.

Bridge domain interface-link state

A BDI maintains a link state that comprises of three states: administratively down, operationally down, and up. The link state of a BDI is derived from two independent inputs: the BDI administrative state set by the corresponding users and the fault indication state from the lower levels of the interface states. It defines a BDI link state based on the state of the two inputs.

Fault Indication State	BDI Admin State	Resulting BDI Link State
No faults asserted	Shutdown	Admin-down
No faults asserted	No Shutdown	Up
At least one fault	Shutdown	Admin-down
At least one fault	No Shutdown	Operationally Down

Bridge domain interface statistics

For virtual interfaces, such as the bridge domain interface, protocol counters are periodically queried from the QFP.

When packets flow from a Layer 2 bridge domain network to a Layer 3 routing network through the bridge domain interface, the packets are treated as bridge domain interface input packets and bytes.

When packets arrive at a Layer 3 interface and are forwarded through the bridge domain interface to a Layer 2 bridge domain, the packets are treated as output packets and bytes, and the counters are updated accordingly.

A BDI maintains a standard set of Layer 3 packet counters, as is the case with all Cisco IOS interfaces.

The convention of the counters is relative to the Layer 3 cloud. For example, input refers to the traffic entry to the Layer 3 cloud from the Layer 2 BD, while output refers to the traffic exit from the Layer 3 cloud to the Layer 2 BD.

Use the **show interfaces accounting** command to display the statistics for the BDI status. Use the **show interface** *<if-name>* command to display the overall count of the packets and bytes that are transmitted and received.

Create or delete a bridge domain interface

When you define an interface or subinterface for a Cisco IOS router, you name it and specify how it is assigned an IP address. You can create a bridge domain interface before adding a bridge domain to the system. The new bridge domain interface is activated after the associated bridge domain is configured.

When you create the bridge domain interface and the bridge domain, the system maintains the required associations for mapping the bridge domain-bridge domain interface pair.

The system maintains the mapping between bridge domains and bridge domain interfaces. The bridge domain interface uses the index of the associated bridge domain to show the association.

> **Note**
>
> When a bridge domain interface is created, a bridge domain is automatically created.

Bridge domain virtual IP interface

The Virtual IP Interface (VIF) feature helps to associate multiple BDI interfaces with a BD instance. The BD-VIF interface inherits all the existing L3 features of IOS logical IP interface.

The Virtual IP Interface (VIF) feature has these limitations:

- BD-VIF interface does not support IP multicast.
- The number of BD-VIF interfaces with automatically generated MAC addresses varies depending on the platform.
- BD-VIF Interface does not support MPLS.
- The maximum number of BD-VIF interfaces per bridge-domain and the total number of BD-VIF interface for per system vary based on the type of platforms.

> **Note**
>
> You must configure every BD-VIF interface with a unique MAC address and it should belong to a different VRF.

The maximum number of BD-VIF supported on Cisco Catalyst 8500 Series Edge Platforms are:

- C8500-12X4QC supports a maximum of 100 BD-VIF interfaces for a bridge domain.
- C8500-12X supports a maximum of 16 BD-VIFs for a bridge domain.

> **Note**
>
> From Cisco IOS XE 17.7 release, BD-VIF supports Flexible Netflow (FNF).

How to configure a bridge domain interface

To configure a bridge domain interface, perform the these steps:

Procedure

1. **enable**

   Example

   ```
   Router> enable
   ```

   Enables privileged EXEC mode. Enter your password if prompted.

2. **configureterminal**

   Example

   ```
   Router# configure terminal
   ```

   Enters global configuration mode.

3. **interface BDI***{interfacenumber}*

   Example

   ```
   Router(config-if)# interface BDI3
   ```

   Specifies a bridge domain interface on a Cisco 8500 Series Catalyst Edge Platform.

4. **encapsulation***encapsulationdot1q<first-tag>[second-dot1q<second-tag>]*

   Example

   ```
   Router(config-if)# encapsulation dot1Q 1 second-dot1q 2
   ```

   Defines the encapsulation type.

   The example shows how to define dot1q as the encapsulation type.

5. Do one of these steps:

Example

```
ipaddress ip-addressmask
```

Example

```
ipv6address {X:X:X:X::X link-local | X:X:X:X::X/prefix [anycast | eui-64] | autoconfig [default]}
```

Example

```
Router(config-if)# ip address 2.2.2.1 255.255.255.0
```

Example

```
Router(config-if)# ipv6 address AB01:CD1:123:C::/64 eui-64
```

Specifies either the IPv4 or IPv6 address for the bridge domain interface.

6. **match security-group destination tag** *sgt-number*

Example

```
Router(config-route-map)# match security-group destination tag 150
```

Configures the value for security-group destination security tag.

7. **mac address** *{mac-address}*

Example

```
Router(config-if)# mac-address 1.1.3
```

Specifies the MAC address for the bridge domain interface.

8. **no shut**

Example

```
Router(config-if)# no shut
```

Enables the bridge domain interface on the Cisco 8500 Series Catalyst Edge Platform.

9. **shut**

Example

```
Router(config-if)# shut
```

Disables the bridge domain interface on the Cisco 8500 Series Catalyst Edge Platform.

Bridge domain interface is configured.

Display and verify bridge domain interface configuration

To display and verify bridge domain interface configuration, follow these steps.

Procedure

1. **enable**

   Example

   ```
   Router> enable
   ```

   Enables privileged EXEC mode. Enter your password if prompted.

2. **show interfaces bdi**

   Example

   ```
   Router#showinterfacesBDI3
   ```

   Displays the configuration summary of the corresponding BDI.

3. **show platform software interface fp active name**

   Example

   ```
   Router# showplatformsoftwareinterfacefpactivenameBDI4
   ```

   Displays the bridge domain interface configuration in a Forwarding Processor.

4. **show platform hardware qfp active interface if-name**

   Example

   ```
   Router# showplatformhardwareqfpactiveinterfaceif-nameBDI4
   ```

   Displays the bridge domain interface configuration in a data path.

5. **debug platform hardware qfp feature**

   Example

   ```
   Router# debugplatformhardwareqfpactivefeaturel2bdclientall
   ```

   The selected CPP L2BD Client debugging is on.

6. **platform trace runtime process forwarding-manager module**

Example

```
Router(config)#platformtraceruntimeslotF0bay0processforwarding-managermoduleinterfaceslevelinfo
```

Enables the Forwarding Manager Route Processor and Embedded Service Processor trace messages for the Forwarding Manager process.

7. **platform trace boottimeprocess forwarding-manager module interfaces**

Example

```
Router(config)#platformtraceboottimeslotR0bay1processforwarding-managerforwarding-managerlevelmax
```

Enables the Forwarding Manager Route Processor and Embedded Service Processor trace messages for the Route Processor Forwarding Manager process during bootup.

Bridge domain interface configuration is verified.

What to do next

For additional information on the commands and the options available with each command, see the Cisco IOS Configuration Fundamentals Command Reference Guide located at:

{start hypertext}http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html{end hypertext}

Configure bridge domain virtual IP interface

Explains virtual IP interface configuration, including associating VIF interfaces with a bridge domain, verifying interface settings, and integrating example configurations for deploying virtual IP interfaces within bridge domain environments.

To configure bridge-domain virtual IP interface.

```
enable
configure terminal
[no] interface BD-VIF interface-number
  [[no] vrf forwarding vrf-name]
  [[no] mac address mac-address]
  [[no] ip address ip-address mask]
  [[no] ipv6 address {X:X:X:X::X link-local| X:X:X:X::X/prefix [anycast |
eui-64] | autoconfig [default]}]

exit
```

> **Note**
>
> To delete BD-VIF interface, use the 'no' form of the command.

Associate VIF interface with a bridge domain

To associate VIF interface with a bridge domain.

```
enable
configure terminal
bridge-domain bridge-domain number
[no] member BD-VIF interface-number
exit
```

Verify bridge domain virtual IP interface

To verify bridge-domain virtual IP interface.

All existing show commands for interface and IP interface can be used for the BD-VIF interface.

- show interface bd-vif bd-vif-id
- show ip interface bd-vif bd-vif-id
- show bd-vif interfaces in fman-fp
- show pla sof inter fp ac brief | i BD_VIF

Example configuration bridge domain virtual IP interface

Example configuration for Bridge-Domain virtual IP interface.

```
Detail sample:
interface Port-channel1
mtu 9000
no ip address
 !Ethernet service endpoint one per neutron network
service instance 1756 ethernet
  description 4e8e5957-649f-477b-9e5b-f1f75b21c03c
  encapsulation dot1q 1756
  rewrite ingress tag pop 1 symmetric
  bridge-domain 1756
!
interface BD-VIF5001
no shutdown
vrf forwarding vrf5001
ip address 10.0.0.1 255.255.255.0
interface BD-VIF5002
no shutdown
vrf forwarding vrf5002
ip address 10.0.0.2 255.255.255.0
bridge-domain 1756
member Port-channel1 service-instance 1756
member bd-vif5001
member bd-vif5002
```

8 Diagnostic Utilities

Topics:

- Packet drops
- Packet trace

Details tools for monitoring and troubleshooting traffic flow, specifically focusing on packet drop analysis and path tracing. It explains how to set warning thresholds and use detailed trace levels to identify the root cause of network performance issues.

Packet drops

Explains how to identify and troubleshoot lost data within the network processor. Use these diagnostic commands and warning thresholds to monitor resource limitations and improve overall application performance.

Packet drops

Packet drops are a network event that

- occur when network devices cannot forward incoming packets due to resource limitations,
- result from insufficient buffer space, CPU overload, or network congestion, and
- impact application performance by causing delays or disconnects.

During a period of high CPU utilization on a Cisco ASR 1000 router, packets destined for a specific interface are dropped once the buffer exceeds its capacity, resulting in application disconnects.

Packet drops occur when a router or network device fails to forward packets to their intended destination, resulting in lost data within the network. Packet drops can significantly impact network performance and application responsiveness, especially for real-time or UDP-based applications that do not have retransmission mechanisms.

Packet drop diagnostics

This section outlines information on packet drop diagnostics.

From Cisco IOS XE 17.6, you can run the show drops command to troubleshoot the root cause of packet drops. Using the **show drops** command, you can identify the following items:

- The root cause of a packet drop based on the feature or the protocol.
- The history of the Quantum Flow Processor drops.

View packet drop information

Packet drops can indicate areas where traffic is not processed as expected. You can use these commands to identify drop types, as well as affected interfaces, features, or protocols.

Procedure

1. **enable**

   Example

   ```
   Router> enable
   ```

   Enable privileged EXEC mode. Enter your password if prompted.

2. **show drops**

   Example

   ```
   Router# show drops
   ```

   Displays the drop statistics.

3. **show drops** { **bqs** | **crypto** | **firewall** | **interface** | **ip-all** | **nat** | **punt** | **qfp** | **qos** | **history** }

Example

```
Router# show drops qfp
```

Displays the drop statistics and provides a summary for the chosen interface or protocol.

> **Note**
>
> From Cisco IOS XE 17.13.1a, a new keyword option **history** is added to the **show drops** command. You can use the **show drops history qfp** command to view the history of QFP drops.

You see lists and summaries of packet drop statistics that help you identify where drops are occurring. This information supports troubleshooting and configuration adjustments.

The following is a sample output of the show drops command. This sample output displays the **packet drops** information related to the Quantum Flow Processor (QFP).

```
Router# show drops
bqs BQS related drops
crypto IPSEC related drops
firewall Firewall related drops
history History of drops
interface Interface drop statistics
ip-all IP related drops
nat NAT related drops
punt Punt path related drops
qfp QFP drop statistics
qos QoS related drops
| Output modifiers
<cr> <cr>
```

```
Router# show drops qfp
show platform hardware qfp active statistics drop detail
Last clearing of QFP drops statistics : Fri Feb 18 08:02:37 2022
(6d 23h 54m 29s ago)
------------------------------------------------------------------
ID Global Drop Stats Packets
Octets
------------------------------------------------------------------
319 BFDoffload 9
1350
61 Icmp 84
3780
53 IpFragErr 32136
48718168
244 IpLispHashLkupFailed 3
213
56 IpsecInput 18
4654
23 TailDrop 26713208
10952799454
216 UnconfiguredIpv6Fia 241788
26596680
show platform hardware qfp active interface all
statistics drop_summary
```

```
------------------------------------------------------------------
Drop Stats Summary:
note:
1) these drop stats are only updated when PAL
reads the interface stats.
2) the interface stats include the subinterface
Interface Rx Pkts Tx Pkts
------------------------------------------------------------------
GigabitEthernet1 60547 0
GigabitEthernet2 60782 27769658
GigabitEthernet3 60581 0
GigabitEthernet4 60502 1323990
Tunnel14095001 0 1990214
Tunnel14095002 0 3883238
Tunnel14095003 0 3879243
Tunnel14095004 0 2018866
Tunnel14095005 0 3875972
Tunnel14095006 0 3991497
Tunnel14095007 0 4107743
Tunnel14095008 0 3990601
```

Commands for viewing packet drop information

This section provides examples of command output used to verify packet information. To view drop statistics for all interfaces in Packet Processor Engine, use the **show drops qfp** command.

> **Note**
>
> The wrapper command **show drops qfp** is the shorthand notation for the original **show platform hardware qfp active statistics drop** command.

```
Router# show drops qfp
----------------------------------------
Global Drop Stats Octets
Packets
----------------------------------------
AttnInvalidSpid 0 0
BadDistFifo 0 0
BadIpChecksum 0 0
```

To view the history of QFP drops for all interfaces in Packet Processor Engine, use the s**how drops history qfp** command. This command can also track the number of packet drops in the last 1 minute, 5 minute, and 30 minute time period.

> **Note**
>
> The wrapper command **show drops history qfp** is the shorthand notation for the original **show platform hardware qfp active statistics drop history** command.

> **Note**
>
> The wrapper command **show drops history qfp** is not available on Catalyst 8500L Edge Platform.

```
Router# show drops history qfp
Last clearing of QFP drops statistics : Mon Jun 26 07:29:14
2023
(21s ago)
---------------------------------------------------------------
Global Drop Stats 1-Min
5-Min 30-Min All
---------------------------------------------------------------
Ipv4NoAdj 0
0 0 99818
Ipv4NoRoute 0
0 0 99853
```

Packet drop warning thresholds

This topic describes the packet drops warnings.

From Cisco IOS XE 17.14, you can configure the warning thresholds for packet drops, either for each drop cause or for the total QFP drop rate, using packets per second as the threshold value. If the configured thresholds are exceeded, then a rate-limited syslog warning is generated. One warning is generated if the total threshold is exceeded, and an additional warning is generated for each drop cause that exceeds its threshold.

The warning is generated a maximum of once per minute for each drop cause. The drops over the previous minute are checked against the threshold (packets per second) x 60, and if the drops exceed this value, a warning is generated.

These are sample warnings for total and individual drop causes.

```
%QFP-5-DROP_OVERALL_RATE: Exceeded the overall drop threshold 10000 pps during
 the last 60-second measurement period, packets dropped in last 1 minute:
641220, last 5 minutes: 1243420, last 30 minutes: 124342200
```

```
%QFP-5-DROP_CAUSE_RATE: Exceeded the drop threshold 1000 pps for QosPolicing
(drop code: 20) during the last 60-second measurement period, packets dropped
 due to QosPolicing in last 1 minute: 61220, last 5 minutes: 43420, last 30
minutes: 4611200
```

Configure packet drop warning threshold

This topic describes how to configure packet drop warning threshold.

Use this procedure when you want to receive alerts for excessive packet drops in the platform's Quantum Flow Processor.

Procedure

1. **enable**

   Example

   ```
   Router> enable
   ```

   Enables the privileged EXEC mode. Enter your password if prompted.

2. **configure terminal**

   Example

   ```
   Router# configure terminal
   ```

   Enters global configuration mode.

3. **platform qfp drops threshold** { **per-cause** *drop_id threshold* | **total** *threshold* }

   Example

   ```
   Router# platform qfp drops threshold per-cause 206 10
   ```

   Specifies the threshold value for each drop cause or the total.

   Note

   Use the **show platform hardware qfp active statistics drop detail** command to view the drop cause ID.

You receive a warning when packet drop counts exceed the configured thresholds.

This example shows how to configure the warning threshold of 15 pps for drop cause ID 24.

```
Router> enable
Router# configure terminal
Router(config)# platform qfp drops threshold
per-cause Set warning threshold for per cause QFP drops
total Set warning threshold for total QFP drops
Router(config)#platform qfp drops threshold per-cause ?
<0-1024> QFP drop cause ID
Router(config)# platform qfp drops threshold per-cause 24 ?
<0-2147483647> Drop threshold in packets per second (pps)
Router(config)# platform qfp drops threshold per-cause 24 15
```

This example shows how to configure the warning threshold of 100 pps for total QFP drops.

```
Router> enable
Router# configure terminal
Router(config)# platform qfp drops threshold
per-cause Set warning threshold for per cause QFP drops
total Set warning threshold for total QFP drops
```

```
Router(config)# platform qfp drops threshold total ?
<0-2147483647> Drop threshold in packets per second (pps)
Router(config)# platform qfp drops threshold total 100
```

View packet drop warning thresholds

This topic describes how to view packet drop warning thresholds.

Identify and display the configured warning thresholds for packet drops by cause and for total QFP drops.

Procedure

1. **enable**

   Example

   ```
   Router> enable
   ```

   Enables the privileged EXEC mode. Enter your password if prompted.

2. **show platform hardware qfp active statistics drop threshold**

   Example

   ```
   Router# show platform hardware qfp active statistics drop thresholds
   ```

   Displays the configured warning thresholds for each drop cause and for total QFP drops.

   Note

   - The wrapper command **show drops thresholds** is the shorthand notation of the **show platform hardware qfp active statistics drop threshold** command.
   - The wrapper command **show drops thresholds** is currently not available on Catalyst 8500L Edge Platform.

You can view the warning threshold values for each drop cause and the total. These values help you monitor and troubleshoot packet drop events.

This is a sample output of the **show platform hardware qfp active statistics drop threshold** command.

```
Router# show platform hardware qfp active statistics drop thresholds
-------------------------------------------------------
Drop ID         Drop Cause Name             Threshold
-------------------------------------------------------
10             BadIpChecksum                 100
206            PuntPerCausePolicerDrops      10
20             QosPolicing                   200
               Total                         30
```

This is a sample output of the **show drops thresholds** wrapper command.

```
Router# show platform hardware qfp active statistics drop thresholds
-------------------------------------------------------
Drop ID         Drop Cause Name             Threshold
-------------------------------------------------------
10             BadIpChecksum                100
206            PuntPerCausePolicerDrops     10
20             QosPolicing                  200
               Total                        30
```

Packet trace

Explains how to inspect detailed packet processing data based on specific debug conditions. Use this feature to verify policy enforcement and analyze forwarding behaviors without significantly impacting system performance.

Packet trace

A packet trace is a diagnostic feature that

- provides detailed insight into how data packets are processed within Cisco IOS XE platforms,
- enables selective debugging and inspection of traffic flow based on user-defined conditions, and
- has inspection detail levels that can impact device performance depending on the amount of information collected.

You can use packet trace to troubleshoot network issues, verify policy enforcement, and analyze unusual forwarding behaviors directly on the device.

Packet trace feature levels

The packet trace feature provides three levels of inspection for packets: accounting, summary, and path data. Each level provides a detailed view of packet processing at the cost of some packet processing capability. However, this feature inspects only packets that match the debug platform condition statements. This approach remains viable even during periods of heavy network traffic in customer environments.

Table 13: Packet trace level

Packet trace level	Description
Accounting	Provides a count of packets entering and leaving the network processor. This activity runs continuously in a lightweight manner until disabled.
Summary	Collects data for a finite number of packets, including input and output interfaces, final packet state, and the handling of punt, drop, or inject packets. This mode increases processing activity compared to normal operation.

Packet trace level	Description
Path data	

Packet trace level	Description
	Data is collected for a finite number of packets. Packet trace path data captures data, including a conditional debugging ID that is useful to correlate with feature debugs, a timestamp, and also feature-specific path-trace data. Path data also has two optional capabilities: packet copy and Feature Invocation Array (FIA) trace. The packet copy option enables you to copy input and output packets at various layers of

Packet trace level	Description
	the packet (layer 2, layer 3 or layer 4). This level uses the most processing resources and is suitable only when performance impact is acceptable.

Packet trace level	Description
	Note Collecting path data consumes more packet processing resources, and the optional capabilities incrementally affect packet performance. Therefore, path data level should be used in limited capacity or in situations where packet performance change is acceptable.

Best practice for packet trace configuration

Use these best practices when you configure the packet trace feature:

- Use ingress conditions when configuring the packet trace feature to gain a comprehensive view of packets.
- If the system has limited data-plane memory, select packet trace values carefully to minimize memory usage.
- Estimate how much memory you need before you enable packet trace. Limit the number of packets you trace and the amount of data you collect as needed.

You can estimate the required memory for packet trace using this formula:

memory required = (statistics overhead) + number of packets * (summary size + data size + packet copy size).

When you enable packet trace, the system allocates a small, fixed amount of memory for statistics. When the system captures per-packet data, each packet summary requires a fixed amount of memory. The total memory consumption depends on how many packets you select for tracing and whether you collect path data and packet copies.

Packet trace display commands

The table provides detailed packet trace information.

These show commands provide detailed packet trace information.

Table 14: Show commands

Command	Description
show platform packet-trace configuration	Displays packet trace configuration, including any defaults.
show platform packet-trace statistics	Displays accounting data for all the traced packets.
show platform packet-trace summary	Displays summary data for the number of packets specified.
show platform packet-trace {*all* | *pkt-num*} [*decode*]	Displays the path data for all the packets or the packet specified. The **decode** option attempts to decode the binary packet into a more human-readable form.

Commands for clearing packet trace data

This section describes the available commands for clearing packet trace data.

Use these commands to clear packet trace data.

Table 15: Clear commands

Command	Description
clear platform packet-trace statistics	Clears the collected packet trace data and statistics.
clear platform packet-trace configuration	Clears the packet trace configuration and the statistics.

Configure packet trace

Perform the following steps to configure the packet trace feature.

Note

The amount of memory consumed by the packet trace feature is affected by the packet trace configuration. You should carefully select the size of per-packet path data and copy buffers and the number of packets to be traced in order to avoid interrupting normal services. You can check the current data-plane DRAM memory consumption by using the **show platform hardware qfp active infrastructure exmem statistics** command.

Procedure

1. **enable**

   Example

   ```
   Router> enable
   ```

   Enables the privileged EXEC mode. Enter your password if prompted.

2. **debug platform packet-trace packet***pkt-num***[fia-trace|summary-only] [circular][data-size***data-size***]**

   Example

   ```
   Router# debug platform packet-trace packets 2048 summary-only
   ```

   Collects summary data for a specified number of packets. Captures feature path data by the default, and optionally performs FIA trace.

   pkt-num –Specifies the maximum number of packets maintained at a given time.

   fia-trace –Provides detailed level of data capture, including summary data, feature-specific data. Also displays each feature entry visited during packet processing.

   summary-only –Enables the capture of summary data with minimal details.

   circular –Saves the data of the most recently traced packets.

   data-size –Specifies the size of data buffers for storing feature and FIA trace data for each packet in bytes. When very heavy packet processing is performed on packets, users can increase the size of the data buffers if necessary. The default value is 2048.

3. **debugplatformpacket-tracepunt**

   Example

   ```
   Router# debug platform packet-trace punt
   ```

   Enables tracing of punted packets from Layer2 to Layer3.

4. **debug platform condition [ipv4|ipv6] [interface***interface***]**
   [access-list*access-list-name*|*ipv4-address*/*subnet-mask*|*ipv6-address*/*subnet-mask***] [ingress|egress]**

Example

```
Router# debug platform condition interface g0/0/0 ingress
```

Specifies the matching criteria for tracing packets. Provides the ability to filter by protocol, IP address and subnet mask, access control list (ACL), interface, and direction.

5. **debug platform condition start**

   Example

   ```
   Router# debug platform condition start
   ```

   Enables the specified matching criteria and starts packet tracing.

6. **debug platform condition stop**

   Example

   ```
   Router# debug platform condition start
   ```

   Deactivates the condition and stops packet tracing.

7. **show platform packet-trace {configuration|statistics|summary|packet {all|***pkt-num***}}**

   Example

   ```
   Router# show platform packet-trace 14
   ```

   Displays packet-trace data according to the specified option. See {start cross reference} Table 21-1 {end cross reference} for detailed information about the **show** command options.

8. **clear platform condition all**

   Example

   ```
   Router(config)# clear platform condition all
   ```

   Removes the configurations provided by the **debug platform condition** and **debug platform packet-trace** commands.

9. **exit**

   Example

   ```
   Router# exit
   ```

   Exits the privileged EXEC mode.

Packet trace is configured. The device will capture and display packet path data per your configuration.

Packet Trace command examples

This example describes how to configure packet trace and display its results.

This example traces incoming packets to Gigabit Ethernet interface 0/0/2. FIA-trace data is captured for the first 128 packets, and the input packets are copied. Use the **show platform packet-trace packet 10** command to display summary data and each feature entry visited during the processing of packet 10.

```
Router>
enable
Router# debug platform packet-trace packet 128 fia-trace
Router# debug platform packet-trace punt
Router# debug platform condition interface g0/0/2 ingress
Router# debug platform condition start
Router# ping to UUT
Router# debug platform condition stop
Router# show platform packet-trace packet 10
Packet: 10           CBUG ID: 52
Summary
  Input     : GigabitEthernet0/0/0
  Output    : internal0/0/rp:1
  State     : PUNT 55  (For-us control)
  Timestamp
  Start   : 597718358383 ns (06/06/2016 09:00:13.643341 UTC)
  Stop    : 597718409650 ns (06/06/2016 09:00:13.643392 UTC)
Path Trace
Feature: IPV4
    Input       : GigabitEthernet0/0/0
    Output      : <unknown>
    Source      : 10.64.68.2
    Destination : 224.0.0.102
    Protocol    : 17 (UDP)
    SrcPort   : 1985
    DstPort   : 1985
  Feature: FIA_TRACE
    Input       : GigabitEthernet0/0/0
    Output      : <unknown>
    Entry       : 0x8a0177bc - DEBUG_COND_INPUT_PKT
    Lapsed time : 426 ns
  Feature: FIA_TRACE
 --More--                                Input       : GigabitEthernet0/0/0
    Output      : <unknown>
    Entry       : 0x8a017788 - IPV4_INPUT_DST_LOOKUP_CONSUME
    Lapsed time : 386 ns
  Feature: FIA_TRACE
    Input       : GigabitEthernet0/0/0
    Output      : <unknown>
    Entry       : 0x8a01778c - IPV4_INPUT_FOR_US_MARTIAN
    Lapsed time : 13653 ns
  Feature: FIA_TRACE
    Input       : GigabitEthernet0/0/0
    Output      : internal0/0/rp:1
    Entry       : 0x8a017730 - IPV4_INPUT_LOOKUP_PROCESS_EXT
    Lapsed time : 2360 ns
  Feature: FIA_TRACE
    Input       : GigabitEthernet0/0/0
    Output      : internal0/0/rp:1
    Entry       : 0x8a017be0 - IPV4_INPUT_IPOPTIONS_PROCESS_EXT
    Lapsed time : 66 ns
  Feature: FIA_TRACE
    Input       : GigabitEthernet0/0/0
```

```
    Output      : internal0/0/rp:1
    Entry       : 0x8a017bfc - IPV4_INPUT_GOTO_OUTPUT_FEATURE_EXT
 --More--                                Lapsed time : 680 ns
  Feature: FIA_TRACE
    Input       : GigabitEthernet0/0/0
    Output      : internal0/0/rp:1
    Entry       : 0x8a017d60 - IPV4_INTERNAL_ARL_SANITY_EXT
    Lapsed time : 320 ns
  Feature: FIA_TRACE
    Input       : GigabitEthernet0/0/0
    Output      : internal0/0/rp:1
    Entry       : 0x8a017a40 - IPV4_VFR_REFRAG_EXT
    Lapsed time : 106 ns
  Feature: FIA_TRACE
    Input       : GigabitEthernet0/0/0
    Output      : internal0/0/rp:1
    Entry       : 0x8a017d2c - IPV4_OUTPUT_DROP_POLICY_EXT
    Lapsed time : 1173 ns
  Feature: FIA_TRACE
    Input       : GigabitEthernet0/0/0
    Output      : internal0/0/rp:1
    Entry       : 0x8a017940 - INTERNAL_TRANSMIT_PKT_EXT
    Lapsed time : 20173 ns
IOSd Path Flow: Packet: 10    CBUG ID: 52
  Feature: INFRA
    Pkt Direction: IN
    Packet Rcvd From CPP
  Feature: IP
    Pkt Direction: IN
    Packet Enqueued in IP layer
    Source      : 10.64.68.2
    Destination : 224.0.0.102
    Interface   : GigabitEthernet0/0/0
  Feature: UDP
    Pkt Direction: IN
    src         : 10.64.68.2(1985)
    dst         : 224.0.0.102(1985)
    length      : 14
Router# clear platform condition all
Router# exit
```

Packet trace output for NAT troubleshooting

The output shows that packets are dropped due to NAT configuration on Gigabit Ethernet interface 0/0/0, which enables you to understand that an issue is occurring on a specific interface. You can use this information to limit which packets to trace, reduce the number of packets for data capture, and increase the level of inspection.

This section displays packet trace output for NAT configuration.

Use packet trace to troubleshoot packet drops for a NAT configuration on a Cisco 8500 Series Catalyst Edge Platform Router. Packet trace provides detailed information to help you diagnose an issue, isolate the cause, and find a solution

If you detect issues but are unsure where to begin troubleshooting, consider accessing the packet trace summary for multiple incoming packets.

```
Router# debug platform condition ingress
Router# debug platform packet-trace packet 2048 summary-only
Router# debug platform condition start
Router# debug platform condition stop
Router# show platform packet-trace summary
Pkt   Input             Output            State  Reason
```

```
0     Gi0/0/2.3060      Gi0/0/2.3060      DROP   402 (NoStatsUpdate)
1     internal0/0/rp:0  internal0/0/rp:0  PUNT   21  (RP<->QFP keepalive)
2     internal0/0/recycle:0  Gi0/0/2.3060 FWD
```

The output shows that packets are dropped due to NAT configuration on Gigabit Ethernet interface 0/0/0, which enables you to understand that an issue is occurring on a specific interface. Using this information, you can limit which packets to trace, reduce the number of packets for data capture, and increase the level of inspection.

```
Router# debug platform packet-trace packet 256
Router# debug platform packet-trace punt
Router# debug platform condition interface Gi0/0/0
Router# debug platform condition start
Router# debug platform condition stop
Router# show platform packet-trace summary
Router# show platform packet-trace 15
Packet: 15          CBUG ID: 238
Summary
Input     : GigabitEthernet0/0/0
Output    : internal0/0/rp:1
State     : PUNT 55  (For-us control)
Timestamp
Start   : 1166288346725 ns (06/06/2016 09:09:42.202734 UTC)
Stop    : 1166288383210 ns (06/06/2016 09:09:42.202770 UTC)
Path Trace
Feature: IPV4
Input       : GigabitEthernet0/0/0
Output      : <unknown>
Source      : 10.64.68.3
Destination : 224.0.0.102
Protocol    : 17 (UDP)
SrcPort   : 1985
DstPort   : 1985
IOSd Path Flow: Packet: 15    CBUG ID: 238
Feature: INFRA
Pkt Direction: IN
Packet Rcvd From CPP
Feature: IP
Pkt Direction: IN
Source      : 10.64.68.122
Destination : 10.64.68.255
Feature: IP
Pkt Direction: IN
Packet Enqueued in IP layer
Source      : 10.64.68.122
Destination : 10.64.68.255
Interface   : GigabitEthernet0/0/0
Feature: UDP
Pkt Direction: IN
src         : 10.64.68.122(1053)
dst         : 10.64.68.255(1947)
length      : 48
```

9 SFP Module and Forward Error Correction

Topics:

- Configure SFP+
- Configure Forward Error Correction

Covers the installation and setup of small form-factor pluggable modules for high-speed data transmission. It includes procedures for configuring forward error correction to improve data reliability and error recovery over long-range links.

Configure SFP+

Use these procedures to configure SFP+ modules and enable error correction on high-speed links. This ensures reliable data transmission and recovers potential errors during long-range communications on supported platforms.

Configure SFP+

Perform this task when installing or reconfiguring SFP+ modules on compatible Cisco devices.

> **Note**
>
> Several Cisco platforms, NIMs, and SM cards support configuring multiple-rate SFPs on the same interface, for example, 1G SFP or 10G SFP+ on a 10G port. In a port-channel bundle, all member interfaces should use the same speed and duplex setting. Use duplex interfaces with the same speed as member interfaces to configure a port-channel.
>
> For more information about interfaces that support multiple-rate SFPs, see the corresponding datasheets.

Procedure

1. **enable***source-interface gigabitethernet slot/port*

   Example

   ```
   Router# enable
   ```

   This command enables privileged EXEC mode. Enter your password if prompted.

2. **configureterminal**

   Example

   ```
   Router# configure terminal
   ```

   Enters the global configuration mode.

3. **interface tengigabitethernet***slot/port*

   Example

   ```
   Router(config)# interface tengigabitethernet 4/11
   ```

   Specifies the 10-Gigabit Ethernet interface to be configured.

   Here:

   slot or port–Specifies the location of the interface.

The SFP+ interface is configured and ready for use.

Configure Forward Error Correction

Forward Error Correction checks and recovers potential errors during long-range data transmission.

Forward Error Correction checks and recovers potential errors during long-range data transmission. Because the Cisco Catalyst 8500L Edge Platforms have long-range SFP, FEC must be configured.

Follow these steps to configure Forward Error Correction:

Procedure

1. **enable***source-interface gigabitethernet slot/port*

   Example

   ```
   Router# enable
   ```

   Enables the privileged EXEC mode. If prompted, enter your password.

2. **configureterminal**

   Example

   ```
   Router# configure terminal
   ```

   Enters the global configuration mode.

3. **interface twentyfivegigabitethernet** *slot/port*

   Example

   ```
   Router(config)# interface twentyfivegigabitethernet0/0/164/11
   ```

   Specifies the 25-gigabit Ethernet interface to be configured.

   Here:

   slot/port specifies the location of the interface.

4. **fec { auto | cl108 | cl74 | off}**

   Example

   ```
   Router(config)#  fec cl74-fc
   ```

   Configuring `fec cl74` enables Clause 74 FC-FEC on the interface, which helps correct errors in the data stream to maintain signal integrity and performance.

   The fec command supports these modes:

   - auto–Enables FEC based on SFP type
   - cl108–Enables clause108 <= RS-FEC(528,514)

- cl74–Enables clause74 <= FC-FEC
- disable–Disables FEC on interface

> **Note**
>
> - The **fec** command is only applicable to 25G links.
> - For 10- or 25-Gbps dual-rate SFP modules, if the speed changes from 25 Gbps to 10 Gbps, remove the **fec** configuration before changing the speed.

What to do next

Minimum supported release: Cisco IOS-XE Release 26.2

You can verify the FEC type from the **show interface** command only for 100G interfaces.

10 Cisco ThousandEyes Enterprise Agent Application

Topics:

- Cisco ThousandEyes Enterprise agent application

Explains how to host and manage network intelligence agents as containerized applications on routing platforms. It provides a complete workflow for installation, configuration, and troubleshooting to gain deep visibility into digital experiences and network paths.

Cisco ThousandEyes Enterprise agent application

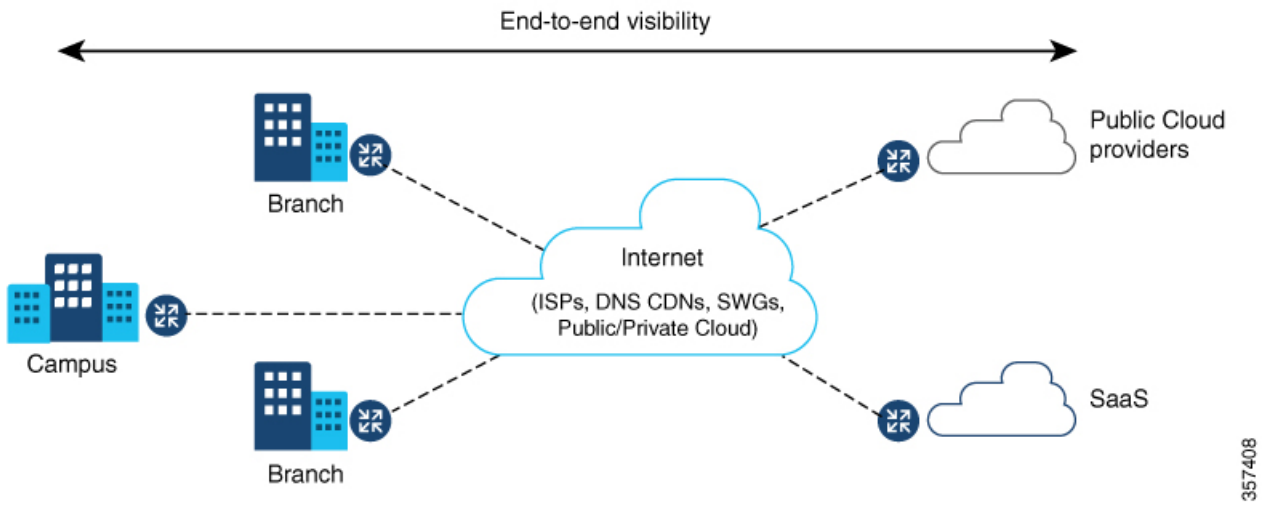
Outlines the workflow for hosting the Cisco ThousandEyes Enterprise Agent as a container application on the platform, including installation, configuration, and troubleshooting.

Cisco ThousandEyes is a network intelligence platform that

- allows you to use its agents to run a variety of tests to monitor network and application performance,
- enables you to view end-to-end paths across networks and services that impact your business,
- actively monitors network traffic paths across internal, external, and internet networks in real time to analyze network performance, and
- provides application-availability insights enriched with routing and device data for a multidimensional view of digital experience.

From Cisco IOS XE Release 17.6.1, you can use application-hosting capabilities to deploy the Cisco ThousandEyes Enterprise Agent as a container application on Cisco Catalyst 8300 and Catalyst 8200 Series Edge Platforms. This agent application runs as a docker image using Cisco IOx docker-type option. For more information on how to configure Cisco ThousandEyes in controller mode, see Cisco SD-WAN Systems and Interfaces Configuration Guide

Figure 2: Network View through ThousandEyes Application



Supported platforms and system requirements

Supported platforms and system requirements are listed.

Table 16: Supported Platforms and System Requirements

Platforms	Bootflash	FRU Storage	DRAM
Catalyst 8300 Series Edge Platforms			
C8300-1N1S-6T	8 GB	16 GB M.2 USB (Default)	8 GB
C8300-1N1S-4T2X	8 GB	16 GB M.2 USB (Default)	8 GB
C8300-2N2S-6T	8 GB	16 GB M.2 USB (Default)	8 GB
C8300-2N2S-4T2X	8 GB	16 GB M.2 USB (Default)	8 GB

Platforms	Bootflash	FRU Storage	DRAM
Catalyst 8200 Series Edge Platforms			
C8200-1N-4T	8 GB	16 GB M.2 USB (Default)	8 GB
C8200L-1N-4T	8 GB	16 GB M.2 USB (Recommended)	8 GB

Note

The minimum DRAM and storage requirement for running Cisco ThousandEyes Enterprise Agent is 8 GB. If the device does not have enough memory or storage, we recommend that you upgrade DRAM or add an external storage such as M.2 USB. When the available resources are not sufficient to run other applications, Cisco IOx generates an error message.

Workflow to install and run the Cisco ThousandEyes application

This task outlines the workflow to install and run the Cisco ThousandEyes Enterprise Agent application on a device

To install and run the Cisco ThousandEyes image on a device, perform these steps:

Procedure

1. Create a new account on the Cisco ThousandEyes portal.
2. Download the Cisco ThousandEyes application package from the software downloads page and ensure that you use the agent version 4.0.2.
3. Copy the image on the device.
4. Install and launch the image.
5. Connect the agent to the controller.

Note

When you order platforms that support Cisco ThousandEyes application with Cisco IOS XE 17.6.1 software, the Cisco ThousandEyes application package is available in the bootflash of the device.

The Cisco ThousandEyes Enterprise Agent application is successfully installed and running on your device.

Workflow to host the Cisco ThousandEyes application

This task provides a detailed workflow to host and configure the Cisco ThousandEyes Enterprise Agent application on a device.

Use this workflow to install, configure, and launch the Cisco ThousandEyes Enterprise Agent application on your device.

To install and launch the application, perform these steps:

Before you begin

Create a new account on the Cisco ThousandEyes portal and generate the token. The Cisco ThousandEyes agent application uses this token to authenticate and check into the correct Cisco ThousandEyes account. If you receive a message stating that your token is invalid and you want to troubleshoot the issue, see Troubleshoot the Cisco ThousandEyes application on page 117.

Note

If you configure the correct token and Domain Name Server (DNS) information, the device is discovered automatically.

Procedure

1. Enable the Cisco IOx application environment on the device.
   - Use these commands for non-SD-WAN (autonomous mode) images:

```
config terminal
 iox
end
write
```

   - Use these commands for SD-WAN (controller mode) images:

```
config-transaction
iox
commit
```

2. If the IOx command is accepted, wait for a few seconds and check whether the IOx process is up and running by using the **show iox** command. The output must display that the show IOxman process is running.

```
Device #show iox

IOx Infrastructure Summary:
---------------------------
IOx service (CAF) 192.0.2.8     : Running
IOx service (HA)               : Not Supported
IOx service (IOxman)           : Running
IOx service (Sec storage)      : Not Supported
Libvirtd 1.3.4                 : Running
```

3. Ensure that the ThousandEyes application LXC tarball is available in the device's *bootflash:*.
4. Create a virtual port group interface to enable the traffic path to the Cisco ThousandEyes application:

```
interface VirtualPortGroup 0
          ip address 192.0.2.22 255.255.255.0
        exit
```

5. Configure the app-hosting application with the generated token:

```
app-hosting appid te
          app-vnic gateway1 virtualportgroup 0 guest-interface 0
          guest-ipaddress 192.0.2.22 netmask 255.255.255.0
          app-default-gateway 192.0.2.22 guest-interface 0
          app-resource docker
             prepend-pkg-opts  Required to get the default run-time options
 from package.yaml
             run-opts 1 "--hostname thousandeyes"
             run-opts 2 "-e TEAGENT_ACCOUNT_TOKEN=<ThousandEyes token>"
       run-opts 3 "-e TEAGENT_PROXY_TYPE=STATIC -e
TEAGENT_PROXY_LOCATION=proxy.something.other:80"
          name-server0 192.0.2.10  ISP's DNS server
        end

app-hosting appid te
 app-resource docker
  prepend-pkg-opts
  run-opts 2 "--hostname
```

> **Note**
>
> You can use the proxy configuration only if the Cisco ThousandEyes agent does not have an internet access without a proxy. Also, the hostname is optional. If you do not provide the hostname during the installation, the device hostname is used as the Cisco ThousandEyes agent hostname. The device hostname is displayed on the Cisco ThousandEyes portal. The DNS name server information is optional. If the Cisco ThousandEyes agent uses a private IP address, ensure that you establish a connection to the device through NAT.

6. Configure the **start** command to run the application automatically when the application is installed on the device using the **install** command:

```
app-hosting appid te
         start
```

7. Install the ThousandEyes application:

```
app-hosting install appid <appid> package [bootflash: | harddisk: | https:]
```

Select a location to install the ThousandEyes application from these options:

```
Device# app-hosting install appid te package ?
          bootflash:  Package path  ISR4K case if image is locally available
 in bootflash:
          harddisk:   Package path  Cat8K case if image is locally available
 in M.2 USB
           https:      Package path  Download over the internet if image is
 not locally present in router. URL to ThousandEyes site hosting agent image
 to be provided here
```

8. Check if the application is up and running:

```
Device#show app-hosting list
```

```
 App id                                          State
------------------------------------------------------------
  te                                              RUNNING
```

> **Note**
>
> If any of these steps fail, use the **show logging** command and check the IOx error message. If the error message is about insufficient disk space, clean the storage media (bootflash or hard disk) to free up the space. Use the **show app-hosting resource** command to check the CPU and disk memory.

The Cisco ThousandEyes application is successfully hosted and running on your device.

Download and copy the image to the device

This task describes how to download the Cisco ThousandEyes application image and copy it to the device's bootflash.

Use this task to ensure the Cisco ThousandEyes application image is correctly placed on your device for installation.

To download and copy the image to bootflash, perform these steps:

Procedure

---

1. Check if the Cisco ThousandEyes image is precopied to *bootflash:/<directory name>*.
2. If the image is not available in the device directory, perform these steps:
   a) If the device has a direct access to internet, use the *https:.* option in the **application install** command. This option downloads the image from the Cisco ThousandEyes software downloads page into *bootflash:/apps* and installs the application.

```
Device# app-hosting install appid <appid string> package [bootflash: |
flash | http | https://  | ftp | ] URL to image location hosted on
ThousandEyes portal

Device# app-hosting install appid te1000 package
https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-4.0.2.cisco.tar

Installing package
'https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-4.0.2.cisco.tar'
 for 'te1000'.

Use 'show app-hosting list' for progress.
*Jun 29 23:43:29.244: %IOSXE-6-PLATFORM: R0/0: IOx:  App verification
successful
*Jun 29 23:45:00.449: %IM-6-INSTALL_MSG: R0/0: ioxman: app-hosting: Install
 succeeded: te1000 installed successfully Current state is DEPLOYED
*Jun 29 23:45:01.801: %IOSXE-6-PLATFORM: R0/0: IOx:  App verification
successful
*Jun 29 23:45:51.054: %IM-6-START_MSG: R0/0: ioxman: app-hosting: Start
succeeded: te1000 started successfully Current state is RUNNING

Device#show app-hosting detail appid te1000  ( Details of Application)
App id                  : te1000
Owner                   : iox
```

```
State                  : RUNNING
Application
  Type                 : docker
  Name                 : ThousandEyes Enterprise Agent
  Version              : 4.0
  Author               : ThousandEyes <support@thousandeyes.com>
  Path                 :
bootflash:thousandeyes-enterprise-agent-4.0-22.cisco.tar
Resource reservation
  Memory               : 500 MB
  Disk                 : 1 MB
  CPU                  : 1500 units
  CPU-percent          : 70 %
```

b) If the device has a proxy server, copy the image manually to *bootflash:/apps*.
c) Download the Cisco ThousandEyes application package from the software downloads page and ensure that you use the agent version 4.0.2.
d) Create an application directory in the *bootflash:* to copy the image:

```
Device# mkdir bootflash:apps
Create directory filename [apps]?
Created dir bootflash:/apps
```

e) Copy the Cisco ThousandEyes image to the *bootflash:apps* directory.
f) Validate the image using the **verify** command:

```
verify /md5 bootflash:apps/<file name>
```

The Cisco ThousandEyes image is successfully downloaded and copied to the device's bootflash.

Connect the Cisco ThousandEyes Agent with the controller

This task describes how the Cisco ThousandEyes agent connects to its controller and provides troubleshooting guidance for connectivity issues.

This task explains the process of the Cisco ThousandEyes agent connecting to its controller in the cloud environment.

To connect the Cisco ThousandEyes agent with the controller, perform these steps:

Before you begin

Ensure that you have an Internet connection before you connect the agent with the controller.

Procedure

After the Cisco ThousandEyes application is up and running, the agent (ThousandEyes-agent) process connects to the controller that is running on the cloud environment.

> **Note**
>
> If you have issues related to connectivity, the application logs the relevant error messages in the application-specific logs (/var/logs).

The Cisco ThousandEyes agent is successfully connected to the controller.

Modify the agent parameters

This task describes the steps to modify the configuration parameters for an existing application agent.

Use this task to update the configuration settings of an application agent, which requires stopping and restarting the application.

To modify the agent parameters, perform these actions:

Procedure

1. Stop the application using the **app-hosting stop appid appid** command.
2. Deactivate the application using the **app-hosting deactivate appid appid** command.
3. Make the required changes to the app-hosting configuration.
4. Activate the application using the **app-hosting activate appid appid** command.
5. Start the application using the **app-hosting start appid appid** command.

The agent parameters have been successfully modified and the application is running with the new configuration.

Uninstall the application

This task describes the steps to completely remove an application from the device.

Use this task to completely remove a previously installed application from your device.

To uninstall the application, perform these steps:

Procedure

1. Stop the application using the **app-hosting stop appid te** command.
2. Check if the application is in active state using the **show app-hosting list** command.
3. Deactivate the application using the **app-hosting deactivate appid te** command.
4. Ensure that the application is not in active state. Use the **show app-hosting list** command to check status of the application.
5. Uninstall the application using the **app-hosting uninstall appid te** command.
6. After the uninstallation process is complete, use the **show app-hosting list** command to check if the application is uninstalled successfully.

The application has been successfully uninstalled from the device.

Troubleshoot the Cisco ThousandEyes application

To troubleshoot the Cisco ThousandEyes application, perform these steps.

- Connect to Cisco ThousandEyes agent application using the **app-hosting connect appid appid session /bin/bash** command.
- Verify the configuration applied to the application /etc/te-agent.cfg.
- View the logs in /var/log/agent/te-agent.log. You can use these logs to troubleshoot the configuration.

When the Cisco ThousandEyes application is in running state, it is registered on the ThousandEyes portal. If the application does not show up in a few minutes after the agent is in running state, check using the **app-hosting connect appid thousandeyes_enterprise_agent session** command.

> **Note**
>
> Check the DNS server connection. If the Cisco ThousandEyes agent is assigned to a private IP address, check the NAT configuration.

```
Device#app-hosting connect appid thousandeyes_enterprise_agent session
Device#cat /var/log/agent/te-agent.log
2021-02-04 08:59:29.642 DEBUG [e4736a40] [te.agent.AptPackageInterface] {}
Initialized APT package interface
2021-02-04 08:59:29.642 INFO  [e4736a40] [te.agent.main] {} Agent version
1.103.0 starting.  Max core size is 0 and max open files is 1024
2021-02-04 08:59:29.642 DEBUG [e4736a40] [te.agent.db] {} Vacuuming database
2021-02-04 08:59:29.643 INFO  [e4736a40] [te.agent.db] {} Found version 0,
expected version 50
2021-02-04 08:59:29.672 INFO  [e4708700] [te.probe.ServerTaskExecutor] {}
ProbeTaskExecutor started with 2 threads.
2021-02-04 08:59:29.673 INFO  [e2f05700] [te.probe.ProbeTaskExecutor.bandwidth]
 {} ProbeTaskExecutor started with 1 threads.
2021-02-04 08:59:29.673 INFO  [e2704700] [te.probe.ProbeTaskExecutor.realtime]
 {} ProbeTaskExecutor started with 1 threads.
2021-02-04 08:59:29.673 INFO  [e1f03700] [te.probe.ProbeTaskExecutor.throughput]
 {} ProbeTaskExecutor started with 1 threads.
2021-02-04 08:59:29.674 DEBUG [e4736a40] [te.agent.DnssecTaskProceessor] {}
Agent is not running bind
2021-02-04 08:59:29.674 DEBUG [e4736a40] [te.snmp.RequestDispatcher] {}
Initialised SNMP++ session
2021-02-04 08:59:29.674 DEBUG [e4736a40] [te.snmp.RequestDispatcher] {}
Initialised SNMP++ session
2021-02-04 08:59:29.674 DEBUG [e4736a40] [te.snmp.RequestDispatcher] {}
Initialised SNMP++ session
2021-02-04 08:59:29.674 INFO  [e4736a40] [te.agent.main] {} Agent starting up
2021-02-04 08:59:29.675 INFO  [e4736a40] [te.agent.main] {} No agent id found,
 attempting to obtain one
2021-02-04 08:59:29.675 INFO  [e4736a40] [te.agent.ClusterMasterAdapter] {}
Attempting to get agent id from sc1.thousandeyes.com
2021-02-04 08:59:29.679 ERROR [e4736a40] [te.agent.main] {} Error calling
create_agent: Curl error - Couldn't resolve host name
2021-02-04 08:59:29.680 INFO  [e4736a40] [te.agent.main] {} Sleeping for 30
seconds
```