

Basic platform configuration

This section includes information about some basic platform configuration in Autonomous mode, and contains these sections:

- Default configuration, on page 1
- Configuring global parameters, on page 15
- Configuring Gigabit Ethernet interfaces, on page 16
- Configuring a loopback interface, on page 17
- Configuring module interfaces, on page 19
- Configure dynamic allocation of cores, on page 19
- Enabling Cisco Discovery Protocol, on page 22
- Configuring command-line access, on page 22
- Configuring static routes, on page 24
- Configuring dynamic routes, on page 26
- Enable Auto-Off Port LED, on page 31
- Enable Blue Beacon LED, on page 31

Default configuration

When you boot up the device in autonomous mode, the device looks for a default file name-the PID of the device. For example, the Cisco 8300 Series Secure Routers look for a file named C8375-E-G2.cfg or the C8355-G2.cfg. The device looks for this file before finding the standard files-router-confg or the ciscortr.cfg.

The device looks for the C8375-E-G2.cfg or the C8355-G2.cfg file in the bootflash. If the file is not found in the bootflash, the device then looks for the standard files-router-confg and ciscortr.cfg. If none of the files are found, the device then checks for any inserted USB that may have stored these files in the same particular order.

Note If there is a configuration file with the PID as its name in an inserted USB, but one of the standard files are in bootflash, the system finds the standard file for use.

Use the **show running-config** command to view the initial configuration. This example shows an example ouptut for C8375-E-G2.

```
Router# show running-config
Current configuration : 6621 bytes
```

```
!
! Last configuration change at 06:24:36 UTC Fri Feb 7 2025 by admin
!
version 17.15
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
!
hostname router
!
boot-start-marker
boot-end-marker
!
!
vrf definition Mgmt-intf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
no logging console
no aaa new-model
!
no ip domain lookup
!
!
!
!
!
!
!
!
!
login on-success log
!
!
!
!
!
!
!
!
subscriber templating
!
!
!
crypto pki trustpoint TP-self-signed-2220840378
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-2220840378
 revocation-check none
 rsakeypair TP-self-signed-2220840378
 hash sha512
!
crypto pki trustpoint SLA-TrustPoint
 enrollment pkcs12
 revocation-check crl
 hash sha512
!
!
crypto pki certificate chain TP-self-signed-2220840378
 certificate self-signed 01
  30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 0D050030
  31312F30 2D060355 04030C26 494F532D 53656C66 2D536967 6E65642D 43657274
```

```
  69666963 6174652D 32323230 38343033 3738301E 170D3235 30313039 30393132
  31315A17 0D333530 31303930 39313231 315A3031 312F302D 06035504 030C2649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 32323038
  34303337 38308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201
  0A028201 01008F2E D295CE5D 6DFDC027 4E7B4410 CD546B85 C14F0844 A4A08A47
  3621C3A8 4AF11F97 9489AD4B 00E1C57F AEAD53CE B08B684A 9018E660 8BCFABCE
  B1DCD79D 86E78BF4 DF278EF3 6C86539E 97217942 05C48B9A CBB057FB FFB2B225
  5A626C11 091376D8 A81E66B2 36ECE937 B44451F5 49D9CBB7 4D674A87 6532F4A7
  0A047D14 481A98A7 15574BE5 BFFFB4B1 F397C982 FECED50C 59605382 39B317F2
  3183C1B4 B83F62CF 3A9D6EE8 A1A34C61 86AD6B15 5474FD41 3151540D 5E387FC8
  B169558A E0DF905E F1187E78 AB59BD67 A38E97D9 79AAF825 E6D2B3A6 CF9239D6
  8B5F7E7D D4645263 F6006E12 FF69C3AF 7B769A2E F7F099AE 03A336EA 294A0423
  748E52EF 99330203 010001A3 53305130 1D060355 1D0E0416 04149FE1 4E1985FF
  AB1E7167 F6A67B35 5F3353E3 5B88301F 0603551D 23041830 1680149F E14E1985
  FFAB1E71 67F6A67B 355F3353 E35B8830 0F060355 1D130101 FF040530 030101FF
  300D0609 2A864886 F70D0101 0D050003 82010100 4F0CF81D C9E72E8B 2D5BC14A
  862DF349 42772862 46777631 3F402A07 DCD34CF7 5ED43C42 3C1839BB B68B0677
  C0C66B83 E97A0980 A54E5444 F0473525 C592D1C0 4D6C101A DA4BCDA0 D9C36EE1
CAD752AB AA37B084 A6C5F926 ED264D20 F6EF4940 F1103FAF 7122F428 0A5221F4
  DFB69177 BD7F5E67 DF662F1A F7888526 8867A938 C7F0B75B C34CDAFB 4AA2386B
  10ECE4FD 348D2028 E66E2FF1 FB6B0089 3D68FB71 E993D055 47CC0AA9 F08586E3
  319C0C26 86082E0A E4A9D4DA 99727580 6BEA0CF3 E530CD60 BBC627C5 16D8B483
  A96D47F4 B4746157 0DD2829E 7FC7E087 BE22D84B 09EDD9D7 A2D09897 247397B5
  AB6BBA3C E37BEDA0 053DE14A 748502E1 510197E4
        quit
crypto pki certificate chain SLA-TrustPoint
 certificate ca 01
  30820321 30820209 A0030201 02020101 300D0609 2A864886 F70D0101 0B050030
  32310E30 0C060355 040A1305 43697363 6F312030 1E060355 04031317 43697363
  6F204C69 63656E73 696E6720 526F6F74 20434130 1E170D31 33303533 30313934
  3834375A 170D3338 30353330 31393438 34375A30 32310E30 0C060355 040A1305
  43697363 6F312030 1E060355 04031317 43697363 6F204C69 63656E73 696E6720
  526F6F74 20434130 82012230 0D06092A 864886F7 0D010101 05000382 010F0030
  82010A02 82010100 A6BCBD96 131E05F7 145EA72C 2CD686E6 17222EA1 F1EFF64D
  CBB4C798 212AA147 C655D8D7 9471380D 8711441E 1AAF071A 9CAE6388 8A38E520
  1C394D78 462EF239 C659F715 B98C0A59 5BBB5CBD 0CFEBEA3 700A8BF7 D8F256EE
  4AA4E80D DB6FD1C9 60B1FD18 FFC69C96 6FA68957 A2617DE7 104FDC5F EA2956AC
  7390A3EB 2B5436AD C847A2C5 DAB553EB 69A9A535 58E9F3E3 C0BD23CF 58BD7188
  68E69491 20F320E7 948E71D7 AE3BCC84 F10684C7 4BC8E00F 539BA42B 42C68BB7
  C7479096 B4CB2D62 EA2F505D C7B062A4 6811D95B E8250FC4 5D5D5FB8 8F27D191
  C55F0D76 61F9A4CD 3D992327 A8BB03BD 4E6D7069 7CBADF8B DF5F4368 95135E44
  DFC7C6CF 04DD7FD1 02030100 01A34230 40300E06 03551D0F 0101FF04 04030201
  06300F06 03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604 1449DC85
  4B3D31E5 1B3E6A17 606AF333 3D3B4C73 E8300D06 092A8648 86F70D01 010B0500
  03820101 00507F24 D3932A66 86025D9F E838AE5C 6D4DF6B0 49631C78 240DA905
  604EDCDE FF4FED2B 77FC460E CD636FDB DD44681E 3A5673AB 9093D3B1 6C9E3D8B
  D98987BF E40CBD9E 1AECA0C2 2189BB5C 8FA85686 CD98B646 5575B146 8DFC66A8
  467A3DF4 4D565700 6ADF0F0D CF835015 3C04FF7C 21E878AC 11BA9CD2 55A9232C
  7CA7B7E6 C1AF74F6 152E99B7 B1FCF9BB E973DE7F 5BDDEB86 C71E3B49 1765308B
  5FB0DA06 B92AFE7F 494E8A9E 07B85737 F3A58BE1 1A48A229 C37C1E69 39F08678
  80DDCD16 D6BACECA EEBC7CF9 8428787B 35202CDC 60E4616A B623CDBD 230E3AFB
  418616A9 4093E049 4D10AB75 27E86F73 932E35B5 8862FDAE 0275156F 719BB2F0
  D697DF7F 28
        quit
!
!
diagnostic bootup level minimal
!
license udi pid C8375-E-G2 sn FDO2833M01A
memory free low-watermark processor 63953
!
spanning-tree extend system-id
!
!
```

```
username admin privilege 15 password 0 admin
!
redundancy
 mode none
!
!
!
!
vlan internal allocation policy ascending
!
!
!
!
!
!
interface TwoGigabitEthernet0/0/0
 no ip address
 shutdown
 negotiation auto
!
interface TwoGigabitEthernet0/0/1
 no ip address
 shutdown
 negotiation auto
!
interface TwoGigabitEthernet0/0/2
 no ip address
 shutdown
 negotiation auto
!
interface TwoGigabitEthernet0/0/3
 no ip address
 shutdown
 negotiation auto
!
interface TenGigabitEthernet0/0/4
 no ip address
 shutdown
!
interface TenGigabitEthernet0/0/5
 no ip address
 shutdown
!
interface TwoGigabitEthernet0/1/0
!
interface TwoGigabitEthernet0/1/1
!
interface TwoGigabitEthernet0/1/2
!
interface TwoGigabitEthernet0/1/3
!
interface TwoGigabitEthernet0/1/4
!
interface TwoGigabitEthernet0/1/5
!
interface TwoGigabitEthernet0/1/6
 switchport
!
interface TwoGigabitEthernet0/1/7
 switchport
!
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 ip address 10.79.58.164 255.255.255.0
```

```
 negotiation auto
!
interface Vlan1
 no ip address
!
ip forward-protocol nd
ip tftp source-interface GigabitEthernet0
ip http server
ip http authentication local
ip http secure-server
ip route 64.104.134.61 255.255.255.255 10.79.58.1
ip route vrf Mgmt-intf 0.0.0.0 0.0.0.0 10.79.58.1
ip route vrf Mgmt-intf 0.0.0.0 0.0.0.0 64.104.134.61
ip ssh bulk-mode 131072
!
snmp-server community public RW
!
!
control-plane
!
!
line con 0
 exec-timeout 0 0
 stopbits 1
line aux 0
line vty 0 4
 exec-timeout 0 0
 privilege level 15
 login local
 transport input telnet
line vty 5 10
 privilege level 15
 login local
 transport input telnet
line vty 11 14
 login
 transport input ssh
!
!
!
!
!
!
!
end
```

This example shows an example ouptut for C8355-G2.

```
Router# show running-config
Current configuration : 5001 bytes
 !

! Last configuration change at 09:02:29 UTC Mon Aug 18 2025

!

version 17.18

service timestamps debug datetime msec

service timestamps log datetime msec

service password-encryption

platform qfp utilization monitor load 80
```

```
platform resource data-plane-heavy
!
hostname Wilson1
!
boot-start-marker
boot-end-marker
!
!
vrf definition 4
!
address-family ipv4
exit-address-family
!
vrf definition 50
!
address-family ipv4
exit-address-family
!
vrf definition 65500
 !
address-family ipv4
exit-address-family
!
vrf definition 65528
description SIG VRF
!
address-family ipv4
exit-address-family
!
vrf definition 65529
description Speedtest VRF
!
```

```
address-family ipv4
exit-address-family
!
vrf definition Mgmt-intf
!
address-family ipv4
exit-address-family
!
address-family ipv6
exit-address-family
!
no logging queue-limit
no logging rate-limit
aaa new-model
!
!
aaa authentication login default local
aaa authorization console
aaa authorization exec default local
!
!
aaa session-id common
no process cpu extended history
!
!
!
!
!
!
!
!
ip dhcp pool PnPWebUI1
vrf 65500
```

```
host 192.168.1.3 255.255.255.0
client-identifier 0077.6562.7569
dns-server 192.168.1.1
!
!
!
login on-success log
!
!
!
!
!
!
fhrp version vrrp v3
ipv6 unicast-routing
ipv6 rip vrf-mode enable
!
!
subscriber templating
!
!
!
!
!
!
!
!
!
!
!
!
!
!
```

```
!
!
crypto pki trustpoint SLA-TrustPoint
 enrollment pkcs12
revocation-check crl
hash sha512
!
crypto pki trustpoint TP-self-signed-758392875
enrollment selfsigned
revocation-check crl
rsakeypair TP-self-signed-758392875
hash sha512
!
!
crypto pki certificate chain SLA-TrustPoint
crypto pki certificate chain TP-self-signed-758392875
!
!
!
!
!
!
!
!
diagnostic bootup level minimal
!
license udi pid C8355-G2 sn FDO2836M06D
license accept end user agreement
memory free low-watermark processor 63115
!
spanning-tree extend system-id
```

```
!
!

!

!

username admin privilege 15 secret 9
$9$nbNcaolXwfHkHk$FxUuHozuwCqmWCyltQ0YkluTGUubx2ijED8/Laeh72k
!

redundancy

 mode none

!

!

!

!

!

no crypto ikev2 diagnose error

!

!

vlan internal allocation policy ascending

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

!

interface Loopback65528
```

```
vrf forwarding 65528
ip address 192.168.1.1 255.255.255.255
!
interface Loopback65529
vrf forwarding 65529
ip address 11.1.0.116 255.255.255.255
!
interface Tunnel1
ip unnumbered GigabitEthernet0/0/4
ipv6 unnumbered GigabitEthernet0/0/4
tunnel source GigabitEthernet0/0/4
tunnel mode sdwan
!
interface Tunnel100009
ip unnumbered TenGigabitEthernet0/0/9
tunnel source TenGigabitEthernet0/0/9
tunnel mode sdwan
!
interface FiveGigabitEthernet0/0/0
no ip address
negotiation auto
!
interface FiveGigabitEthernet0/0/1
no ip address
 negotiation auto
!
interface FiveGigabitEthernet0/0/2
no ip address
negotiation auto
!
interface FiveGigabitEthernet0/0/3
no ip address
```

```
!
interface GigabitEthernet0/0/4
ip address 10.1.12.116 255.255.255.0
negotiation auto
!
interface GigabitEthernet0/0/5
no ip address
negotiation auto
!
interface TenGigabitEthernet0/0/6
!
interface TenGigabitEthernet0/0/7
!
interface TenGigabitEthernet0/0/8
vrf forwarding 50
ip address 8.2.1.1 255.255.255.0
no plim qos input queue 0 pause enable
!
!
interface TenGigabitEthernet0/0/9
mtu 1734
ip address 10.1.9.1 255.255.255.0
no plim qos input queue 0 pause enable
!
interface GigabitEthernet0
vrf forwarding Mgmt-intf
ip address 10.75.163.116 255.255.255.0
negotiation auto
ipv6 address autoconfig
!
 interface Vlan1
 vrf forwarding 65500
 ip address 192.168.1.1 255.255.255.0
```

```
!
router omp
!
ip forward-protocol nd
no ip forward-protocol udp
ip tftp source-interface GigabitEthernet0
ip ftp source-interface GigabitEthernet0
ip http server
ip http authentication local
ip http secure-server
!
ip nat settings central-policy
ip nat settings gatekeeper-size 1024
ip nat route vrf 65528 0.0.0.0 0.0.0.0 global
no ip nat service all-algs
ip route 0.0.0.0 0.0.0.0 10.1.9.2
ip route vrf Mgmt-intf 0.0.0.0 0.0.0.0 10.75.163.1
ip route vrf 50 0.0.0.0 0.0.0.0 10.1.9.2
no ip ssh bulk-mode
ip scp server enable
!
no ipv6 mld ssm-map query dns
!
!
!
!
!
control-plane
!
!
mgcp behavior rsip-range tgcp-only
mgcp behavior comedia-role none
```

```
mgcp behavior comedia-check-media-src disable
mgcp behavior comedia-sdp-force disable
!

mgcp profile default
!
 !
!
!
!
!
 !
!
line con 0
exec-timeout 0 0
activation-character 13
stopbits 1
speed 115200
line aux 0
activation-character 13
line vty 0 4
privilege level 15
activation-character 13
transport input all
line vty 5 80
privilege level 15
activation-character 13
transport input none
!
no network-clock revertive
!
!
!
```

```
!
!
!
!
!
!
!
!
 !
 netconf-yang
netconf-yang feature candidate-datastore
no netconf-yang ssh server algorithm encryption aes128-cbc
no netconf-yang ssh server algorithm encryption aes256-cbc
no netconf-yang ssh server algorithm hostkey ssh-rsa
no netconf-yang ssh server algorithm kex diffie-hellman-group14-sha1
no netconf-yang ssh server algorithm mac hmac-sha1
end
```

Configuring global parameters

To configure the global parameters for your device, follow these steps.

SUMMARY STEPS

1. **configure terminal**
2. **hostname** *name*
3. **enable secret** *password*
4. **no ip domain-lookup**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Router> enable`	Enters global configuration mode when using the console port. Use the following to connect to the device with a remote terminal:

	Command or Action	Purpose

	`Router# configure terminal`
	`Router(config)#`
		`telnet router-name or address`
		`Login: login-id`
		`Password: *********`
		`Router> enable`

Step 2	**hostname** *name* **Example:** `Router(config)# hostname Router`	Specifies the name for the device.
Step 3	**enable secret** *password* **Example:** `Router(config)# enable secret cr1ny5ho`	Specifies an encrypted password to prevent unauthorized access to the device.
Step 4	**no ip domain-lookup** **Example:** `Router(config)# no ip domain-lookup`	Disables the device from translating unfamiliar words (typos) into IP addresses. For complete information on global parameter commands, see the Cisco IOS Release Configuration Guide documentation set.

Configuring Gigabit Ethernet interfaces

To manually define onboard Gigabit Ethernet interfaces, follow these steps, beginning from global configuration mode.

SUMMARY STEPS

1. **interface TwoGigabitEthernet** *slot/bay/port*
2. **ip address** *ip-address mask*
3. **ipv6 address** *ipv6-address/prefix*
4. **no shutdown**
5. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**interface TwoGigabitEthernet** *slot/bay/port* **Example:** `Router(config)# interface TwoGigabitEthernet 0/0/1`	Enters the configuration mode for a Gigabit Ethernet interface on the device.

	Command or Action	Purpose
Step 2	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.0.2.2 255.255.255.0`	Sets the IP address and subnet mask for the specified Gigabit Ethernet interface. Use this Step if you are configuring an IPv4 address.
Step 3	**ipv6 address** *ipv6-address/prefix* **Example:** `Router(config-if)# ipv6 address 2001.db8::ffff:1/128`	Sets the IPv6 address and prefix for the specified Gigabit Ethernet interface. Use this step instead of Step 2, if you are configuring an IPv6 address.
Step 4	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the Gigabit Ethernet interface and changes its state from administratively down to administratively up.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits configuration mode for the Gigabit Ethernet interface and returns to privileged EXEC mode.

Configuring a loopback interface

Before you begin

The loopback interface acts as a placeholder for the static IP address and provides default routing information.

To configure a loopback interface, follow these steps.

SUMMARY STEPS

1. **interface** *type number*
2. (Option 1) **ip address** *ip-address mask*
3. (Option 2) **ipv6 address** *ipv6-address/prefix*
4. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**interface** *type number* **Example:**	Enters configuration mode on the loopback interface.

	Command or Action	Purpose
	`Router(config)# interface Loopback 0`	
Step 2	(Option 1) **ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.108.1.1 255.255.255.0`	Sets the IP address and subnet mask on the loopback interface. (If you are configuring an IPv6 address, use the **ipv6 address** *ipv6-address/prefix* command described below.
Step 3	(Option 2) **ipv6 address** *ipv6-address/prefix* **Example:** `Router(config-if)# 2001:db8::ffff:1/128`	Sets the IPv6 address and prefix on the loopback interface.
Step 4	**exit** **Example:** `Router(config-if)# exit`	Exits configuration mode for the loopback interface and returns to global configuration mode.

Example

Verifying Loopback Interface Configuration

This configuration example shows the loopback interface configured on the Gigabit Ethernet interface with an IP address of 203.0.113.1/32, which acts as a static IP address. The loopback interface points back to virtual-template1, which has a negotiated IP address.

```
!
interface loopback 0
ip address 203.0.113.1 255.255.255.255 (static IP address)
ip nat outside
!
interface Virtual-Template1
ip unnumbered loopback0
no ip directed-broadcast
ip nat outside
```

Enter the **show interface loopback** command. You should see an output similar to this example:

```
Router# show interface loopback 0
Loopback0 is up, line protocol is up
  Hardware is Loopback
  Internet address is 203.0.113.1/32
  MTU 1514 bytes, BW 8000000 Kbit/sec, DLY 5000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation LOOPBACK, loopback not set
  Keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
```

```
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     Output 0 broadcasts (0 IP multicasts)
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Alternatively, use the **ping** command to verify the loopback interface, as shown in this example:

```
Router# ping  203.0.113.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 203.0.113.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Configuring module interfaces

For detailed information about configuring service modules, see "Service Modules" in the "Service Module Management" section of the Cisco Service Module Configuration Guide.

Configure dynamic allocation of cores

Dynamic core allocations on the Cisco 8300 Series Secure Routers provide flexibility for users to leverage the CPU cores for different services and/or CEF/IPSec performances. The Cisco 8300 Series Secure Routers are equipped with a minimum of 16 CPU cores and have the flexibility to allocate cores into the service plane from the data plane. The core allocation is based on the customer configuration of the different services available on these platforms.

From Cisco IOS XE Release 17.15.3 onwards, you can use the **platform resource { service-plane-heavy | data-plane-heavy }** command to adjust the cores across service plane and data plane.

```
Router(config)# platform resource { service-plane-heavy | data-plane-heavy }
```

Following are the list of Cisco 8300 Series Secure Routers that support changing the core allocations dynamically:

- C8375-E-G2

Show command output for C8375-E-G2

This show command output shows the CPU cores allocaton for the data plane for C8375-E-G2:

Note By default, when a device boots up, the mode is service-plane-heavy.

```
Router# show platform software cpu alloc

CPU alloc information:
  Control plane cpu alloc: 0
  Data plane cpu alloc: 0-15
```

```
  Service plane cpu alloc: 0
  Slow control plane cpu alloc:
  Template used: CLI-data_plane_heavy
```



Note In the example, the maximum data plane core allocation is 15.

This show command output shows the CPU cores allocaton for the service plane for C8375-E-G2:

```
Router# show platform software cpu alloc
CPU alloc information:
  Control plane cpu alloc: 0
  Data plane cpu alloc: 0,6-15
  Service plane cpu alloc: 1-5
  Slow control plane cpu alloc:
  Template used: default-service_plane_heavy
```

The show command output shows the PPE status for C8375-E-G2:

```
Router# show platform hardware qfp active datapath infrastructure sw-cio

ID      Port  Wght  Global WRKR0  WRKR1  WRKR2  WRKR3  WRKR4  WRKR5  WRKR6  WRKR12  WRKR13
  WRKR14  Total
   1      rcl0     4:   6080     0      0      0      0      0      0      0       0
 0      64   6144
   1      rcl0     8:   6080     0      0      0      0      0      0      0       0
 0      64   6144
   2       ipc     1:      0     0      0      0      0      0      0      0       0
 0       0      0
   3 vxe_punti     1:    468     0      0      0      0      0      0      0       0
 0      44    512
   4      fpe0    LO:   1024     -      -      -      -      -      -      -       -
 -       -   1024
   4      fpe0    HI:   1024     -      -      -      -      -      -      -       -
 -       -   1024
   5      fpe1    LO:   1024     -      -      -      -      -      -      -       -
 -       -   1024
   5      fpe1    HI:   1024     -      -      -      -      -      -      -       -
 -       -   1024
   6      fpe2    LO:   1024     -      -      -      -      -      -      -       -
 -       -   1024
   6      fpe2    HI:   1024     -      -      -      -      -      -      -       -
 -       -   1024
   7      fpe3    LO:   1024     -      -      -      -      -      -      -       -
 -       -   1024
   7      fpe3    HI:   1024     -      -      -      -      -      -      -       -
 -       -   1024
   8      fpe4    LO:   1024     -      -      -      -      -      -      -       -
 -       -   1024
   8      fpe4    HI:   1024     -      -      -      -      -      -      -       -
 -       -   1024
   9      fpe5    LO:   1024     -      -      -      -      -      -      -       -
 -       -   1024
   9      fpe5    HI:   1024     -      -      -      -      -      -      -       -
 -       -   1024

Core Utilization over preceding 1147610.0731 seconds
-------------------------------------------------
      ID:       0       1       2       3       4       5       6      12      13      14
    % PP:    0.73    0.18    0.19    0.19    0.19    0.19    0.19    0.00    0.00    0.00
    % RX:    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.47
```

```
    % TM:    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.95    0.96    0.00
  % IDLE:   99.27   99.82   99.81   99.81   99.81   99.81   99.81   99.05   99.04   99.53
```

Show command output for C8355-G2

This show command output shows the CPU cores allocaton for the data plane for C8355-G2:

Note By default, when a device boots up, the mode is service-plane-heavy.

```
Router# show platform software cpu alloc

CPU alloc information:
Control plane cpu alloc: 0
Data plane cpu alloc: 0-11
Service plane cpu alloc: 0
Slow control plane cpu alloc:
Template used: CLI-data_plane_heavy
```

Note In the example, the maximum data plane core allocation is 11.

The show command output shows the PPE status for C8355-G2:

```
Router# show platform hardware qfp active datapath infrastructure sw-cio
Credits Usage:

                                                                                ID      Port  Wght  Global
WRKR0  WRKR1  WRKR2  WRKR3  WRKR4  WRKR5  WRKR6  WRKR7  WRKR8  WRKR9  WRKR10  Total

1      rcl0     4:   4544     0      0      0      0      0      0      0      0      0
  64       0   4608
1      rcl0     8:   4544     0      0      0      0      0      0      0      0      0
  64       0   4608
2       ipc     1:      0     0      0      0      0      0      0      0      0      0
   0       0      0
3 vxe_punti     1:    271    17     23     13     18     29     31     21     22     18
   0      49    512
4    l2_mod    LO:   1024     -      -      -      -      -      -      -      -      -
   -       -   1024
4    l2_mod    HI:   1024     -      -      -      -      -      -      -      -      -
   -       -   1024
5      fpe0    LO:   1024     -      -      -      -      -      -      -      -      -
   -       -   1024
5      fpe0    HI:   1024     -      -      -      -      -      -      -      -      -
   -       -   1024
6      fpe1    LO:   1024     -      -      -      -      -      -      -      -      -
   -       -   1024
6      fpe1    HI:   1024     -      -      -      -      -      -      -      -      -
   -       -   1024
7      fpe2    LO:   1024     -      -      -      -      -      -      -      -      -
   -       -   1024
7      fpe2    HI:   1024     -      -      -      -      -      -      -      -      -
   -       -   1024
8      fpe3    LO:   1024     -      -      -      -      -      -      -      -      -
   -       -   1024
8      fpe3    HI:   1024     -      -      -      -      -      -      -      -      -
   -       -   1024
9      fpe8    LO:   1019     -      -      -      -      -      -      -      -      -
```

```
   -       -   1024
9       fpe8    HI:   1024     -       -       -       -       -       -       -       -       -
   -       -   1024
10      fpe9    LO:   1019     -       -       -       -       -       -       -       -       -
    -       -   1024
10      fpe9    HI:   1024     -       -       -       -       -       -       -       -       -
    -       -   1024
Core Utilization over preceding 2793.4130 seconds

-------------------------------------------------

ID:        0       1       2       3       4       5       6       7       8       9
10
% PP:    64.19   64.51   64.10   64.61   64.08   64.13   64.20   64.57   64.42    0.00
0.00
% RX:     0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
0.00
% TM:     0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    1.89
49.31
% IDLE:   35.81   35.49   35.90   35.39   35.92   35.87   35.80   35.43   35.58   98.11
50.68
```

Enabling Cisco Discovery Protocol

Cisco Discovery Protocol (CDP) is enabled by default on the router.

For more information on using CDP, see Cisco Discovery Protocol Configuration Guide.

Configuring command-line access

To configure parameters to control access to the device, follow these steps.

Procedure

Step 1 **line** [| **console** | **tty** | **vty**] *line-number*

Example:

```
Router(config)# line console 0
```

Enters line configuration mode, and specifies the type of line.

The example provided specifies a console terminal for access.

Step 2 **password** *password*

Example:

```
Router(config-line)# password 5dr4Hepw3
```

Specifies a unique password for the console terminal line.

Step 3 **login**

Example:

```
Router(config-line)# login
```

Enables password checking at terminal session login.

Step 4 **exec-timeout** *minutes* [*seconds*]

Example:

```
Router(config-line)# exec-timeout 5 30
Router(config-line)#
```

Sets the interval during which the EXEC command interpreter waits until user input is detected. The default is 10 minutes. Optionally, adds seconds to the interval value.

The example provided here shows a timeout of 5 minutes and 30 seconds. Entering a timeout of **0 0** specifies never to time out.

Step 5 **exit**

Example:

```
Router(config-line)# exit
```

Exits line configuration mode to re-enter global configuration mode.

Step 6 **line** [| **console** | **tty** | **vty**] *line-number*

Example:

```
Router(config)# line vty 0 4
Router(config-line)#
```

Specifies a virtual terminal for remote console access.

Step 7 **password** *password*

Example:

```
Router(config-line)# password aldf2ad1
```

Specifies a unique password for the virtual terminal line.

Step 8 **login**

Example:

```
Router(config-line)# login
```

Enables password checking at the virtual terminal session login.

Step 9 **end**

Example:

```
Router(config-line)# end
```

Exits line configuration mode, and returns to privileged EXEC mode.

Example

These configurations show the command-line access commands.

You do not have to input the commands marked **default**. These commands appear automatically in the configuration file that is generated when you use the **show running-config** command.

```
!
line console 0
 exec-timeout 10 0
 password 4youreyesonly
 login
transport input none (default)
stopbits 1 (default)
line vty 0 4
 password secret
 login
!
```

Configuring static routes

Static routes provide fixed routing paths through the network. They are manually configured on the device. If the network topology changes, the static route must be updated with a new route. Static routes are private routes unless they are redistributed by a routing protocol.

To configure static routes, follow these steps.

Procedure

Step 1 (Option 1) **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]}

Example:

```
Router(config)# ip route 192.0.2.8 255.255.0.0 10.10.10.2
```

Specifies a static route for the IP packets. (If you are configuring an IPv6 address, use the **ipv6 route** command described below.)

Step 2 (Option 2) **ipv6 route** *prefix/mask* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]}

Example:

```
Router(config)# ipv6 route 2001:db8:2::/64 2001:DB8:3000:1
```

Specifies a static route for the IP packets.

Step 3 **end**

Example:

```
Router(config)# end
```

Exits global configuration mode and enters privileged EXEC mode.

Verifying Configuration

In this configuration example, the static route sends out all IP packets with a destination IP address of 192.0.2.8 and a subnet mask of 255.255.255.0 on the Gigabit Ethernet interface to another device with an IP address of 10.10.10.2. Specifically, the packets are sent to the configured interface.

You do not have to enter the command marked **default**. This command appears automatically in the configuration file generated when you use the **running-config** command.

```
!
ip classless (default)
ip route 192.0.2.8 255.255.255.0 10.10.10.2
```

To verify that you have configured static routing correctly, enter the **show ip route** command (or **show ipv6 route** command) and look for static routes marked with the letter S.

When you use an IPv4 address, you should see verification output similar to this example:

```
Router# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
       n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       H - NHRP, G - NHRP registered, g - NHRP registration summary
       o - ODR, P - periodic downloaded static route, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
       & - replicated local route overrides by connected

Gateway of last resort is 10.0.10.1 to network 192.0.2.6

S*    192.0.2.6/0 [254/0] via 10.0.10.1
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.0.10.0/24 is directly connected, GigabitEthernet0/0/0
L        10.0.10.13/32 is directly connected, GigabitEthernet0/0/0
C        10.108.1.0/24 is directly connected, Loopback0
L        10.108.1.1/32 is directly connected, Loopback0
```

When you use an IPv6 address, you should see verification output similar to this example:

```
Router# show ipv6 route
IPv6 Routing Table - default - 5 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE -
Destination
       NDr - Redirect, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       ls - LISP site, ld - LISP dyn-EID, a - Application

C   2001:DB8:3::/64 [0/0]
       via GigabitEthernet0/0/2, directly connected
```

```
S    2001:DB8:2::/64 [1/0]
       via 2001:DB8:3::1
```

Configuring dynamic routes

In dynamic routing, the network protocol adjusts the path automatically, based on network traffic or topology. Changes in dynamic routes are shared with other devices in the network.

A device can use IP routing protocols, such as Routing Information Protocol (RIP) or Enhanced Interior Gateway Routing Protocol (EIGRP), to learn about routes dynamically.

- Configuring Routing Information Protocol, on page 26
- Configuring Enhanced Interior Gateway Routing Protocol, on page 30

Configuring Routing Information Protocol

To configure the RIP on a router, follow these steps.

Procedure

Step 1 **router rip**

Example:

```
Router(config)# router rip
```

Enters router configuration mode, and enables RIP on the router.

Step 2 **version** {**1** | **2**}

Example:

```
Router(config-router)# version 2
```

Specifies use of RIP version 1 or 2.

Step 3 **network** *ip-address*

Example:

```
Router(config-router)# network 192.0.2.8
Router(config-router)# network 10.10.7.1
```

Specifies a list of networks on which RIP is to be applied, using the address of the network of each directly connected network.

Step 4 **no auto-summary**

Example:

```
Router(config-router)# no auto-summary
```

Disables automatic summarization of subnet routes into network-level routes. This allows subprefix routing information to pass across classful network boundaries.

Step 5 **end**

Example:

```
Router(config-router)# end
```

Exits router configuration mode, and enters privileged EXEC mode.

Example

Verifying Configuration

To see this configuration, use the **show running-config** command from privileged EXEC mode.

```
!
Router# show running-config
Building configuration...

Current configuration : 6504 bytes
!
! Last configuration change at 05:04:58 UTC Mon Jul 6 2020
!
version 17.3
service config
service timestamps debug datetime msec
service timestamps log datetime msec
service call-home
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
platform hardware throughput crypto 1G
!
hostname Router
!
boot-start-marker
boot system bootflash:c8000be-universalk9.17.03.01prd8.SPA.bin
boot-end-marker
!
!
!
no aaa new-model
!
login on-success log

!
subscriber templating
!
!
multilink bundle-name authenticated
no device-tracking logging theft


!
crypto pki trustpoint TP-self-signed-2347094934
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-2347094934
 revocation-check none
```

```
 rsakeypair TP-self-signed-2347094934
!
crypto pki trustpoint SLA-TrustPoint
 enrollment pkcs12
 revocation-check crl
!
!

crypto pki certificate chain SLA-TrustPoint
 certificate ca 01
  30820321 30820209 A0030201 02020101 300D0609 2A864886 F70D0101 0B050030
  32310E30 0C060355 040A1305 43697363 6F312030 1E060355 04031317 43697363
  6F204C69 63656E73 696E6720 526F6F74 20434130 1E170D31 33303533 30313934
  3834375A 170D3338 30353330 31393438 34375A30 32310E30 0C060355 040A1305
  43697363 6F312030 1E060355 04031317 43697363 6F204C69 63656E73 696E6720
  526F6F74 20434130 82012230 0D06092A 864886F7 0D010101 05000382 010F0030
  82010A02 82010100 A6BCBD96 131E05F7 145EA72C 2CD686E6 17222EA1 F1EFF64D
  CBB4C798 212AA147 C655D8D7 9471380D 8711441E 1AAF071A 9CAE6388 8A38E520
  1C394D78 462EF239 C659F715 B98C0A59 5BBB5CBD 0CFEBEA3 700A8BF7 D8F256EE
  4AA4E80D DB6FD1C9 60B1FD18 FFC69C96 6FA68957 A2617DE7 104FDC5F EA2956AC
  7390A3EB 2B5436AD C847A2C5 DAB553EB 69A9A535 58E9F3E3 C0BD23CF 58BD7188
  68E69491 20F320E7 948E71D7 AE3BCC84 F10684C7 4BC8E00F 539BA42B 42C68BB7
  C7479096 B4CB2D62 EA2F505D C7B062A4 6811D95B E8250FC4 5D5D5FB8 8F27D191
  C55F0D76 61F9A4CD 3D992327 A8BB03BD 4E6D7069 7CBADF8B DF5F4368 95135E44
  DFC7C6CF 04DD7FD1 02030100 01A34230 40300E06 03551D0F 0101FF04 04030201
  06300F06 03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604 1449DC85
  4B3D31E5 1B3E6A17 606AF333 3D3B4C73 E8300D06 092A8648 86F70D01 010B0500
  03820101 00507F24 D3932A66 86025D9F E838AE5C 6D4DF6B0 49631C78 240DA905
  604EDCDE FF4FED2B 77FC460E CD636FDB DD44681E 3A5673AB 9093D3B1 6C9E3D8B
  D98987BF E40CBD9E 1AECA0C2 2189BB5C 8FA85686 CD98B646 5575B146 8DFC66A8
  467A3DF4 4D565700 6ADF0F0D CF835015 3C04FF7C 21E878AC 11BA9CD2 55A9232C
  7CA7B7E6 C1AF74F6 152E99B7 B1FCF9BB E973DE7F 5BDDEB86 C71E3B49 1765308B
  5FB0DA06 B92AFE7F 494E8A9E 07B85737 F3A58BE1 1A48A229 C37C1E69 39F08678
  80DDCD16 D6BACECA EEBC7CF9 8428787B 35202CDC 60E4616A B623CDBD 230E3AFB
  418616A9 4093E049 4D10AB75 27E86F73 932E35B5 8862FDAE 0275156F 719BB2F0
  D697DF7F 28
        quit

!
!
license feature hseck9
license udi pid C8300-1N1S-6T sn FDO2320A0CF

diagnostic bootup level minimal
!
spanning-tree extend system-id
!
!
redundancy
 mode none

!
interface GigabitEthernet0/0/0
 ip dhcp client client-id ascii FDO2320A0CF
 ip address dhcp
 negotiation auto
!
interface GigabitEthernet0/0/1
 no ip address
 negotiation auto
!
!
ip http server
ip http authentication local
```

```
ip http secure-server
ip http client source-interface GigabitEthernet0/0/0
ip forward-protocol nd

!
!
!
control-plane
!
!
mgcp behavior rsip-range tgcp-only
mgcp behavior comedia-role none
mgcp behavior comedia-check-media-src disable
mgcp behavior comedia-sdp-force disable
!
mgcp profile default

!
!
dspfarm profile 7 conference security
 shutdown

!
line con 0
 exec-timeout 0 0
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
 transport input ssh
!
call-home
 ! If contact email address in call-home is configured as sch-smart-licensing@cisco.com
 ! the email address configured in Cisco Smart License Portal will be used as contact email
 address to send SCH notifications.
 contact-email-addr sch-smart-licensing@cisco.com
 profile "CiscoTAC-1"
  active
  destination transport-method http

!
!
end
```

To verify that you have configured RIP correctly, enter the **show ip route** command and look for RIP routes marked with the letter R. You should see an output similar to the one shown in this example:

```
Router# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.108.1.0 is directly connected, Loopback0
R    192.0.2.3/8 [120/1] via 192.0.2.2, 00:00:02, Ethernet0/0/0
```

Configuring Enhanced Interior Gateway Routing Protocol

To configure Enhanced Interior Gateway Routing Protocol (EIGRP), follow these steps.

Procedure

Step 1 **router eigrp** *as-number*

Example:

```
Router(config)# router eigrp 109
```

Enters router configuration mode, and enables EIGRP on the router. The autonomous-system number identifies the route to other EIGRP routers and is used to tag the EIGRP information.

Step 2 **network** *ip-address*

Example:

```
Router(config)# network 192.0.2.8
Router(config)# network 10.10.12.15
```

Specifies a list of networks on which EIGRP is to be applied, using the IP address of the network of directly connected networks.

Step 3 **end**

Example:

```
Router(config-router)# end
```

Exits router configuration mode, and enters privileged EXEC mode.

Verifying the Configuration

The following configuration example shows the EIGRP routing protocol enabled in IP networks 192.0.2.8 and 10.10.12.15. The EIGRP autonomous system number is 109. To see this configuration, use the **show running-config** command.

```
Router# show running-config
.
.
.
!
router eigrp 109
 network 192.0.2.8
  network 10.10.12.15
!
.
.
.
```

To verify that you have configured IP EIGRP correctly, enter the **show ip route** command, and look for EIGRP routes marked by the letter D. You should see verification output similar to this example:

```
Router# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.108.1.0 is directly connected, Loopback0
D     192.0.2.3/8 [90/409600] via 192.0.2.2, 00:00:02, Ethernet0/0
```

Enable Auto-Off Port LED

A global configuration CLI command is available to enable or disable the port LED control. By default, this feature is disabled. When auto-off is enabled, all port LEDs for front panel ports and module port will be set to OFF regardless of the link status.

- To enable the auto-off port LED, use this command in config mode:

  ```
  Router(config)# hw-module auto-off led
  ```

Note Only port LEDs will have an impact with this configuration. All the other LEDs will function normally.

- To disable the auto-off port LED, use this command in config mode:

  ```
  Router(config)# no hw-module auto-off led
  ```

Enable Blue Beacon LED

The blue beacon LED is a visual indicator on the device which is usually located at the front panel. This LED is designed to help network administrators to easily identify a specific device in environments where multiple devices are installed.

The beacon LED can be turned on by the administrator using a CLI command to indicate that the router needs attention. The Beacon LED can be enabled or disabled only by using the command on CLI.

- To turn on the Beacon LED use this command:

  ```
  Router#hw-module beacon R0 on
  ```

- To turn off the Beacon LED use this command:

  ```
  Router#hw-module beacon R0 off
  ```

- To check the status of the Beacon LED use this command:

  ```
  Router#hw-module beacon R0 status
  ```