# Basic Platform Configuration

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

# 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 Catalyst 8000 Series Edge Platforms look for a file named c8000.cfg. The device looks for this file before finding the standard files-router-confg or the ciscortr.cfg.

The device looks for the c8000.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, as shown in the following example:

```
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 TP-self-signed-2347094934
 certificate self-signed 01
  30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 32333437 30393439 3334301E 170D3230 30353238 32333331
  30325A17 0D333030 31303130 30303030 305A3031 312F302D 06035504 03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 33343730
  39343933 34308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201

  8B2FA1A7 29F5E8BD 57EB2459 CBBA7D64 4471BD34 0EC80AF2 0B693D0C 8DC3F771
  5D377065 57F16FD6 1B7AE4D3 3C5824B5 46FCDA97 4A5CA003 8B0BF2C9 E04A84E5
  E34E5EC6 AF94ACF3 DE5F9295 AA1C474F 30902D92 77F67A29 E4934212 DB9B253F
  1EC8F61F FD32D662 2F062666 13B8DC71 031F2119 551A487F 77E3BD46 3E5E7BBD
  9669BD8E FC4AEE6E EAD00DA5 DD56E370 716EC5CC 67DA7F35 6F4B3428 AD6EF6BD
  92868FAD 84871242 08C4FBED D5DB5249 336EB488 0D9A0B02 8BEE4BF9 5D03C416
  266E0F49 81030203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF
  301F0603 551D2304 18301680 14AE8751 EF7BF338 F7AB9FD8 E3EB151C F9E68DFA
  8A301D06 03551D0E 04160414 AE8751EF 7BF338F7 AB9FD8E3 EB151CF9 E68DFA8A
  300D0609 2A864886 F70D0101 05050003 82010100 925E6454 796E21F8 6401B0D1
  F2E09800 0B41752A B72F240E 21466633 1A2DAF8B 6F1C81B5 CE069EE0 F88888E4
  F6BAB34D 8328C2C7 781C4A6C FBB3DBCE 6F5C7100 388A6ADD 97D0E0CB 9407A5A3
  FF51FBD7 816E3D74 41769DAD C861B83B 68C58783 0A369849 32C27426 04513E09
  E3393274 201F3C44 D3EA63B2 EAB62240 B57200FE 3E3018C6 8013136A D9A51431
  DAB97350 17CEBF1F 2CFC553A 2C95A041 8426DABC AEFC27F7 B4A9F3F3 8C58C682
  2BDD7B4C 77F419A7 3F0B775B 8110B16F A67FEFE1 41EF7FE1 C9F0268B 943A9C62
  E367846A D2208BEF FE2562B3 FE96D8A9 2D2D4FB0 74C40850 914A0BDD 2B7C2C6E
  23F9BEB8 52A23129 4265A869 C2FA2BA5 039F4933
        quit
        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

!
!
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
!
interface GigabitEthernet0/0/2
 no ip address
 negotiation auto
!
interface GigabitEthernet0/0/3
 no ip address
 negotiation auto
!
interface GigabitEthernet0/0/4
 no ip address
 negotiation auto
!
interface GigabitEthernet0/0/5
 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
```

# 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

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Router> `**`enable`**<br>`Router# `**`configure terminal`**<br>`Router(config)#` | Enters global configuration mode when using the console port.<br>Use the following to connect to the device with a remote terminal:<br>`telnet router-name or address`<br>`Login: login-id`<br>`Password: *********`<br>`Router> enable` |
| **Step 2** | **hostname** *name*<br>**Example:**<br>`Router(config)# `**`hostname Router`** | Specifies the name for the device. |
| **Step 3** | **enable secret** *password*<br>**Example:**<br>`Router(config)# `**`enable secret cr1ny5ho`** | Specifies an encrypted password to prevent unauthorized access to the device. |
| **Step 4** | **no ip domain-lookup**<br>**Example:**<br>`Router(config)# `**`no ip domain-lookup`** | Disables the device from translating unfamiliar words (typos) into IP addresses.<br>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 gigabitethernet** *slot/bay/port*
2. **ip address** *ip-address mask*
3. **ipv6 address** *ipv6-address/prefix*
4. **no shutdown**
5. **exit**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **interface gigabitethernet** *slot/bay/port*<br>**Example:**<br>`Router(config)# interface gigabitethernet 0/0/1` | Enters the configuration mode for a Gigabit Ethernet interface on the device. |
| **Step 2** | **ip address** *ip-address mask*<br>**Example:**<br>`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*<br>**Example:**<br>`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**<br>**Example:**<br>`Router(config-if)# no shutdown` | Enables the Gigabit Ethernet interface and changes its state from administratively down to administratively up. |
| **Step 5** | **exit**<br>**Example:**<br>`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

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **interface** *type number*<br>**Example:**<br>`Router(config)# interface Loopback 0` | Enters configuration mode on the loopback interface. |
| **Step 2** | (Option 1) **ip address** *ip-address mask*<br>**Example:**<br>`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*<br>**Example:**<br>`Router(config-if)# 2001:db8::ffff:1/128` | Sets the IPv6 address and prefix on the loopback interface. |
| **Step 4** | **exit**<br>**Example:**<br>`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 the following 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 the following 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.

# Dynamic Allocation of Cores

Dynamic core allocations on the Catalyst 8000 Series Edge platforms provide flexibility for users to leverage the CPU cores for different services and/or CEF/IPSec performances. The Catalyst 8000 Series Edge platforms are equipped with a minimum of 8 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 17.4 onwards, you can use the **platform resource { service-plane-heavy | data-plane-heavy }** command to adjust the cores across service plane and data plane. However, you have to reboot the device for the configured profile to take effect.

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

From Cisco IOS XE Release 17.5.1 onwards, Catalyst 8000 Series Edge Platforms supports changing the core allocation dynamically. You do not have to reboot the devices to have the new allocation to take effect.

Following are the list of Catalyst 8000 Series Edge platforms that support changing the core allocations dynamically:

- C8300-2N1S-6T
- C8300-2N1S-4T2X
- C8300-2N2S-6T
- C8300-2N2S-4T2X
- C8200-1N-4T

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

The following show command output shows the CPU cores allocaiton for the data plane :

```
Router# show platform software cpu alloc

CPU alloc information:
 Control plane cpu alloc: 0
 Data plane cpu alloc: 1-7
 Service plane cpu alloc: 0
 Template used: CLI-data_plane_heavy
```

**Note** In the above example, the maximum data plane core allocation is 7.

The following show command output shows the CPU cores allocaiton for the service plane:

```
Router# show platform software cpu alloc

CPU alloc information:
  Control plane cpu alloc: 0
   Data plane cpu alloc: 4-7
  Service plane cpu alloc: 1-3
  Template used: CLI-service_plane_heavy
```

The following show command output shows the PPE status:

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

Credits Usage:

  ID      Port  Wght  Global WRKR0  WRKR1  Total
   1      rcl0     1:    474     0     38    512
   1       rcl0   128:    480     0     32    512
   2        ipc     1:    508     0      3    511
   3 vxe_punti     1:    474     0     38    512
   4       fpe0     1:    976     0     48   1024
   5       fpe1     1:    976     0     48   1024
   6       fpe2     1:    976     0     48   1024
   7       fpe3     1:    976     0     48   1024

Core Utilization over preceding 5475356.7738 seconds
----------------------------------------------------
      ID:       0       1
    % PP:    0.63    0.00
    % RX:    0.00    1.54
    % TM:    0.00    1.63
  % COFF:    0.00    0.69
  % IDLE:   99.37   96.15
```

# 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.

### SUMMARY STEPS

1. **line** [| **console** | **tty** | **vty**] *line-number*
2. **password** *password*
3. **login**
4. **exec-timeout** *minutes* [*seconds*]
5. **exit**
6. **line** [| **console** | **tty** | **vty**] *line-number*
7. **password** `password`
8. **login**
9. **end**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **line** [| **console** | **tty** | **vty**] *line-number*<br>**Example:**<br>`Router(config)# line console 0` | Enters line configuration mode, and specifies the type of line.<br>The example provided here specifies a console terminal for access. |
| **Step 2** | **password** *password*<br>**Example:** | Specifies a unique password for the console terminal line. |

| | Command or Action | Purpose |
|---|---|---|
| | `Router(config-line)# password 5dr4Hepw3` | |
| **Step 3** | **login**<br>**Example:**<br>`Router(config-line)# login` | Enables password checking at terminal session login. |
| **Step 4** | **exec-timeout** *minutes* [*seconds*]<br>**Example:**<br>`Router(config-line)# exec-timeout 5 30`<br>`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.<br>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**<br>**Example:**<br>`Router(config-line)# exit` | Exits line configuration mode to re-enter global configuration mode. |
| **Step 6** | **line** [\| **console** \| **tty** \| **vty**] *line-number*<br>**Example:**<br>`Router(config)# line vty 0 4`<br>`Router(config-line)#` | Specifies a virtual terminal for remote console access. |
| **Step 7** | **password** *password*<br>**Example:**<br>`Router(config-line)# password aldf2ad1` | Specifies a unique password for the virtual terminal line. |
| **Step 8** | **login**<br>**Example:**<br>`Router(config-line)# login` | Enables password checking at the virtual terminal session login. |
| **Step 9** | **end**<br>**Example:**<br>`Router(config-line)# end` | Exits line configuration mode, and returns to privileged EXEC mode. |

### Example

The following configuration shows 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.

### SUMMARY STEPS

1. (Option 1) **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]}
2. (Option 2) **ipv6 route** *prefix/mask* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]}
3. **end**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | (Option 1) **ip route** *prefix mask* {*ip-address* \| *interface-type interface-number* [*ip-address*]}<br>**Example:**<br>`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*]}<br>**Example:**<br>`Router(config)# ipv6 route 2001:db8:2::/64 2001:DB8:3000:1` | Specifies a static route for the IP packets. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **end**<br>**Example:**<br>`Router(config)# end` | Exits global configuration mode and enters privileged EXEC mode. |

### Verifying Configuration

In the following 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 the following:

```
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 the following:

```
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 14
- Configuring Enhanced Interior Gateway Routing Protocol, on page 18

## Configuring Routing Information Protocol

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

### SUMMARY STEPS

1. **router rip**
2. **version** {**1** | **2**}
3. **network** *ip-address*
4. **no auto-summary**
5. **end**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **router rip**<br>**Example:**<br>`Router(config)# router rip` | Enters router configuration mode, and enables RIP on the router. |
| **Step 2** | **version** {**1** \| **2**}<br>**Example:**<br>`Router(config-router)# version 2` | Specifies use of RIP version 1 or 2. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **network** *ip-address*<br>**Example:**<br>`Router(config-router)# network 192.0.2.8`<br>`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**<br>**Example:**<br>`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**<br>**Example:**<br>`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 the following 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.

### SUMMARY STEPS

1. **router eigrp** *as-number*
2. **network** *ip-address*
3. **end**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **router eigrp** *as-number*<br>**Example:**<br>`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*<br>**Example:**<br>`Router(config)# network 192.0.2.8`<br>`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**<br>**Example:**<br>`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 the following:

```
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
```