Basic router configuration

This chapter contains the following sections:

Default configuration

When you boot up the router for the first time, the router looks for a default file name-the PID of the router. For example, C8161-G2 looks for a file named **C8161-G2.cfg**. The Cisco 8100 Series Secure Routers looks for this file before finding the standard files-**router-confg** or the **ciscortr.cfg**.

The C8161-G2 looks for a file named **C8161-G2.cfg** file in the bootflash. If the file is not found in the bootflash, the router then looks for the standard files-**router-confg** and **ciscortr.cfg**. If none of the files are found, the router 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 : 6118 bytes
!
! Last configuration change at 18:09:54 UTC Tue Sep 9 2025
!
version 17.18
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
```

```
platform resource service-plane-heavy
!
hostname Router
!
boot-start-marker
boot-end-marker
!
!
aaa new-model
!
!
!
!
aaa session-id common
!
!
!
!
!
!
!
!
!
!
login on-success log
!
!
!
!
!
!
!
subscriber templating
!
!
!
!
!
!
!
!
!
!
!
!
!
!
product-analytics
!
!
crypto pki trustpoint TP-self-signed-303776382
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-303776382
 revocation-check none
 rsakeypair TP-self-signed-303776382
 hash sha512
!
crypto pki trustpoint SLA-TrustPoint
 enrollment pkcs12
 revocation-check crl
 hash sha512
!
!
crypto pki certificate chain TP-self-signed-303776382
 certificate self-signed 01
```

```
  3082032E 30820216 A0030201 02020101 300D0609 2A864886 F70D0101 0D050030
  30312E30 2C060355 04030C25 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 33303337 37363338 32301E17 0D323530 38323730 36303931
  315A170D 33353038 32373036 30393131 5A303031 2E302C06 03550403 0C25494F
  532D5365 6C662D53 69676E65 642D4365 72746966 69636174 652D3330 33373736
  33383230 82012230 0D06092A 864886F7 0D010101 05000382 010F0030 82010A02
  82010100 B2521F68 C09E24B1 89E40B24 853626B1 7F3F531D 6D02C649 66F1BD76
  8D5E402E 96D34B24 94E7FFE6 3CDFE83B C5FF2734 BCB5C95B 96A8470F F73A5DD4
  7F5CEF51 17BF69F9 61E8921D 4DB29641 DEA5DC94 DEEEF577 F8BC38AF 5EDA4DFD
  7BEC6B6F 22B387E9 228C26B8 24E6874F 15E37DDE 2DACAB5B CE9145A7 D927CC5F
  E406C5FB E0644A0A 5DD223AA D7BE44A3 9BECB90B 770B033E 31F3D7F3 818BF19A
  7249E78C F746D6B0 E2ECD2CC C6338E9D 67292CC0 2B4C0C5E 2FBE57A0 CCBBDF1B
  C0732BC7 55D55A5D AC2C8511 F9AEE8DE F36678A2 08B4693D 5325AB35 A67724F8
  CCC604BA C0D2BB14 E26CC9C4 50B9818E F311FE57 F397FD1A FCAE2041 A1B2DDEC
  79EB45C1 02030100 01A35330 51301D06 03551D0E 04160414 0AB72B54 4F5A1C91
  6B4D0922 B5EB5529 24638466 301F0603 551D2304 18301680 140AB72B 544F5A1C
  916B4D09 22B5EB55 29246384 66300F06 03551D13 0101FF04 05300301 01FF300D
  06092A86 4886F70D 01010D05 00038201 0100A9D5 BAE37659 4226FF9A 59835CAC
  9ECC9170 BCCC78AE EE48674A DFCF359C AD363065 61706435 50E96ACB 82B30090
  6A417C53 4E7E9000 77AAAC84 887A5006 E1DE278B 0F3B59DF 306A6240 7344AE5B
  C8B75372 EDEB27A4 E4497541 D67ECD79 97F5910A 17181502 CE1417BE 867C2151
  8CBE3380 8BE23C6A BC633AAB 252491A5 E3B40685 F5AE5AFE 3184884D AD0AEA0F
  BA2EC3D7 3C8BF748 84BFF882 99DA3471 11BE6758 29144FC9 18CAE5FB 2399743C
  30FC8AFC 84E61852 BAEA0CD7 14B13BC3 67D58D25 5408266B 2A442399 926169A0
  4ADBE01B F7F7F790 075B37D7 C2B9EDCF 3427C015 9401B552 3DE68D26 88B24C19
  FDF935A7 9CB0CD21 273FBF2C 77BC31CF 080F
        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 C8161-G2 sn FCW2832Y4YB
memory free low-watermark processor 63127
!
spanning-tree extend system-id
!
!
!
!
redundancy
 mode none
!
!
!
!
!
!
!
vlan internal allocation policy ascending
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
interface GigabitEthernet0/0/0
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet0/0/1
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet0/1/0
!
interface GigabitEthernet0/1/1
!
interface GigabitEthernet0/1/2
!
interface GigabitEthernet0/1/3
!
interface GigabitEthernet0/1/4
!
interface GigabitEthernet0/1/5
!
interface GigabitEthernet0/1/6
```

```
 switchport
!
interface GigabitEthernet0/1/7
 switchport
!
interface Vlan1
 no ip address
!
router rip
 version 2
 network 10.0.0.0
 network 192.168.1.0
 no auto-summary
!
ip forward-protocol nd
ip forward-protocol udp
ip http server
ip http authentication local
ip http secure-server
!
ip ssh bulk-mode 131072
!
!
!
!
!
!
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
 activation-character 13
 stopbits 1
line vty 0 4
 activation-character 13
 transport input ssh
line vty 5 14
 activation-character 13
 transport input ssh
!
!
!
!
!
!
!
```

Configuring global parameters

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`Router> enable`
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode when using the console port.
		Use the following to connect to the router with a remote terminal:
		`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 router.
Step 3	**enable password** *password* **Example:** `Router(config)# enable password cr1ny5ho`	Specifies a password to prevent unauthorized access to the router. **Note** In this form of the command, password is not encrypted.
Step 4	**no ip domain-lookup** **Example:** `Router(config)# no ip domain-lookup`	Disables the router 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** *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** *slot/bay/port* **Example:** `Router(config)# interface 0/0/1`	Enters the configuration mode for an interface on the router.
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 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 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 interface and changes its state from administratively down to administratively up.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits the configuration mode of interface and returns to the global configuration 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:** `Router(config)# interface Loopback 0`	Enters configuration mode on the loopback interface.
Step 2	(Option 1) **ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.10.1.1 255.255.255.0`	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.

The loopback interface in this sample configuration is used to support Network Address Translation (NAT) on the virtual-template interface. This configuration example shows the loopback interface configured on the Gigabit Ethernet interface with an IP address of 192.0.2.0/16, 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 192.0.2.1 255.255.0.0 (static IP address)
ip nat outside
!
interface Virtual-Template1
ip unnumbered loopback0
no ip directed-broadcast
ip nat outside
```

Verifying Loopback Interface Configuration

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 192.0.2.0/16
  MTU 1514 bytes, BW 8000000 Kbit, DLY 5000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation LOOPBACK, loopback not set
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/0, 0 drops; input queue 0/75, 0 drops
  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 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
     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 192.0.2.0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.0.2.0, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Configuring command-line access

To configure parameters to control access to the router, 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

Procedure

	Command or Action	Purpose
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 here 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.

	Command or Action	Purpose
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

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 router. 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

Procedure

	Command or Action	Purpose
Step 1	(Option 1) **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} **Example:** `Router(config)# ip route 192.0.2.1 255.255.0.0 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:db8:3::0`	Specifies a static route for the IP packets.
Step 3	**end** **Example:** `Router(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

In the following configuration example, the static route sends out all IP packets with a destination IP address of 192.168.1.0 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 PVC.

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 2001:db8:2::/64 2001:db8:3::0
```

Verifying Configuration

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: 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
S*   0.0.0.0/0 is directly connected, FastEthernet0
```

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 routers in the network.

A router 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

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

Procedure

	Command or Action	Purpose
Step 1	**router rip** **Example:**	Enters router configuration mode, and enables RIP on the router.

	Command or Action	Purpose
	`Router(config)# router rip`	
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 192.168.1.1`
	`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.

The following configuration example shows RIP Version 2 enabled in IP networks 10.0.0.0 and 192.168.1.0. To see this configuration, use the **show running-config** command from privileged EXEC mode.

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

Current configuration : 6118 bytes
!
! Last configuration change at 18:09:54 UTC Tue Sep 9 2025
!
version 17.18
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
platform resource service-plane-heavy
!
hostname Router
!
boot-start-marker
boot-end-marker
!
!
aaa new-model
!
!
!
!
```

```
aaa session-id common
!
!
!
!
!
!
!
!
!
!
login on-success log
!
!
!
!
!
!
!
subscriber templating
!
!
!
!
!
!
!
!
!
!
!
!
product-analytics
!
!
crypto pki trustpoint TP-self-signed-303776382
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-303776382
 revocation-check none
 rsakeypair TP-self-signed-303776382
 hash sha512
!
crypto pki trustpoint SLA-TrustPoint
 enrollment pkcs12
 revocation-check crl
 hash sha512
!
!
crypto pki certificate chain TP-self-signed-303776382
 certificate self-signed 01
  3082032E 30820216 A0030201 02020101 300D0609 2A864886 F70D0101 0D050030
  30312E30 2C060355 04030C25 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 33303337 37363338 32301E17 0D323530 38323730 36303931
  315A170D 33353038 32373036 30393131 5A303031 2E302C06 03550403 0C25494F
  532D5365 6C662D53 69676E65 642D4365 72746966 69636174 652D3330 33373736
  33383230 82012230 0D06092A 864886F7 0D010101 05000382 010F0030 82010A02
  82010100 B2521F68 C09E24B1 89E40B24 853626B1 7F3F531D 6D02C649 66F1BD76
  8D5E402E 96D34B24 94E7FFE6 3CDFE83B C5FF2734 BCB5C95B 96A8470F F73A5DD4
  7F5CEF51 17BF69F9 61E8921D 4DB29641 DEA5DC94 DEEEF577 F8BC38AF 5EDA4DFD
  7BEC6B6F 22B387E9 228C26B8 24E6874F 15E37DDE 2DACAB5B CE9145A7 D927CC5F
  E406C5FB E0644A0A 5DD223AA D7BE44A3 9BECB90B 770B033E 31F3D7F3 818BF19A
  7249E78C F746D6B0 E2ECD2CC C6338E9D 67292CC0 2B4C0C5E 2FBE57A0 CCBBDF1B
  C0732BC7 55D55A5D AC2C8511 F9AEE8DE F36678A2 08B4693D 5325AB35 A67724F8
```

```
  CCC604BA C0D2BB14 E26CC9C4 50B9818E F311FE57 F397FD1A FCAE2041 A1B2DDEC
  79EB45C1 02030100 01A35330 51301D06 03551D0E 04160414 0AB72B54 4F5A1C91
  6B4D0922 B5EB5529 24638466 301F0603 551D2304 18301680 140AB72B 544F5A1C
  916B4D09 22B5EB55 29246384 66300F06 03551D13 0101FF04 05300301 01FF300D
  06092A86 4886F70D 01010D05 00038201 0100A9D5 BAE37659 4226FF9A 59835CAC
  9ECC9170 BCCC78AE EE48674A DFCF359C AD363065 61706435 50E96ACB 82B30090
  6A417C53 4E7E9000 77AAAC84 887A5006 E1DE278B 0F3B59DF 306A6240 7344AE5B
  C8B75372 EDEB27A4 E4497541 D67ECD79 97F5910A 17181502 CE1417BE 867C2151
  8CBE3380 8BE23C6A BC633AAB 252491A5 E3B40685 F5AE5AFE 3184884D AD0AEA0F
  BA2EC3D7 3C8BF748 84BFF882 99DA3471 11BE6758 29144FC9 18CAE5FB 2399743C
  30FC8AFC 84E61852 BAEA0CD7 14B13BC3 67D58D25 5408266B 2A442399 926169A0
  4ADBE01B F7F7F790 075B37D7 C2B9EDCF 3427C015 9401B552 3DE68D26 88B24C19
  FDF935A7 9CB0CD21 273FBF2C 77BC31CF 080F
        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 C8161-G2 sn FCW2832Y4YB
memory free low-watermark processor 63127
!
spanning-tree extend system-id
!
!
!
!
redundancy
 mode none
```

```
!
!
!
!
!
!
!
vlan internal allocation policy ascending
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
interface GigabitEthernet0/0/0
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet0/0/1
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet0/1/0
!
interface GigabitEthernet0/1/1
!
interface GigabitEthernet0/1/2
!
interface GigabitEthernet0/1/3
!
interface GigabitEthernet0/1/4
!
interface GigabitEthernet0/1/5
!
interface GigabitEthernet0/1/6
 switchport
!
interface GigabitEthernet0/1/7
 switchport
!
interface Vlan1
 no ip address
!
router rip
 version 2
 network 10.0.0.0
 network 192.168.1.0
 no auto-summary
```

```
!
ip forward-protocol nd
ip forward-protocol udp
ip http server
ip http authentication local
ip http secure-server
!
ip ssh bulk-mode 131072
!
!
!
!
!
!
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
 activation-character 13
 stopbits 1
line vty 0 4
 activation-character 13
 transport input ssh
line vty 5 14
 activation-character 13
 transport input ssh
!
!
!
!
!
!
!
end

Router#
```

Verifying Configuration

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    192.0.2.2/8 [120/1] via 192.0.2.1, 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

Procedure

	Command or Action	Purpose
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 192.168.1.0`
	`Router(config)# network 10.10.12.115`
		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.

Example

The following configuration example shows the EIGRP routing protocol enabled in IP networks 192.168.1.0 and 10.10.12.115. 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 192.168.1.0
  network 10.10.12.115
!
.
.
.
```

Verifying Configuration

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:

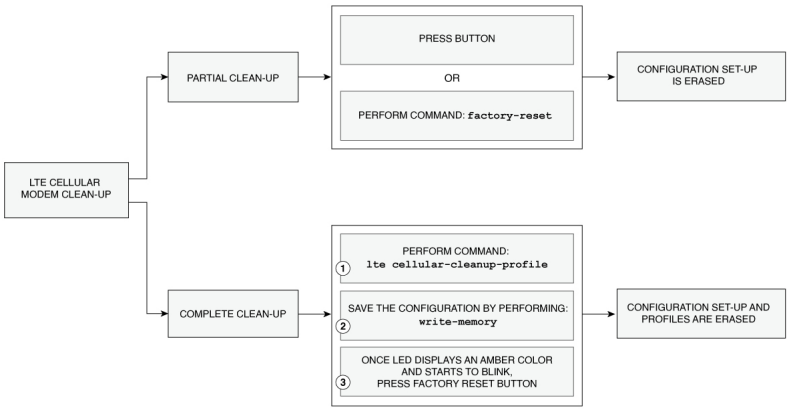
```
Router# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, 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     3.0.0.0/8 [90/409600] via 2.2.2.1, 00:00:02, Ethernet0/0
```

Erase configuration setup and cellular profiles on LTE modems

When using a cellular LTE modem, users have the option to perform a clean-up on the device. There are two types of clean-ups available for users: partial and complete.

A partial clean-up will remove the configuration set-up, while leaving user profiles intact. On the other hand, a complete clean-up will wipe the device of both configuration and profiles present in the modem.

It is up to the user to decide which clean-up option best suits their needs. The figure below shows the two types of clean-ups available for users:



Partial clean-up

The partial clean-up of an LTE cellular device involves removing the existing IOS XE configuration to ensure optimal clean-up of the device before it is repurposed.

There are two ways to enable the partial clean-up process: by pressing the factory reset button or by configuring the **factory-reset** command.

Prerequisites for erasing the configuration set-up

- Pressing the button: When the Router boots up, the LED displays an Amber color and starts to blink, take a pin or a toothpick and gently press on factory reset button for about 10 to 20 seconds.
- There are no pre-requisites before performing the **factory-reset** command.

Restrictions partial clean-up

- When using the partial clean-up method on a cellular LTE modem, only the configuration setup will be erased, leaving the profiles intact on the device.

Configuring partial cellular modem clean-up

Before you begin

Performing the **factory-reset** command is one of the ways to partially erase profiles on a cellular modem. Here are the steps:

SUMMARY STEPS

1. **configure terminal**
2. **factory-reset**
3. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`Router> enable`
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode.

Step 2	**factory-reset**
	Example:
	`Router#factory-reset ?`
	`all  All factory reset operations`
		Performs a partial clean-up of the cellular modem that erases the configuration setup.

	Command or Action	Purpose
	`keep-licensing-info Keep license usage info` `Router#`**`factory-reset`**	
Step 3	**exit** **Example:** `Router(config-if)#` **`exit`**	Exits the configuration mode.

The following configuration example shows partial clean-up of the cellular modem that erases the configuration set-up:

```
Router#factory-r
Router#factory-reset ?
  all                 All factory reset operations
  keep-licensing-info  Keep license usage info
```