Basic Router Configuration

This chapter contains the following sections:

ESR6300 Interface Naming

The supported hardware interfaces and their naming conventions are in the following table:

Hardware Interface	IOS-XE Naming Convention
Gigabit Ethernet combo port WAN/Layer3	gigabitEthernet 0/0/0 gigabitEthernet 0/0/1
Gigabit Ethernet LAN/Layer 2 ports	gigabitEthernet 0/1/0 gigabitEthernet 0/1/1 gigabitEthernet 0/1/2 gigabitEthernet 0/1/3
Console Port	Line console 0
USB Port	usbflash0: (IOS and rommon)

Basic Configuration

The basic configuration is a result of the entries you made during the initial configuration dialog. This means the router has at least one interface set with an IP address to be reachable, either through WebUI or to allow the PnP process to work. 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 : 13001 bytes
!
! Last configuration change at 12:17:45 UTC Wed Sep 25 2019
!
version 17.1
service timestamps debug datetime msec
service timestamps log datetime msec
service call-home
platform qfp utilization monitor load 80
platform hardware throughput level 2G
no platform punt-keepalive disable-kernel-core
!
hostname Router
!
boot-start-marker
boot system bootflash:c6300-universalk9.2019-09-23_04.57_gpollepa.SSA.bin
boot-end-marker
!
!
no logging monitor
!
aaa new-model
--More--
*Sep 25 23:53:59.524: %PARSER-5-CFGLOG_LOGGEDCMD: User:console logged command:!ex! e
!
aaa authentication enable default none
aaa authorization exec default local
!
aaa session-id common
clock timezone UTC -8 0
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
!
login on-success log
ipv6 unicast-routing
!
subscriber templating
!
multilink bundle-name authenticated
!
flow exporter 10.10.10.11
destination 10.10.10.11
!
crypto pki trustpoint SLA-TrustPoint
```

```
revocation-check crl
!
crypto pki trustpoint DNAC-CA
enrollment mode ra
enrollment terminal
usage ssl-client
revocation-check crl none
!
crypto pki trustpoint TP-self-signed-2633875772
enrollment selfsigned
subject-name cn=IOS-Self-Signed-Certificate-2633875772
revocation-check none
rsakeypair TP-self-signed-2633875772
!
!
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
crypto pki certificate chain DNAC-CA
crypto pki certificate chain TP-self-signed-2633875772
certificate self-signed 01
30820330 30820218 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
69666963 6174652D 32363333 38373537 3732301E 170D3139 30393235 30303034
35305A17 0D333030 31303130 30303030 305A3031 312F302D 06035504 03132649
4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 36333338
37353737 32308201 22300D06 092A8648 86F70D01 01010500 0382010F 00308201
0A028201 0100E848 4E2F3B31 0FA018F3 32D29A16 D8FDD1E1 14820D00 1D9A0AEF
994283C8 938803DA 3344AF4E BB4E94CD A8FDD1B8 AE35DC14 97BB8EC6 DC475C5F
438D80A9 6EC125BE 77246D7C 7D3F15DA CA340EDE 238595C5 CE70A762 315934B4
CA08037A 83E31EFB 5D070986 3C2553C0 C5B63A45 3A1FE935 EF0F74E2 2D5AC8C8
F21408A8 6AE6C799 D0F882D1 A2CA684D 29DA01A3 A31527F1 C613FF3F 547CEC2F
82ADFC5E 9461AA9A 546448AC BB3D82B6 449D3BC0 C844C92B 9D951423 7869250E
936E1147 AF8C5367 D34ECCA7 833AF8D1 CAC6F0CD 8A41BDD3 AE68CBD7 FA0794FF
D984763C 33E7F772 13A0E35B C9FA823E F6262FA1 269407B1 68BE318D EB3E13E2
2C32B3DF FC9D0203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF
301F0603 551D2304 18301680 14F5FA3B 95F59783 28BF5857 1EF8C95B CC6F758C
98301D06 03551D0E 04160414 F5FA3B95 F5978328 BF58571E F8C95BCC 6F758C98
```

```
300D0609 2A864886 F70D0101 05050003 82010100 31F5A7B6 DA464923 BFB45830
24FB2D90 D044B831 EDF45EB8 7671C54B 9F309AC9 1BE168EB C4AE8142 CE939520
4633531E 8DFFA22A A46373F5 9AEFFBDC A33DC696 C8606AEB 0771B923 E3731049
CE802EE9 825A93AD EA304F4B DE495E02 25142F57 4B50CBF7 08480E62 BF9587E4
0303DB1E 5B524CA9 0AAFF669 7319074B CE560D90 409F0976 60DE6299 12F360EC
36D50FB1 8471FF91 5C613CDE 836C1CD0 DB876E34 60BA6FF8 8DBE4F76 180F0A36
DAEB023C C3672AB6 F3CA2792 6585700A E9B3F173 08D02311 8FBECDD5 0B4D459E
5352D388 7414DED9 964AB488 68F6B3D7 0EB39A46 2351DBCD 8FD97D24 DFA4C9E1
02A312DF 31E49F1F F10F9D39 D9A7B12A 62F7875A
quit
!
no license feature hseck9
license udi pid ESR-6300-NCP-K9 sn FOC23032UUN
license boot level network-advantage addon dna-advantage
archive
log config
logging enable
logging size 1000
notify syslog contenttype plaintext
path bootflash:saved-config
maximum 5
memory free low-watermark processor 49965
!
diagnostic bootup level minimal
!
spanning-tree extend system-id
!
username cisco privilege 15 password 0 cisco
!
redundancy
mode none
!
vlan internal allocation policy ascending
!
lldp run
!
bridge irb
!
interface VirtualPortGroup0
ip address 192.168.0.1 255.255.255.0
ip nat inside
ipv6 address autoconfig
ipv6 enable
no mop enabled
no mop sysid
!
interface GigabitEthernet0/0/0
ip address 172.27.168.161 255.255.255.128
media-type rj45
negotiation auto
!
interface GigabitEthernet0/0/1
ip address 15.0.0.1 255.255.255.0
ip nat inside
shutdown
media-type sfp
negotiation auto
ipv6 enable
!
interface GigabitEthernet0/1/0
shutdown
!
interface GigabitEthernet0/1/1
shutdown
```

```
!
interface GigabitEthernet0/1/2
shutdown
!
interface GigabitEthernet0/1/3
shutdown
!
interface Vlan1
no ip address
!
interface vmi1
no ip address
!
iox (Note: Some protocols appear in the configuration although they are not supported)
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
ip route 0.0.0.0 0.0.0.0 GigabitEthernet0/0/0 172.27.168.129
ip ssh rsa keypair-name sshkeys
ip ssh version 2
!
snmp-server group mac v3 priv read mac write mac notify maca
snmp-server group fips v3 priv read fips write fips notify fips
snmp-server group cgnms v3 priv
snmp-server group cg-nms-administrator v3 priv
snmp-server view mac mib-2 included
snmp-server view mac system included
snmp-server view mac sysUpTime included
snmp-server view fips mib-2 included
snmp-server view fips system included
snmp-server view fips sysUpTime included
snmp-server community public RO R0
snmp-server community private RW
snmp-server enable traps snmp authentication linkdown linkup coldstart warmstart
snmp-server enable traps vrrp
snmp-server enable traps pfr
snmp-server enable traps flowmon
snmp-server enable traps ds1
snmp-server enable traps entity-perf throughput-notif
snmp-server enable traps ds3
snmp-server enable traps call-home message-send-fail server-fail
snmp-server enable traps tty
snmp-server enable traps eigrp
snmp-server enable traps ospf state-change
snmp-server enable traps ospf errors
snmp-server enable traps ospf retransmit
snmp-server enable traps ospf lsa
snmp-server enable traps ospf cisco-specific state-change nssa-trans-change
snmp-server enable traps ospf cisco-specific state-change shamlink interface
snmp-server enable traps ospf cisco-specific state-change shamlink neighbor
snmp-server enable traps ospf cisco-specific errors
snmp-server enable traps ospf cisco-specific retransmit
snmp-server enable traps ospf cisco-specific lsa
snmp-server enable traps casa
snmp-server enable traps xgcp
snmp-server enable traps license
snmp-server enable traps smart-license
snmp-server enable traps cef resource-failure peer-state-change peer-fib-state-change
inconsistency
snmp-server enable traps memory bufferpeak
snmp-server enable traps config-copy
snmp-server enable traps config
snmp-server enable traps config-ctid
```

```
snmp-server enable traps dhcp
snmp-server enable traps dsp card-status
snmp-server enable traps dsp oper-state
snmp-server enable traps dsp video-usage
snmp-server enable traps dsp video-out-of-resource
snmp-server enable traps fru-ctrl
snmp-server enable traps entity
snmp-server enable traps event-manager
snmp-server enable traps frame-relay multilink bundle-mismatch
snmp-server enable traps frame-relay
snmp-server enable traps frame-relay subif
snmp-server enable traps hsrp
snmp-server enable traps pimstdmib neighbor-loss invalid-register invalid-join-prune
rp-mapping-change interface-election
snmp-server enable traps ipmulticast
snmp-server enable traps isis
snmp-server enable traps ip local pool
snmp-server enable traps msdp
snmp-server enable traps mvpn
snmp-server enable traps ospfv3 state-change
snmp-server enable traps ospfv3 errors
snmp-server enable traps pim neighbor-change rp-mapping-change invalid-pim-message
snmp-server enable traps pppoe
snmp-server enable traps cpu threshold
snmp-server enable traps rsvp
snmp-server enable traps syslog
snmp-server enable traps l2tun session
snmp-server enable traps l2tun pseudowire status
snmp-server enable traps aaa_server
snmp-server enable traps atm subif
snmp-server enable traps pki
snmp-server enable traps ethernet evc status create delete
snmp-server enable traps ether-oam
snmp-server enable traps ethernet cfm cc mep-up mep-down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-missing mep-unknown service-up
snmp-server enable traps entity-state
snmp-server enable traps vdsl2line
snmp-server enable traps entity-sensor
snmp-server enable traps flash insertion removal lowspace
snmp-server enable traps srp
snmp-server enable traps nhrp nhs
snmp-server enable traps nhrp nhc
snmp-server enable traps nhrp nhp
snmp-server enable traps nhrp quota-exceeded
snmp-server enable traps otn
snmp-server enable traps pw vc
snmp-server enable traps ipsla
snmp-server enable traps sonet
snmp-server enable traps dlsw
snmp-server enable traps resource-policy
snmp-server enable traps lisp
snmp-server enable traps stpx inconsistency root-inconsistency loop-inconsistency
snmp-server enable traps dial
snmp-server enable traps sbc adj-status
snmp-server enable traps sbc blacklist
snmp-server enable traps sbc congestion-alarm
snmp-server enable traps c3g
snmp-server enable traps LTE
snmp-server enable traps vtp
snmp-server enable traps vlancreate
snmp-server enable traps ike policy add
snmp-server enable traps ike policy delete
snmp-server enable traps ipsec cryptomap add
snmp-server enable traps ipsec cryptomap delete
```

```
snmp-server enable traps ipsec cryptomap attach
snmp-server enable traps ipsec cryptomap detach
snmp-server enable traps ipsec tunnel start
snmp-server enable traps ipsec tunnel stop
snmp-server enable traps ipsec too-many-sas
snmp-server enable traps gdoi gm-start-registration
snmp-server enable traps gdoi gm-registration-complete
snmp-server enable traps gdoi gm-re-register
snmp-server enable traps gdoi gm-rekey-rcvd
snmp-server enable traps gdoi gm-rekey-fail
snmp-server enable traps gdoi ks-rekey-pushed
snmp-server enable traps gdoi gm-incomplete-cfg
snmp-server enable traps gdoi ks-no-rsa-keys
snmp-server enable traps gdoi ks-new-registration
snmp-server enable traps gdoi ks-reg-complete
snmp-server enable traps gdoi ks-role-change
snmp-server enable traps gdoi ks-gm-deleted
snmp-server enable traps gdoi ks-peer-reachable
snmp-server enable traps gdoi ks-peer-unreachable
snmp-server enable traps firewall serverstatus
snmp-server enable traps bulkstat collection transfer
snmp-server enable traps alarms informational
snmp-server enable traps vrfmib vrf-up vrf-down vnet-trunk-up vnet-trunk-down
!
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
stopbits 1
speed 115200
line vty 0 4
transport input all
transport output all
line vty 5 15
transport input all
transport output all
!
app-hosting system-resources
end
```

Configuring Global Parameters

To configure global parameters for your router, follow these 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 router 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 ESR6300`	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. To encrypt the password use enable secret password as noted in the previously mentioned Device Hardening Guide.

Serial Port Support

Additional protocol capabilities have been added to the ESR6300 to bring it into feature compatibility with the IR1101. These include:

- SCADA Gateway functionality (IEC10x and DNP3)
- Raw Socket (TCP and UDP)
- Line Relay
- Reverse Telnet

All of the configuration and show commands will be the same as are available on the IR1101 platform.

https://www.cisco.com/c/en/us/td/docs/routers/access/1101/software/configuration/guide/b_IR1101config.html

Configuring the Gigabit Ethernet Interface

The default configuration for the Gigabit Ethernet Interface (GI0/0/0 and GI0/0/1) on the ESR6300 are only Layer 3 (L3). The Gigabit Ethernet Interface on the ESR6300 is a combo port, which means it is a RJ45+SFP connector. These interfaces are combo ports, which means you can connect either through the RJ45 or SFP port by configuring the "media-type" to either auto-select, rj45 or sfp

To manually define the Gigabit Ethernet interface, follow these steps, beginning from global configuration mode.

Procedure

	Command or Action	Purpose
Step 1	**interface GigabitEthernet** *slot/bay/port* **Example:** `Router(config)# interface GigabitEthernet 0/0/0`	Enters the configuration mode for an interface on the router.
Step 2	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.168.12.2 255.255.255.0`	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. IPv6 unicast-routing needs to be set-up as well, see further information in the **IPv6 Addressing and Basic Connectivity Configuration Guide** located here: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipv6_basic/configuration/xe-16-10/ip6b-xe-16-10-book/read-me-first.html
Step 4	**ipv6 unicast-routing** **Example:** `Router (config)# ipv6 unicast-routing`	Enables forwarding of IPv6 unicast data packets.
Step 5	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface and changes its state from administratively down to administratively up.
Step 6	**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.

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 10.108.1.1 255.255.255.0`	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)# ipv6 address 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

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
```

Enabling Cisco Discovery Protocol

Cisco Discovery Protocol (CDP) is enabled by default on the router. It may be disabled if needed for security purposes.

For more information on using CDP, see Cisco Discovery Protocol Configuration Guide, Cisco IOS XE Release 3S.

Configuring Command-Line Access

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

Note Transport input must be set as explained in the previous Telnet and SSH sections of the guide.

Procedure

	Command or Action	Purpose
Step 1	**line** [**aux** | **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.

	Command or Action	Purpose

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** [**aux** | **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.

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.

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 192.10.2.3 255.255.0.0 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. See additional information for IPv6 here: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipv6_basic/configuration/xe-16-10/ip6b-xe-16-10-book/read-me-first.html
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.

All of the Cisco IOS-XE configuration guides can be found here: https://www.cisco.com/c/en/us/support/ios-nx-os-software/ios-xe-gibralter-16-10-1/model.html#ConfigurationGuides

Configuring Routing Information Protocol

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

Procedure

	Command or Action	Purpose
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 192.168.1.1`
	`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.

	Command or Action	Purpose
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 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    3.0.0.0/8 [120/1] via 2.2.2.1, 00:00:02, Ethernet0/0/0
```

Configuring Enhanced Interior Gateway Routing Protocol

The Enhanced Interior Gateway Routing Protocol (EIGRP) is an enhanced version of the Interior Gateway Routing Protocol (IGRP) developed by Cisco. The convergence properties and the operating efficiency of EIGRP have improved substantially over IGRP, and IGRP is now obsolete.

The convergence technology of EIGRP is based on an algorithm called the Diffusing Update Algorithm (DUAL). The algorithm guarantees loop-free operation at every instant throughout a route computation and allows all devices involved in a topology change to synchronize. Devices that are not affected by topology changes are not involved in recomputations

Details on configuring Enhanced Interior Gateway Routing Protocol (EIGRP), are found in the following guide: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_eigrp/configuration/xe-16-10/ire-xe-16-10-book/ire-enhanced-igrp.html

Configuring Open Shortest Path First (OSPF)

OSPF is an Interior Gateway Protocol (IGP) developed by the OSPF working group of the Internet Engineering Task Force (IETF). OSPF was designed expressly for IP networks and it supports IP subnetting and tagging of externally derived routing information. OSPF also allows packet authentication and uses IP multicast when sending and receiving packets.

Cisco supports RFC 1253, *OSPF Version 2 Management Information Base*, August 1991. The OSPF MIB defines an IP routing protocol that provides management information related to OSPF and is supported by Cisco routers.

Details on configuring OSPF are found in the following guide:https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_ospf/configuration/xe-16-12/iro-xe-16-12-book.html

Configuring Integrated Intermediate System-to-Intermediate System (IS-IS) Routing Protocol

IS-IS is a link-state Interior Gateway Protocol (IGP). Link-state protocols are characterized by the propagation of the information required to build a complete network connectivity map on each participating device. That map is then used to calculate the shortest path to destinations.

The IS-IS protocol was developed in the late 1980s by Digital Equipment Corporation (DEC) and was standardized by the International Standards Organization (ISO) in ISO/IEC 10589. The current version of this standard is ISO/IEC 10589:2002.

ISO/IEC 10589 defines support for the ISO Connectionless Network Protocol (CLNP) as defined in ISO 8473. However, the protocol was designed to be extensible to other network protocols. RFC 1195 defined IS-IS support for IP, and additional IETF extensions have defined IS-IS support for IPv6. Integration of support for multiple network layer protocols has led to the term Integrated IS-IS. The Cisco IOS IS-IS implementation supports CLNP, IPv4, and IPv6. This module and its related modules use the term IS-IS to refer to the Integrated IS-IS that is implemented by Cisco IOS software.

For details on configuring IS-IS, see the following guide:

https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_isis/configuration/xe-16-12/irs-xe-16-12-book.html

Modular QoS (MQC)

This section provides an overview of Modular QoS CLI (MQC), which is how all QoS features are configured on the IoT Integrated Services Router. MQC is a standardized approach to enabling QoS on Cisco routing and switching platforms.

Follow the procedures that are in the QoS Modular QoS Command-Line Interface Configuration Guide, Cisco IOS XE 17 guide.