

Basic platform configuration

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

- Default configuration, on page 1
- Configuring global parameters, on page 12
- Configuring Gigabit Ethernet interfaces, on page 12
- Configuring a loopback interface, on page 13
- Configuring module interfaces, on page 15
- Dynamic allocation of cores, on page 15
- Enabling Cisco Discovery Protocol, on page 17
- Configuring command-line access, on page 17
- Configuring static routes, on page 19
- Configuring dynamic routes, on page 20
- Enable Auto-Off Port LED, on page 33
- Enable Blue Beacon LED, on page 34

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 C8231-G2 Secure Routers look for a file named C8231-G2.cfg. The device looks for this file before finding the standard files-router-confg or the ciscortr.cfg.

The device looks for the C8231-G2.cfg file in the bootflash. If the file is not found in the bootflash, the device then looks for the standard files-router-confg and ciscortr.cfg. If none of the files are found, the device then checks for any inserted USB that may have stored these files in the same particular order.

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

Use the **show running-config** command to view the initial configuration, as shown in the following example:

```
Router#show running-config

Current configuration : 5480 bytes
```

```
!
! Last configuration change at 08:30:36 UTC Mon Aug 11 2025 by admin
!
version 17.18
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
service internal
platform qfp utilization monitor load 80
!
hostname arata-ethp3
!
boot-start-marker
boot-end-marker
!
!

vrf definition 1
!
address-family ipv4
exit-address-family
!
address-family ipv6
exit-address-family
!
vrf definition 65500
!
address-family ipv4
exit-address-family
!
vrf definition 65528
description SIG VRF
!
```

```
address-family ipv4

exit-address-family

!

vrf definition 65529

description Speedtest VRF

!


address-family ipv4

exit-address-family

!

vrf definition Mgmt-intf

!

address-family ipv4

exit-address-family

!

address-family ipv6

exit-address-family

!

no logging queue-limit

no logging rate-limit

aaa new-model

!

!
aaa authentication login default local

aaa authorization console

aaa authorization exec default
local
!

!

aaa session-id common

no process cpu extended history

!

ip name-server vrf Mgmt-intf 64.104.76.247

ip domain lookup vrf Mgmt-intf source-interface GigabitEthernet0
```

```
!
!
!
!
!
!
!
!
ip dhcp pool PnPWebUI1
vrf 65500
host 192.168.1.3 255.255.255.0
client-identifier 0077.6562.7569
dns-server 192.168.1.1
!
!
!
login on-success log
!
!
!
!
!
!
fhrp version vrrp v3
ipv6 unicast-routing
ipv6 rip vrf-mode enable
!
!
subscriber
templating
!
!
```

```
!
!
!
!
!
!
!
!
!
!
!
!
!
crypto pki trustpoint SLA-TrustPoint
enrollment pkcs12
revocation-check crl
hash sha512
!
crypto pki trustpoint TP-self-signed-363619999
enrollment selfsigned
revocation-check crl
rsakeypair TP-self-signed-363619999
hash sha512
!
!
crypto pki certificate chain SLA-TrustPoint
crypto pki certificate chain TP-self-signed-363619999
!
!
!
!
!
```

```
!

!

!

!

diagnostic bootup level minimal

!

license udi pid C8231-G2 sn FGL2903L28C

license accept end user agreement

license smart transport smart

memory free low-watermark processor 63122
!

spanning-tree extend system-id

!

!

!

username admin privilege 15 secret 9
$9$gIZ/Nyi9zyL6t.$wa0OV5z.ihFvtF9vDrUzGhYtb.T/mpAevEzdlxSe4rY
!

redundancy

mode none

!

!

!

!

!

no crypto ikev2 diagnose error

!

!

vlan internal allocation policy ascending

!

!

!

!
```

```
!
!


!
!
!
!
!
!
!
!
!
!
interface Loopback65528
vrf forwarding 65528
ip address 192.168.1.1 255.255.255.255
!
interface Loopback65529
vrf forwarding 65529
no ip address
!
interface GigabitEthernet0/0/0
!
interface GigabitEthernet0/0/1
!
interface GigabitEthernet0/0/2
!
interface GigabitEthernet0/0/3
!
interface TwoGigabitEthernet0/0/4
no ip address
negotiation auto
!
```

```
interface TwoGigabitEthernet0/0/5

no ip address
no ip address

negotiation auto

!

interface TwoGigabitEthernet0/0/6

no ip address

negotiation auto

!

interface TwoGigabitEthernet0/0/7

no ip address

negotiation auto

!

interface TenGigabitEthernet0/0/8

no ip address

ipv6 dhcp client request vendor

ipv6 address dhcp

ipv6 address autoconfig

ipv6 enable

!

interface TenGigabitEthernet0/0/8.28

encapsulation dot1Q 28

vrf forwarding 1

ip address 28.1.1.1 255.255.255.0

!

interface TenGigabitEthernet0/0/9

no ip address

ipv6 dhcp client request vendor

ipv6 address dhcp

ipv6 address autoconfig

ipv6 enable

ethernet oam
```

```
!
interface TenGigabitEthernet0/0/9.14
encapsulation dot1Q 14
vrf forwarding 1
ip address 14.1.1.2 255.255.255.0
!
interface TenGigabitEthernet0/0/9.17
encapsulation dot1Q 17
vrf forwarding 1
ip address 17.1.1.2 255.255.255.0
ipv6 address 3FFE:501:FFFF:100:E6A4:1CFF:FE82:F665/64
ipv6 enable
!
interface GigabitEthernet0
vrf forwarding Mgmt-intf
ip address 10.124.24.214 255.255.255.0
negotiation auto
ipv6 address autoconfig
ipv6 enable
!
interface Vlan1
vrf forwarding 65500
ip address 192.168.1.1 255.255.255.0
!
ip forward-protocol nd
no ip forward-protocol udp
ip tftp source-interface GigabitEthernet0
ip http server
ip http authentication local
ip http secure-server
ip http client source-interface GigabitEthernet0
!
ip nat settings central-policy
```

```
ip nat settings gatekeeper-size 1024

ip nat route vrf 65528 0.0.0.0 0.0.0.0 global

no ip nat service all-algs

ip route vrf Mgmt-intf 0.0.0.0 0.0.0.0 10.124.24.1

ip route vrf 1 36.1.1.0 255.255.255.0 14.1.1.1

ip route vrf 1 36.1.1.0 255.255.255.0 17.1.1.1

no ip ssh bulk-mode

ip scp server enable

!

no ipv6 mld ssm-map query dns

tftp-server bootflash:c8kg2be-universalk9.17.18.01a.SPA.bin

!

!

!

!

!

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

!

!

!

!

!

alias exec scp copy scp://tester:Login_999@10.75.28.59/images/ flash: vrf Mgmt-intf

!
```

```
line con 0

activation-character 13

stopbits 1

speed 115200

line aux 0

activation-character 13

line vty 0 4

privilege level 15

activation-character 13

transport input all

transport output all

line vty 5 15

privilege level 15

activation-character 13
transport input none

!

no network-clock revertive

ntp server vrf Mgmt-intf 10.64.58.51

!

!

!

!

!

!

!

netconf-yang

netconf-yang feature candidate-datastore

no netconf-yang ssh server algorithm encryption aes128-cbc

no netconf-yang ssh server algorithm encryption aes256-cbc

no netconf-yang ssh server algorithm hostkey ssh-rsa

no netconf-yang ssh server algorithm kex diffie-hellman-group14-sha1

no netconf-yang ssh server algorithm mac hmac-sha1
end
```

Configuring global parameters

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`Router> enable`
	`Router# configure terminal`
	`Router(config)#`
		Enters global configuration mode when using the console port.
		Use the following to connect to the device 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 device.
Step 3	**enable secret** *password* **Example:** `Router(config)# enable secret cr1ny5ho`	Specifies an encrypted password to prevent unauthorized access to the device.
Step 4	**no ip domain-lookup** **Example:** `Router(config)# no ip domain-lookup`	Disables the device from translating unfamiliar words (typos) into IP addresses. For complete information on global parameter commands, see the Cisco IOS Release Configuration Guide documentation set.

Configuring Gigabit Ethernet interfaces

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**interface TwoGigabitEthernet** *slot/bay/port* **Example:** `Router(config)# interface TwoGigabitEthernet 0/0/1`	Enters the configuration mode for a Gigabit Ethernet interface on the device.
Step 2	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 192.0.2.2 255.255.255.0`	Sets the IP address and subnet mask for the specified Gigabit Ethernet interface. Use this Step if you are configuring an IPv4 address.
Step 3	**ipv6 address** *ipv6-address/prefix* **Example:** `Router(config-if)# ipv6 address 2001.db8::ffff:1/128`	Sets the IPv6 address and prefix for the specified Gigabit Ethernet interface. Use this step instead of Step 2, if you are configuring an IPv6 address.
Step 4	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the Gigabit Ethernet interface and changes its state from administratively down to administratively up.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits configuration mode for the Gigabit Ethernet interface and returns to privileged EXEC mode.

Configuring a loopback interface

Before you begin

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

To configure a loopback interface, follow these steps.

SUMMARY STEPS

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

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**interface** *type number* **Example:** `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)# 2001:db8::ffff:1/128`	Sets the IPv6 address and prefix on the loopback interface.
Step 4	**exit** **Example:** `Router(config-if)# exit`	Exits configuration mode for the loopback interface and returns to global configuration mode.

Example

Verifying Loopback Interface Configuration

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

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

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

```
Router# show interface loopback 0
Loopback0 is up, line protocol is up
  Hardware is Loopback
  Internet address is 203.0.113.1/32
  MTU 1514 bytes, BW 8000000 Kbit/sec, DLY 5000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation LOOPBACK, loopback not set
  Keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     Output 0 broadcasts (0 IP multicasts)
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

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

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

Configuring module interfaces

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

Dynamic allocation of cores

Dynamic core allocations on the Cisco 8200 Series Secure Routers provide flexibility for users to leverage the CPU cores for different services and/or CEF/IPSec performances. The Cisco 8200 Series Secure Routers 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 17.18.1 onwards, you can use the **platform resource { service-plane-heavy | data-plane-heavy }** command to adjust the cores across service plane and data plane.

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

Following are the list of Cisco 8200 Series Secure Routers that support changing the core allocations dynamically:

- C8231-G2
- C8235-G2
- C8231-E-G2
- C8235-E-G2

Show command output

This show command output shows the CPU cores allocaton for the data plane for C8231-G2:

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

```
Router# show platform software cpu alloc
CPU alloc information:
Control plane cpu alloc: 0
Data plane cpu alloc: 0,4-7
Service plane cpu alloc: 1-3

Slow control plane cpu alloc:
Template used: default-service_plane_heavy
```

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

The show command output shows the PPE status for C8231-G2

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

Credits Usage:
ID      Port  Wght  Global WRKR0  WRKR1  WRKR2  WRKR6  Total
1      rcl0     4:   3008     0      0      0     64   3072
1      rcl0     8:   3008     0      0      0     64   3072
2       ipc     1:      0     0      0      0      0      0
3 vxe_punti     1:    406     9     27      9     61    512
4    l2_mod    LO:   1024     -      -      -      -   1024
4    l2_mod    HI:   1024     -      -      -      -   1024
5      fpe4    LO:   1024     -      -      -      -   1024
5      fpe4    HI:   1024     -      -      -      -   1024
6      fpe5    LO:   1024     -      -      -      -   1024
6      fpe5    HI:   1024     -      -      -      -   1024
7      fpe6    LO:   1024     -      -      -      -   1024
7      fpe6    HI:   1024     -      -      -      -   1024
8      fpe7    LO:   1024     -      -      -      -   1024
8      fpe7    HI:   1024     -      -      -      -   1024
9      fpe8    LO:    154     -      -      -      -   1024
9      fpe8    HI:   1024     -      -      -      -   1024
10      fpe9    LO:    151     -      -      -      -   1024
10      fpe9    HI:   1024     -      -      -      -   1024
Core Utilization over preceding 17934.6870 seconds
-------------------------------------------------
ID:        0       1       2       6
% PP:   99.33   99.40   99.34    0.00
% RX:    0.00    0.00    0.00    0.00
% TM:    0.00    0.00    0.00   26.94
% IDLE:    0.67    0.60    0.66   73.06
```

Enabling Cisco Discovery Protocol

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

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

Configuring command-line access

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

Procedure

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

Example:

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

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

The example provided specifies a console terminal for access.

Step 2 **password** *password*

Example:

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

Specifies a unique password for the console terminal line.

Step 3 **login**

Example:

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

Enables password checking at terminal session login.

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

Example:

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

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

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

Step 5 **exit**

Example:

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

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

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

Example:

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

Specifies a virtual terminal for remote console access.

Step 7 **password** *password*

Example:

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

Specifies a unique password for the virtual terminal line.

Step 8 **login**

Example:

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

Enables password checking at the virtual terminal session login.

Step 9 **end**

Example:

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

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

Example

These configurations show the command-line access commands.

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

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

Configuring static routes

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

To configure static routes, follow these steps.

Procedure

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

Example:

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

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

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

Example:

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

Specifies a static route for the IP packets.

Step 3 **end**

Example:

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

Exits global configuration mode and enters privileged EXEC mode.

Verifying Configuration

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

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

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

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

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

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

Gateway of last resort is 10.0.10.1 to network 192.0.2.6

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

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

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

C   2001:DB8:3::/64 [0/0]
       via GigabitEthernet0/0/2, directly connected
S   2001:DB8:2::/64 [1/0]
       via 2001:DB8:3::1
```

Configuring dynamic routes

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

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

- Configuring Routing Information Protocol, on page 20
- Configuring Enhanced Interior Gateway Routing Protocol, on page 32

Configuring Routing Information Protocol

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

Procedure

Step 1 **router rip**

Example:

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

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

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

Example:

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

Specifies use of RIP version 1 or 2.

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

Example:

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

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

Step 4 **no auto-summary**

Example:

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

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

Step 5 **end**

Example:

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

Exits router configuration mode, and enters privileged EXEC mode.

Example

Verifying Configuration

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

```
!
Router# show running-config
Current configuration : 5480 bytes
```

```
!
! Last configuration change at 08:30:36 UTC Mon Aug 11 2025 by admin
!
version 17.18
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
service internal
platform qfp utilization monitor load 80
!
hostname arata-ethp3
!
boot-start-marker
boot-end-marker
!
!

vrf definition 1
!
address-family ipv4
exit-address-family
!
address-family ipv6
exit-address-family
!
vrf definition 65500
!
address-family ipv4
exit-address-family
!
vrf definition 65528
description SIG VRF
!
```

```
address-family ipv4

exit-address-family

!

vrf definition 65529

description Speedtest VRF

!


address-family ipv4

exit-address-family

!

vrf definition Mgmt-intf

!

address-family ipv4

exit-address-family

!

address-family ipv6

exit-address-family

!

no logging queue-limit

no logging rate-limit

aaa new-model

!

!
aaa authentication login default local

aaa authorization console

aaa authorization exec default
local
!

!

aaa session-id common

no process cpu extended history

!

ip name-server vrf Mgmt-intf 64.104.76.247

ip domain lookup vrf Mgmt-intf source-interface GigabitEthernet0
```

```
!
!
!
!
!
!
!
!
ip dhcp pool PnPWebUI1
vrf 65500
host 192.168.1.3 255.255.255.0
client-identifier 0077.6562.7569
dns-server 192.168.1.1
!
!
!
login on-success log
!
!
!
!
!
!
fhrp version vrrp v3
ipv6 unicast-routing
ipv6 rip vrf-mode enable
!
!
subscriber
templating
!
```

```
!


!

!

!

!

!

!

!

!

!

!

!

!

crypto pki trustpoint SLA-TrustPoint
enrollment pkcs12

revocation-check crl

hash sha512

!

crypto pki trustpoint TP-self-signed-363619999

enrollment selfsigned

revocation-check crl

rsakeypair TP-self-signed-363619999

hash sha512

!

!

crypto pki certificate chain SLA-TrustPoint

crypto pki certificate chain TP-self-signed-363619999

!

!

!

!
```

```
!
!
!
!
!
diagnostic bootup level minimal
!
license udi pid C8231-G2 sn FGL2903L28C
license accept end user agreement
license smart transport smart
memory free low-watermark processor 63122
!
spanning-tree extend system-id
!
!
!
username admin privilege 15 secret 9
$9$gIZ/Nyi9zyL6t.$wa0OV5z.ihFvtF9vDrUzGhYtb.T/mpAevEzdlxSe4rY
!
redundancy
mode none
!
!
!
!
!
no crypto ikev2 diagnose error
!
!
vlan internal allocation policy ascending
!
!
!
!
```

```
!
!

!
!
!
!
!
!
!
!
!
!
interface Loopback65528
vrf forwarding 65528
ip address 192.168.1.1 255.255.255.255
!
interface Loopback65529
vrf forwarding 65529
no ip address
!
interface GigabitEthernet0/0/0
!
interface GigabitEthernet0/0/1
!
interface GigabitEthernet0/0/2
!
interface GigabitEthernet0/0/3
!
interface TwoGigabitEthernet0/0/4
no ip address
negotiation auto
```

```
!
interface TwoGigabitEthernet0/0/5
 no ip address
 no ip address
negotiation auto
!
interface TwoGigabitEthernet0/0/6
no ip address
negotiation auto
!
interface TwoGigabitEthernet0/0/7
no ip address
negotiation auto
!
interface TenGigabitEthernet0/0/8
no ip address
ipv6 dhcp client request vendor
ipv6 address dhcp
ipv6 address autoconfig
ipv6 enable
!
interface TenGigabitEthernet0/0/8.28
encapsulation dot1Q 28
vrf forwarding 1
ip address 28.1.1.1 255.255.255.0
!
interface TenGigabitEthernet0/0/9
no ip address
ipv6 dhcp client request vendor
ipv6 address dhcp
ipv6 address autoconfig
ipv6 enable
ethernet oam
```

```
!
interface TenGigabitEthernet0/0/9.14
encapsulation dot1Q 14
vrf forwarding 1
ip address 14.1.1.2 255.255.255.0
!
interface TenGigabitEthernet0/0/9.17
encapsulation dot1Q 17
vrf forwarding 1
ip address 17.1.1.2 255.255.255.0
ipv6 address 3FFE:501:FFFF:100:E6A4:1CFF:FE82:F665/64
ipv6 enable
!
interface GigabitEthernet0
vrf forwarding Mgmt-intf
ip address 10.124.24.214 255.255.255.0
negotiation auto
ipv6 address autoconfig
ipv6 enable
!
interface Vlan1
 vrf forwarding 65500
ip address 192.168.1.1 255.255.255.0
!
ip forward-protocol nd
 no ip forward-protocol udp
ip tftp source-interface GigabitEthernet0
ip http server
ip http authentication local
ip http secure-server
ip http client source-interface GigabitEthernet0
!
```

```
ip nat settings central-policy
ip nat settings gatekeeper-size 1024
ip nat route vrf 65528 0.0.0.0 0.0.0.0 global
no ip nat service all-algs
ip route vrf Mgmt-intf 0.0.0.0 0.0.0.0 10.124.24.1
ip route vrf 1 36.1.1.0 255.255.255.0 14.1.1.1
ip route vrf 1 36.1.1.0 255.255.255.0 17.1.1.1
no ip ssh bulk-mode
ip scp server enable
!
no ipv6 mld ssm-map query dns
tftp-server bootflash:c8kg2be-universalk9.17.18.01a.SPA.bin
!
 !
!
!
!
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
!
!
!
!
!
alias exec scp copy scp://tester:Login_999@10.75.28.59/images/ flash: vrf Mgmt-intf
!
```

```
line con 0

activation-character 13

stopbits 1

 speed 115200

line aux 0

activation-character 13

line vty 0 4

privilege level 15

activation-character 13

transport input all

transport output all

line vty 5 15

privilege level 15

activation-character 13
 transport input none

!

no network-clock revertive

ntp server vrf Mgmt-intf 10.64.58.51

!

!

!

!

!

!

!

netconf-yang

netconf-yang feature candidate-datastore

no netconf-yang ssh server algorithm encryption aes128-cbc

no netconf-yang ssh server algorithm encryption aes256-cbc

no netconf-yang ssh server algorithm hostkey ssh-rsa

no netconf-yang ssh server algorithm kex diffie-hellman-group14-sha1

no netconf-yang ssh server algorithm mac hmac-sha1
```

```
end
```

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

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

Gateway of last resort is not set

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

Configuring Enhanced Interior Gateway Routing Protocol

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

Procedure

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

Example:

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

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

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

Example:

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

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

Step 3 **end**

Example:

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

Exits router configuration mode, and enters privileged EXEC mode.

Verifying the Configuration

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

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

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

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

Gateway of last resort is not set

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

Enable Auto-Off Port LED

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

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

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

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

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

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

Enable Blue Beacon LED

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

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

- To turn on the Beacon LED use this command:

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

- To turn off the Beacon LED use this command:

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

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

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