



Configuring Communication Services

This chapter includes the following sections:

- [Configuring HTTP, page 1](#)
- [Configuring SSH, page 2](#)
- [Configuring XML API, page 3](#)
- [Configuring IPMI, page 4](#)
- [Configuring SNMP, page 5](#)

Configuring HTTP

Before You Begin

You must log in as a user with admin privileges to configure HTTP.

Procedure

	Command or Action	Purpose
Step 1	Server# scope http	Enters the HTTP command mode.
Step 2	Server /http # set enabled {yes no}	Enables or disables HTTP and HTTPS service on the CIMC.
Step 3	Server /http # set http-port number	Sets the port to use for HTTP communication. The default is 80.
Step 4	Server /http # set https-port number	Sets the port to use for HTTPS communication. The default is 443.
Step 5	Server /http # set http-redirect {yes no}	Enables or disables the redirection of an HTTP request to HTTPS.
Step 6	Server /http # set timeout seconds	Sets the number of seconds to wait between HTTP requests before the CIMC times out and terminates the session.

	Command or Action	Purpose
		Enter an integer between 60 and 10,800. The default is 1,800 seconds.
Step 7	Server /http # commit	Commits the transaction to the system configuration.

This example configures HTTP for the CIMC:

```
Server# scope http
Server /http # set enabled yes
Server /http *# set http-port 80
Server /http *# set https-port 443
Server /http *# set http-redirect yes
Server /http *# set timeout 1800
Server /http *# commit
Server /http # show
HTTP Port    HTTPS Port Timeout Active Sessions Enabled HTTP Redirected
-----
80           443         1800        0                yes      yes
Server /http #
```

Configuring SSH

Before You Begin

You must log in as a user with admin privileges to configure SSH.

Procedure

	Command or Action	Purpose
Step 1	Server# scope ssh	Enters the SSH command mode.
Step 2	Server /ssh # set enabled {yes no}	Enables or disables SSH on the CIMC.
Step 3	Server /ssh # set ssh-port <i>number</i>	Sets the port to use for secure shell access. The default is 22.
Step 4	Server /ssh # set timeout <i>seconds</i>	Sets the number of seconds to wait before the system considers an SSH request to have timed out. Enter an integer between 60 and 10,800. The default is 300 seconds.
Step 5	Server /ssh # commit	Commits the transaction to the system configuration.
Step 6	Server /ssh # show [detail]	(Optional) Displays the SSH configuration.

This example configures SSH for the CIMC:

```
Server# scope ssh
Server /ssh # set enabled yes
```

```

Server /ssh *# set ssh-port 22
Server /ssh *# set timeout 600
Server /ssh *# commit
Server /ssh # show
SSH Port      Timeout    Active Sessions Enabled
-----
22            600        1                yes

Server /ssh #

```

Configuring XML API

XML API for CIMC

The Cisco CIMC XML application programming interface (API) is a programmatic interface to CIMC for a C-Series Rack-Mount Server. The API accepts XML documents through HTTP or HTTPS.

For detailed information about the XML API, see *Cisco UCS Rack-Mount Servers CIMC XML API Programmer's Guide*.

Enabling XML API

Before You Begin

You must log in as a user with admin privileges to perform this task.

Procedure

	Command or Action	Purpose
Step 1	Server# scope xmlapi	Enters XML API command mode.
Step 2	Server /xmlapi # set enabled {yes no}	Enables or disables XML API control of CIMC.
Step 3	Server /xmlapi # commit	Commits the transaction to the system configuration.

This example enables XML API control of CIMC and commits the transaction:

```

Server# scope xmlapi
Server /xmlapi # set enabled yes
Server /xmlapi *# commit
Server /xmlapi # show detail
XMLAPI Settings:
  Enabled: yes
  Active Sessions: 0
  Max Sessions: 4

Server /xmlapi #

```

Configuring IPMI

IPMI Over LAN

Intelligent Platform Management Interface (IPMI) defines the protocols for interfacing with a service processor embedded in a server platform. This service processor is called a Baseboard Management Controller (BMC) and resides on the server motherboard. The BMC links to a main processor and other on-board elements using a simple serial bus.

During normal operations, IPMI lets a server operating system obtain information about system health and control system hardware. For example, IPMI enables the monitoring of sensors, such as temperature, fan speeds and voltages, for proactive problem detection. If server temperature rises above specified levels, the server operating system can direct the BMC to increase fan speed or reduce processor speed to address the problem.

Configuring IPMI over LAN

Configure IPMI over LAN when you want to manage the CIMC with IPMI messages.

Before You Begin

You must log in with admin privileges to perform this task.

Procedure

	Command or Action	Purpose
Step 1	Server# scope ipmi	Enters the IPMI command mode.
Step 2	Server /ipmi # set enabled {yes no}	Enables or disables IPMI access on this server.
Step 3	Server /ipmi # set privilege-level {readonly user admin}	Specifies the highest privilege level that can be assigned to an IPMI session on this server. This can be: <ul style="list-style-type: none"> • readonly — IPMI users can view information but cannot make any changes. If you select this option, IPMI users with the "Administrator", "Operator", or "User" user roles can only create read-only IPMI sessions, regardless of their other IPMI privileges. • user — IPMI users can perform some functions but cannot perform administrative tasks. If you select this option, IPMI users with the "Administrator" or "Operator" user role can create user and read-only sessions on this server. • admin — IPMI users can perform all available actions. If you select this option, IPMI users with the "Administrator" user role can create admin, user, and read-only sessions on this server.

	Command or Action	Purpose
Step 4	Server /ipmi # set encryption-key <i>key</i>	Sets the IPMI encryption key to use for IPMI communications. The key value must be 40 hexadecimal numbers.
Step 5	Server /ipmi # commit	Commits the transaction to the system configuration.

This example configures IPMI over LAN for the CIMC:

```
Server# scope ipmi
Server /ipmi # set enabled yes
Server /ipmi *# set privilege-level admin
Server /ipmi *# set encryption-key abcdef01234567890abcdef01234567890abcdef
Server /ipmi *# commit
Server /ipmi # show
Enabled Encryption Key                               Privilege Level Limit
-----
yes      abcdef01234567890abcdef01234567890abcdef admin
Server /ipmi #
```

Configuring SNMP

SNMP

The Cisco UCS C-Series Rack-Mount Servers support the Simple Network Management Protocol (SNMP) for viewing server configuration and status and for sending fault and alert information by SNMP traps. For information on Management Information Base (MIB) files supported by CIMC, see the *MIB Quick Reference for Cisco UCS* at this URL: http://www.cisco.com/en/US/docs/unified_computing/ucs/sw/mib/reference/UCS_MIBRef.html.

Configuring SNMP Properties

Before You Begin

You must log in as a user with admin privileges to perform this task.

Procedure

	Command or Action	Purpose
Step 1	Server# scope snmp	Enters SNMP command mode.
Step 2	Server /snmp # set enabled {yes no}	Enables or disables SNMP. Note SNMP must be enabled and saved before additional SNMP configuration commands are accepted.
Step 3	Server /snmp # commit	Commits the transaction to the system configuration.

	Command or Action	Purpose
Step 4	Server /snmp # set community-str <i>community</i>	Specifies the default SNMP v1 or v2c community name that CIMC includes on any trap messages it sends to the SNMP host. The name can be up to 18 characters.
Step 5	Server /snmp # set sys-contact <i>contact</i>	Specifies the system contact person responsible for the SNMP implementation. The contact information can be up to 254 characters, such as an email address or a name and telephone number. To enter a value that contains spaces, you must enclose the entry with quotation marks.
Step 6	Server /snmp # set sys-location <i>location</i>	Specifies the location of the host on which the SNMP agent (server) runs. The location information can be up to 254 characters. To enter a value that contains spaces, you must enclose the entry with quotation marks.
Step 7	Server /snmp # commit	Commits the transaction to the system configuration.

This example configures the SNMP properties and commits the transaction:

```

Server# scope snmp
Server /snmp # set enabled yes
Server /snmp *# commit
Server /snmp # set community-str cimcpublish
Server /snmp *# set sys-contact "User Name <username@example.com> +1-408-555-1212"
Server /snmp *# set sys-location "San Jose, California"
Server /snmp *# commit
Server /snmp # show detail
SNMP Settings:
  SNMP Port: 161
  System Contact: User Name <username@example.com> +1-408-555-1212
  System Location: San Jose, California
  SNMP Community: cimcpublish
  SNMP Trap community: 0
  Enabled: yes
  SNMP Trap Version: 1
  SNMP Inform Type: inform

Server /snmp #

```

What to Do Next

Configure SNMP trap settings as described in [Configuring SNMP Trap Settings](#), on page 6.

Configuring SNMP Trap Settings

Before You Begin

- You must log in with admin privileges to perform this task.
- SNMP must be enabled and saved before trap settings can be configured.

Procedure

	Command or Action	Purpose
Step 1	Server# scope snmp	Enters the SNMP command mode.
Step 2	Server /snmp # set trap-community-str <i>string</i>	Enter the name of the SNMP community to which trap information should be sent.
Step 3	Server /snmp # set trap-ver {1 2 3}	Specify the desired SNMP version of the trap message. Note SNMPv3 traps will be delivered only to locations where the SNMPv3 user and key values are configured correctly.
Step 4	Server /snmp # set inform-type {trap inform}	Specifies whether SNMP notification messages are sent as simple traps or as inform requests requiring acknowledgment by the receiver.
Step 5	Server /snmp # scope trap-destination <i>number</i>	Enters the SNMP trap destination command mode for the specified destination. Four SNMP trap destinations are available. The destination <i>number</i> is an integer between 1 and 4.
Step 6	Server /snmp/trap-destination # set enabled {yes no}	Enables or disables the SNMP trap destination.
Step 7	Server /snmp/trap-destination # set addr <i>ip-address</i>	Specifies the destination IP address to which SNMP trap information is sent.
Step 8	Server /snmp/trap-destination # commit	Commits the transaction to the system configuration.

This example configures general SNMP trap settings and trap destination number 1 and commits the transaction:

```

Server# scope snmp
Server /snmp # set trap-community-str public
Server /snmp # set trap-ver 3
Server /snmp # set inform-type inform
Server /snmp *# scope trap-destination 1
Server /snmp/trap-destination *# set enabled yes
Server /snmp/trap-destination *# set addr 192.0.20.41
Server /snmp/trap-destination *# commit
Server /snmp/trap-destination # show
Trap Destination IP Address      Enabled
-----
1                               192.0.20.41    yes
Server /snmp/trap-destination #

```

Sending a Test SNMP Trap Message

Before You Begin

You must log in with admin privileges to perform this task.

Procedure

	Command or Action	Purpose
Step 1	Server# scope snmp	Enters the SNMP command mode.
Step 2	Server /snmp # scope trap-destination <i>number</i>	Enters the SNMP trap destination command mode for the specified destination. Four SNMP trap destinations are available. The destination <i>number</i> is an integer between 1 and 4.
Step 3	Server /snmp/trap-destination # sendSNMPtrap	Sends an SNMPv1 test trap to the configured SNMP trap destination. Note The trap must be configured and enabled in order to send a test message.

This example sends a test message to SNMP trap destination 1:

```
Server# scope snmp
Server /snmp # scope trap-destination 1
Server /snmp/trap-destination # sendSNMPtrap
SNMP Test Trap sent to Destination:1
Server /snmp/trap-destination #
```

Configuring SNMPv3 Users

Before You Begin

- You must log in as a user with admin privileges to perform this task.
- SNMP must be enabled and saved before these configuration commands are accepted.

Procedure

	Command or Action	Purpose
Step 1	Server# scope snmp	Enters the SNMP command mode.
Step 2	Server /snmp # scope v3users <i>number</i>	Enters the SNMPv3 users command mode for the specified user number.
Step 3	Server /snmp/v3users # set v3add {yes no}	Adds or deletes an SNMPv3 user. This can be one of the following: <ul style="list-style-type: none"> • yes—This user is enabled as an SNMPv3 user and is allowed to access the SNMP OID tree. <p>Note The security name and security level must also be configured at this time or the user addition will fail.</p> <ul style="list-style-type: none"> • no—This user configuration is deleted.

	Command or Action	Purpose
Step 4	Server /snmp/v3users # set v3security-name <i>security-name</i>	Enter an SNMP username for this user.
Step 5	Server /snmp/v3users # set v3security-level { noauthnopriv authnopriv authpriv }	<p>Select a security level for this user. This can be one of the following:</p> <ul style="list-style-type: none"> • noauthnopriv—The user does not require an authorization or privacy password. • authnopriv—The user requires an authorization password but not a privacy password. If you select this option, you must configure an authentication key. • authpriv—The user requires both an authorization password and a privacy password. If you select this option, you must configure an authentication key and a private encryption key.
Step 6	Server /snmp/v3users # set v3proto { MD5 SHA }	Select an authentication protocol for this user.
Step 7	Server /snmp/v3users # set v3auth-key <i>auth-key</i>	Enter an authorization password for this user.
Step 8	Server /snmp/v3users # set v3priv-PROTO { DES AES }	Select an encryption protocol for this user.
Step 9	Server /snmp/v3users # set v3priv-auth-key <i>priv-auth-key</i>	Enter a private encryption key (privacy password) for this user.
Step 10	Server /snmp/v3users # commit	Commits the transaction to the system configuration.

This example configures SNMPv3 user number 2 and commits the transaction:

```

Server# scope snmp
Server /snmp # scope v3users 2
Server /snmp/v3users # set v3add yes
Server /snmp/v3users *# set v3security-name ucsSNMPV3user
Server /snmp/v3users *# set v3security-level authpriv
Server /snmp/v3users *# set v3proto SHA
Server /snmp/v3users *# set v3auth-key
Please enter v3auth-key:ex4mplek3y
Please confirm v3auth-key:ex4mplek3y
Server /snmp/v3users *# set v3priv-PROTO AES
Server /snmp/v3users *# set v3priv-auth-key
Please enter v3priv-auth-key:!1@2#3$4%5^6&7*8
Please confirm v3priv-auth-key:!1@2#3$4%5^6&7*8
Server /snmp/v3users *# commit
Settings are being applied ... allow a few minutes for the process to complete
Server /snmp/v3users # show detail
User 2:
  Add User: yes
  Security Name: ucsSNMPV3user
  Security Level: authpriv
  Auth Type: SHA
  Auth Key: *****

```

```
Encryption: AES
Private Key: *****
Server /snmp/v3users #
```