System Setup and Software Installation Guide for Cisco NCS 1014, IOS XR Releases 26.x.x

Cisco Network Convergence System 1014
Release: 26.x.x | Updated September 10, 2026

© 2026 Cisco Systems, Inc. All rights reserved.

Full Cisco Trademarks with Software License

Full Cisco Trademarks with Software License

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

Topics included

1 Bring up Cisco NCS 1014.....7
Boot behaviors of Cisco NCS 1014.....8
Zero Touch Provisioning.....9
How fresh boot using DHCP works.....10
ZTP configuration files.....12
ZTP bootscripts.....17
Manually initiate ZTP through CLI on Cisco NCS 1014.....18
Force ZTP through reload.....19
Log files for ZTP operations on Cisco NCS 1014.....21
Tech support information for ZTP on Cisco NCS 1014.....23
Configure a management interface.....23
Boot the Cisco NCS 1014 through the console port.....25
Boot Cisco NCS 1014 using a USB drive.....25
iPXE boot methods.....27
Configure the DHCP server.....28
Boot the Cisco NCS 1014 using iPXE.....30
Insert and verify operation of a Cisco NCS 1014 line card.....33
NTP servers.....36
Synchronize the clock with an NTP server.....37
Verify external reference clock status.....38
NTP troubleshooting information.....40

2 Perform Preliminary Checks.....41
Preliminary checks for Cisco NCS 1014.....42
Inventory support in Cisco NCS 1014.....42
Verify hardware components status.....48
Verify the software version.....50
Verify environmental parameters.....51
Sample outputs of the show environment command.....52
Verify management interface status.....59
Verify firmware version.....62
Display alarms on Cisco NCS 1014.....65
Verify core dump context information.....67
Verify core files.....68

3 Perform System Upgrade and Install Feature Packages.....69
Software and firmware compatibility matrix.....70
Upgrade the software on NCS 1014.....71

Software upgrade and downgrade matrix 74
Supported software upgrade and downgrade versions 75
FPD upgrades....78
FPD upgrade verification fields....84
Verify whether an FPD upgrade is required....85
Upgrade FPDs manually....92
Configure automatic FPD upgrade....93
Factory reset....95
Perform factory reset....96

4 Disaster Recovery....97
Disaster Recovery....98
CPU replacement considerations....98
Recovering nodes after chassis SSD replacement 98
Backup ISO image health checks....99

5 Connection Verification....101
Connection verification methods....102
Power data readings....102
How OLT line card connection verifications work....102
Verify connection for a CCMD-16 line card....103

6 System Health Check....107
System health checks....108
Enable the health check....109
Health check refresh cadence....110
View the status of all metrics....110
Metric threshold values....113
View health status of an individual metric....113
Disable health check....115

7 Configure AAA....117
AAA administrative models....118
Deprecation of Type 7 password and Type 5 secret....118
TACACS+ applications 124
Configure a TACACS+ server....124
Configure TACACS+ server groups....125
About RADIUS....126
Configure RADIUS server groups....126

8 Access Control Lists....129
Access control lists....130

Best practices for ACL workflows....131
How access control lists operate....131
Access control list application behavior....132
ACL configuration procedures....133
Configure an ingress IPv4 ACL on management Ethernet interface....133
Configure an egress IPv4 ACL on the management Ethernet interface....136
Configure an ingress IPv6 ACL on the management Ethernet interface....138
Configure an egress IPv6 ACL on the management Ethernet interface....140
Configure extended access lists....142
Modify ACLs....143

9 Smart Licensing....147
Smart Licensing....148
Smart licensing benefits for Cisco NCS 1014....150
Limitations of Smart Licesing....151
Create a token....151
Supported transport modes for Smart Licensing....152
Configure Call Home for Smart Licensing....152
Configure Smart Transport....153
Configure CSLU transport mode....155
Configure offline license mode....156
License entitlements and smart licensing features for NCS1K4-QXP-L-K9 and NCS1K14-OXP2-L-K9 line card....157
Smart licensing features for NCS1014-EDFA2 line card....158

10 Automated file management....161
Automated file management system....162
File types for automated file management....162
How automated file management works....162

11 Implementing audit monitoring....165
Audit logging....166
How audit logging works....166
Guidelines for audit logging....167
Notes about audit log storage....168
Configure audit logging....168
AuditD process keepalive....169
Guidelines for AuditD process keepalive....171
Restrictions for AuditD process keepalive....171
Configure the AuditD process keepalive....171

12 Process Memory Management....175
Core dump folders....176
Requirement: Maintain core dump folder disk space and usage thresholds....176

How core dump folder limit works.....177
Configure core dump folder limit.....179

1 Bring up Cisco NCS 1014

Topics:

- Boot behaviors of Cisco NCS 1014
- Zero Touch Provisioning
- Configure a management interface
- Boot the Cisco NCS 1014 through the console port
- Boot Cisco NCS 1014 using a USB drive
- iPXE boot methods
- Insert and verify operation of a Cisco NCS 1014 line card
- NTP servers

Provides comprehensive procedures for the initial bring-up and configuration of the Cisco NCS 1014. It covers system booting, ZTP, management interface setup, line card installation, and NTP synchronization to ensure efficient deployment and operation within service provider infrastructures.

Boot behaviors of Cisco NCS 1014

Describes how to boot Cisco NCS 1014 after the hardware is installed and how to log on to the XR console using the root credentials that you create during the first boot.

Boot behaviors of Cisco NCS 1014 are post-installation startup activities that

- power on the chassis and run the boot process from the pre-installed operating system image,
- use the iPXE boot or an external bootable USB drive when no valid pre-installed image is available, and
- establish the root username and password that you use to log on to the XR console.

Table 1: Feature History

Feature Name	Release Information	Feature Description
IPv6 support for protocols	Cisco IOS XR Release 25.1.1	IPv6 addressing is now supported for the protocols such as PXE, DHCP, SCP, HTTP, HTTPS, and NTP which are used to bring up the NCS1014 node. however, PXE does not support IPv6 when using HTTPS. Configuring IPv6 addresses on the management interfaces is supported, enabling communication between nodes to utilize the extensive address space. Additionally, IPv6 addressing ensures efficient and secure device management.

Boot behavior of Cisco NCS 1014

After installing the hardware, boot the Cisco NCS 1014 system. You can connect to the XR console port and power on the system. Cisco NCS 1014 completes the boot process using the pre-installed operating system (OS) image. If no image is available, Cisco NCS 1014 can be booted using the iPXE boot or an external bootable USB drive.

During the first boot, create the root username and password and then use these credentials to log on to the XR console.

Note

The output of the examples in the procedures is not from the latest software release. The output will change for any explicit references to the current release.

IPv6 support for bring-up protocols

From Release 25.1.1, IPv6 addressing is supported for the protocols used in bringing up the NCS1014 node. These protocols include PXE, DHCP, SCP, HTTP, HTTPS, and NTP.

> **Note**
>
> PXE does not support IPv6 when using HTTPS.

Configuring IPv6 addresses on the management interfaces is supported, enabling communication between nodes to utilize the extensive address space. Additionally, IPv6 addressing ensures efficient and secure device management.

Zero Touch Provisioning

Describes how Zero Touch Provisioning (ZTP) provisions Cisco NCS 1014 with day 0 configurations on management ports and data ports, the methods used to start ZTP, and the benefits of automated provisioning.

Zero Touch Provisioning (ZTP) is an automated provisioning capability on Cisco NCS 1014 that

- provisions the network device with day 0 configurations,
- supports both management ports and data ports, and
- downloads and applies a configuration file or runs a user script that the DHCP server points to.

What you can do with ZTP

ZTP provides multiple options:

- Automatically apply specific configuration in a large-scale environment.
- Download and install specific IOS XR image.
- Install specific application package or third-party applications automatically.
- Deploy containers without manual intervention.
- Upgrade or downgrade software versions effortlessly on thousands of network devices at a time.

Benefits of using ZTP

ZTP helps you manage large-scale service provider infrastructures effortlessly. Following are the added benefits of using ZTP:

- ZTP helps you to remotely provision a router anywhere in the network. This eliminates the need for an expert to deploy network devices and reduces IT cost.
- Automated provisioning using ZTP removes delay, increases accuracy, provides better customer experience and is cost-effective. By automating repeated tasks, ZTP allows network administrators to concentrate on more important stuff.
- ZTP process helps you to quickly restore service. Rather than troubleshooting an issue on hand, you can reset a system to a well-known working status.

Prerequisites for ZTP

ZTP does not execute, if a username is already configured in the system.

Methods to start ZTP

ZTP is initiated through one of the following methods:

- **Automated fresh boot**: On booting a device without any prior configuration, ZTP starts automatically. The router receives configuration file details from the DHCP server. This method is used for devices without pre-loaded configuration. Refer to Fresh boot using DHCP.
- **Configuration file**: The first line of the configuration file must be "!! IOS XR configuration" for ZTP to process it. If bringing up multiple new nodes, each requires a separate configuration file. Refer to ZTP configuration files on page 12.
- **ZTP bootscript**: Define a script to execute on every boot for customized automated actions. Refer to ZTP bootscript.
- **Manual invocation using CLI**: Use this method to forcefully start ZTP on a fully configured device by running a CLI command. Refer to Invoke ZTP manually through CLI.
- **Invocation using reload command**: Use the reload command to start ZTP on a fully configured device. Refer to Invoke ZTP through reload.

How fresh boot using DHCP works

Describes how the ZTP process initiates on a Cisco NCS 1014 device that has no prior configuration after you boot it with an IOS XR image, and the sequence in which DHCP requests, file downloads, and configuration application occur.

The fresh boot using DHCP starts the ZTP process on a Cisco NCS 1014 device that has no existing configuration. This process automates day 0 provisioning by coordinating DHCP-based discovery, automated file downloads, and applying the necessary configuration or scripts, enabling rapid deployment without manual intervention.

Summary

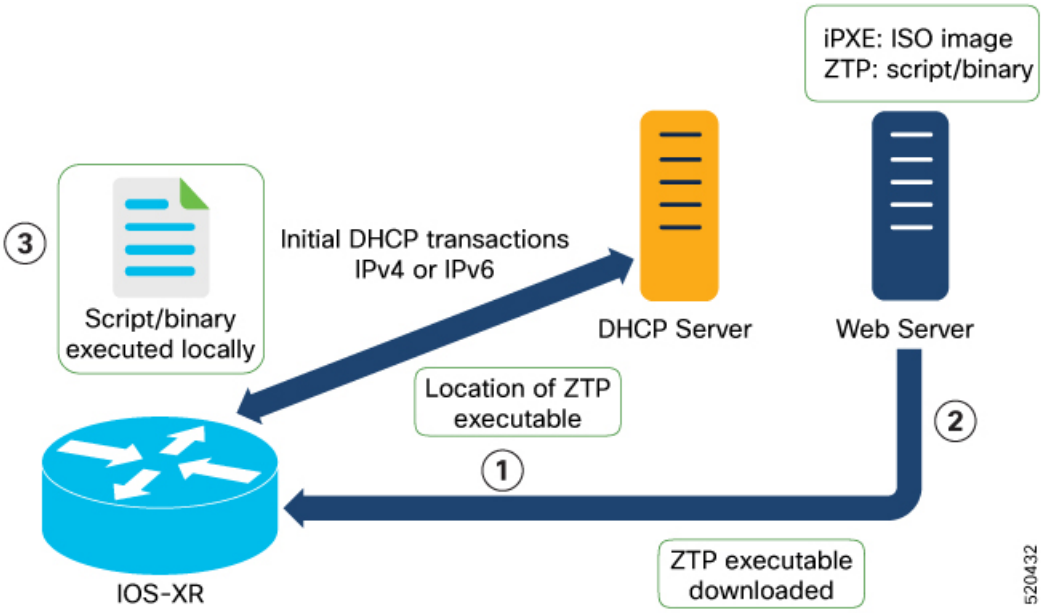
The key components involved in this process are:

- Cisco NCS 1014 device: Sends DHCP requests on management and data ports, downloads the configuration or script file, and applies the file or runs the script.
- DHCP server: Identifies the device through the DHCP options that ZTP sends and responds with the location of the configuration file or script.
- Web (HTTP) server: Hosts the configuration file or script file that the device downloads using the URL location provided in the DHCP response.

The figure depicts the high-level workflow of the ZTP process.

Workflow

Figure 1: ZTP fresh boot workflow



These stages describe the fresh boot using DHCP:

1. Z

   Initiating DHCP requests:

   ZTP begins by sending DHCP requests to obtain the configuration file or user script. To help the bootstrap server uniquely identify the device, these DHCP options are sent:

   - DHCP(v4/v6) client-id set to the device serial number
   - DHCPv4 option 124 providing the vendor, platform, and serial number
   - DHCPv6 option 16 providing the vendor, platform, and serial number

   DHCP requests are performed in this order:

   - ZTP sends an IPv4 DHCP request to all management ports. If this fails, it follows with an IPv6 request on the same ports.
   - Next, ZTP sends an IPv4 DHCP request on all data ports. If this fails, it follows with an IPv6 request on these ports.

2. DHCP server response: The DHCP server recognizes the device from the received options and replies with the location of the configuration or script file using one of these DHCP options:

   - DHCPv4 using BOOTP filename.
   - DHCPv4 using Option 67 (bootfile-name).
   - DHCPv6 using Option 59 (OPT_BOOTFILE_URL).

3. Downloading the file: The device uses the URL from the DHCP response to download the required file (configuration or script) from the web (HTTP) server.
4. Identifying the downloaded file: The device assesses the file type based on its content:

   - If the downloaded file content starts with **!! IOS XR**, it is considered as a configuration file.
   - If the downloaded file content starts with **#! /bin/bash**, **#! /bin/sh**, or **#!/usr/bin/python**, it is considered as a script file.

> **Note**

5. Applying the configuration or running the script: The device either applies the configuration file or runs the script or binary in the default bash shell.
6. Device is operational: The device is now up and running, fully provisioned with the day 0 configuration or script as directed by the DHCP server.

Result

The Cisco NCS 1014 device completes its initial provisioning and is ready for use, having automatically applied the appropriate configuration or script obtained through the ZTP and DHCP process.

ZTP configuration files

Describes the ZTP configuration file that initiates the ZTP process and provides a sample configuration file you can use as a starting point to automate Cisco NCS 1014 configuration.

A ZTP configuration file is an IOS XR configuration file that

- starts with **!! IOS XR** on the first line so that ZTP processes it as a configuration,
- contains the day 0 configuration that you want to automate on Cisco NCS 1014, and
- is downloaded from the DHCP server when ZTP is executed.

You can use either a configuration file or a script file to initiate the ZTP process, based on your business needs. The configuration file must start with **!! IOS XR** to be recognized by ZTP.

For more information on how to create ZTP configuration files, see ZTP Configuration Files Creation.

Sample ZTP configuration file

The following sample configuration file illustrates how you can automate all IOS XR configurations required during Zero Touch Provisioning:

```
!! Building configuration
!! IOS XR Configuration 7.11.1.35I
!! Last configuration change at Fri Sep 15 17:18:53 2023 by cisco
!
hostname IOS_P2B_FLT
logging console debugging
username cisco
 group root-lr
 group cisco-support
 secret 10
$6$4gjnzvvwDCz1z...$bovO.6uRYD9qsujiw6DNjTx6bngeDIVMvXxVbReal6bpd0SRo5qyfHk5S4D23r9hjntYtXnyQWNcrgbK0USB20
!
grpc
 port 57400
!
line template vty
 timestamp disable
 exec-timeout 0 0
!
line template test
 exec-timeout 0 0
```

```
!
line console
 timeout login response 30
 timestamp
 exec-timeout 0 0
 width 0
 length 0
!
line default
 timestamp disable
 exec-timeout 0 0
 length 0
 absolute-timeout 0
 session-timeout 0
!
vty-pool default 0 10 line-template default
fpd auto-upgrade enable
ntp
 max-associations 99
!
call-home
 service active
 contact smart-licensing
 profile CiscoTAC-1
  active
  destination transport-method email disable
  destination transport-method http
 !
!
netconf-yang agent
 ssh
!
hw-module location 0/1/NXR0
 mxponder-slice 0
  trunk-rate 600G
  client-rate 100GE
 !
!
hw-module location 0/2/NXR0
 mxponder-slice 0
  trunk-rate 800G
  client-port-rate 1 client-type 400GE
 !
!
interface MgmtEth0/RP0/CPU0/0
 description mgmt0
 ipv4 address 10.105.57.64 255.255.255.0
!
interface MgmtEth0/RP0/CPU0/1
 ipv4 address 10.127.60.44 255.255.255.0
 ipv6 enable
!
controller Optics0/0/0/0
 description optics0/0/0/0
 pm 30-sec optics threshold opt min 2
 fastpoll enable
 perf-mon enable
!
controller Optics0/0/0/1
 description optics0/0/0/1
 fastpoll enable
!
```

```
controller Optics0/0/0/2
 description optics0/0/0/2
 perf-mon enable
!
controller Optics0/0/0/3
 description optics0/0/0/3
!
controller Optics0/0/0/4
 description optics0/0/0/4
!
controller Optics0/0/0/5
 description optics0/0/0/5
!
controller Optics0/0/0/6
 description optics0/0/0/6
!
controller Optics0/0/0/7
 description optics0/0/0/7
!
controller Optics0/0/0/8
 description optics0/0/0/8
!
controller Optics0/0/0/9
 description optics0/0/0/9
 pm 15-min optics report opt max-tca enable
 pm 15-min optics threshold opt-dbm max -200
 pm 30-sec optics report opr min-tca enable
 pm 30-sec optics report opt max-tca enable
 pm 30-sec optics threshold opr-dbm min 500
 pm 30-sec optics threshold opt-dbm max -210
!
controller Optics0/0/0/10
 description optics0/0/0/10
!
controller Optics0/0/0/11
 description optics0/0/0/11
!
controller Optics0/0/0/12
 description optics0/0/0/12
!
controller Optics0/0/0/13
 description optics0/0/0/13
!
controller Optics0/1/0/0
 description optics0/1/0/0
 pm 15-min optics report opr min-tca enable
 pm 15-min optics threshold opr-dbm min 200
 pm 30-sec optics report opr min-tca enable
 pm 30-sec optics threshold opr-dbm min 200
 fastpoll enable
!
controller Optics0/1/0/1
 description optics0/1/0/1
!
controller Optics0/1/0/2
 description optics0/1/0/2
!
controller Optics0/1/0/3
 description optics0/1/0/3
!
controller Optics0/1/0/4
 description optics0/1/0/4
```

```
!
controller Optics0/1/0/5
 description optics0/1/0/5
!
controller Optics0/1/0/6
 description optics0/1/0/6
!
controller Optics0/1/0/7
 description optics0/1/0/7
!
controller Optics0/1/0/8
 description optics0/1/0/8
!
controller Optics0/1/0/9
 description optics0/1/0/9
!
controller Optics0/1/0/10
 description optics0/1/0/10
!
controller Optics0/1/0/11
 description optics0/1/0/11
!
controller Optics0/1/0/12
 description optics0/1/0/12
!
controller Optics0/1/0/13
 description optics0/1/0/13
!
controller Optics0/2/0/0
 description optics0/2/0/0
 transmit-power -25
 dwdm-carrier 100MHz-grid frequency 1923500
 rx-low-threshold -120
 rx-high-threshold 40
 tx-low-threshold -101
 tx-high-threshold 40
!
controller Optics0/2/0/1
 description optics0/2/0/1
!
controller Optics0/2/0/2
 description optics0/2/0/2
!
controller Optics0/2/0/3
 description optics0/2/0/3
!
controller Optics0/2/0/4
 description optics0/2/0/4
!
controller Optics0/2/0/5
 description optics0/2/0/5
!
controller Optics0/2/0/6
 description optics0/2/0/6
!
controller Optics0/2/0/7
 description optics0/2/0/7
!
controller Optics0/3/0/0
 description optics0/3/0/0
!
controller Optics0/3/0/1
```

```
 description optics0/3/0/1
!
controller Optics0/3/0/2
 description optics0/3/0/2
 pm 30-sec optics report opr min-tca enable
 pm 30-sec optics threshold opr-dbm min 200
!
controller Optics0/3/0/3
 description optics0/3/0/3
!
controller Optics0/3/0/4
 description optics0/3/0/4
!
controller Optics0/3/0/5
 description optics0/3/0/5
!
controller Optics0/3/0/6
 description optics0/3/0/6
!
controller Optics0/3/0/7
 description optics0/3/0/7
!
controller Optics0/3/0/8
 description optics0/3/0/8
!
controller Optics0/3/0/9
 description optics0/3/0/9
!
controller Optics0/3/0/10
 description optics0/3/0/10
!
controller Optics0/3/0/11
 description optics0/3/0/11
!
controller Optics0/3/0/12
 description optics0/3/0/12
!
controller Optics0/3/0/13
 description optics0/3/0/13
!
interface PTP0/RP0/CPU0/0
 shutdown
!
interface PTP0/RP0/CPU0/1
 shutdown
!
router static
 address-family ipv4 unicast
  0.0.0.0/0 10.105.57.1
  0.0.0.0/0 10.127.60.1
 !
!
snmp-server traps sensor
snmp-server traps fru-ctrl
netconf agent tty
!
lldp
!
ains-soak hours 47 minutes 59
ssh timeout 120
ssh server rate-limit 600
ssh server session-limit 100
```

```
ssh server v2
ssh server vrf default
ssh server netconf vrf default
end
```

ZTP bootscripts

Describes the Zero Touch Provisioning (ZTP) bootscript that runs on every boot of Cisco NCS 1014, the configuration commands that register the script, and sample scripts you can use as a starting point.

A ZTP bootscript is a downloadable script file that

- starts with **#! /bin/bash** or **#! /bin/sh** on the first line so that ZTP processes it as a script,
- provides a programmatic way to complete a task such as a patch upgrade using IOS XR commands, and
- runs every time the device boots when you register it through the **ztp bootscript** command.

You can either use the ZTP bash script or the ZTP configuration file. ZTP downloads and executes the script files. These script files include a programmatic approach to complete a task. For example, scripts created using IOS XR commands to perform patch upgrades.

Register the bootscript

To manually execute a script during every boot, use this configuration:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#ztp bootscript /disk0:/myscript
RP/0/RP0/CPU0:ios(config)#commit
```

This configuration waits for the first data plane interface to be configured and waits an extra minute for the management interface to receive an IP address, ensuring connectivity in the third-party namespace. If you do not want this delay, use:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#ztp bootscript preip /disk0:/myscript
RP/0/RP0/CPU0:ios(config)#commit
```

> **Note**
>
> When the above command is first configured, you are prompted whether you wish to invoke it now. The prompt helps with testing.

Sample content of /disk0:/myscript

```
host ncs1010_P1B_DT_08_ETH0 {
#hardware ethernet 68:9e:0b:b8:6f:5c ;
option dhcp-client-identifier "FCB2437B05N" ;
if exists user-class and option user-class = "iPXE" {
filename "http://192.0.2.51/P1B_DT_08/ncs1010-x64.iso";
} else {
filename "http://192.0.2.51/P1B_DT_08/startup.cfg";
}
fixed-address 192.0.2.19;
}
```

Sample ZTP bash script

```
#! /bin/bash
#
# NCS1014 Demo Sample
# ZTP installation of config and day-0 SMU's
#
source ztp_helper

wget http://downloads.sourceforge.net/project/yourcode/application.tgz
#install the downloaded application.tgz

#Run XR CLI's from the script
`xrcmd "show version"`
```

Sample ZTP configuration file

```
Tue May 4 18:08:59.544 UTC
Building configuration...
IOS XR Configuration 7.11.1.35I
!! Last configuration change at Fri Sep 15 17:18:53 2023 by cisco
!
line console
exec-timeout 0 0
!
line default
exec-timeout 0 0
session-timeout 0
!
vty-pool default 0 20
alias alarms show alarms brief system active
interface MgmtEth0/RP0/CPU0/0
ipv4 address dhcp
no shut
!
interface MgmtEth0/RP0/CPU0/1
description noshut-interface-ztp
ipv4 address 10.127.60.160 255.255.255.0
no shut
!
interface PTP0/RP0/CPU0/0
description noshut-interface-ztp
no shut
!
interface PTP0/RP0/CPU0/1
description noshut-interface-ztp
no shut
end
```

Manually initiate ZTP through CLI on Cisco NCS 1014

Configure ZTP by manually initiating, terminating, or cleaning ZTP sessions using CLI commands on a Cisco NCS 1014 device so you can provision the router in stages or test ZTP without a system reboot.

Configure ZTP by manually initiating, terminating, or cleaning ZTP sessions using CLI commands on a Cisco NCS 1014 device so you can provision the router in stages or test ZTP without a system reboot.

Use this task to provision the router in stages or test your ZTP configuration without needing to restart the device. You do not need to pre-configure or bring up interfaces beforehand; ZTP commands work on any data or management interface regardless of state and automatically bring up interfaces as needed.

Procedure

1. (Optional) Remove all the ZTP logs and saved settings using the command Initiate a new ZTP session.

   Example

```
RP/0/RP0/CPU0:ios#ztp clean
Fri Sep 15 17:12:33.477 IST
This would remove all ZTP temporary files.
Would you like to proceed? [no]: yes
All ZTP operation files have been removed.
ZTP logs are present in /var/log/ztp*.log for logrotate.
Please remove manually if needed.
If you now wish ZTP to run again from boot, do 'conf t/commit replace' followed
 by reload.
RP/0/RP0/CPU0:ios#
```

2. Reboot the Cisco NCS 1014 system to start a new ZTP session using the command **ztp initiate**.

   Example

```
RP/0/RP0/CPU0:ios#ztp initiate
Fri Sep 15 17:13:28.580 IST
Inititaing ZTP may change your configuration.
Interfaces might be brought up if they are in shutdown state
Would you like to proceed? [no]: yes
ZTP will now run in the background.
Please use "show logging" or look at /var/log/ztp.log to check progress.
RP/0/RP0/CPU0:ios#
```

3. (Optional) Terminate the active ZTP session using the command **ztp terminate**.

   Example

```
RP/0/RP0/CPU0:ios#ztp terminate
Fri Sep 15 17:15:04.592 IST
This would terminate active ZTP session if any (this may leave your system
in a partially configured state)
Would you like to proceed? [no]: yes
Terminating ZTP
RP/0/RP0/CPU0:ios#
```

ZTP runs on the specified interfaces of Cisco NCS 1014, and the device is provisioned with the day 0 configuration or script that the DHCP server pointed to.

Force ZTP through reload

Configure ZTP on a fully provisioned Cisco NCS 1014 by removing all custom configuration and ZTP state, then reloading the device to trigger ZTP.

Restore the device to its ZTP-initiated state by wiping the running configuration, cleaning ZTP logs and saved settings, and rebooting the system.

Use this task to automatically invoke the ZTP process on a Cisco NCS 1014 device by performing a reload. This operation is valuable when you need to re-provision a device or return it to its default, fully provisioned state.

Procedure

1. Enter the configuration mode.

   configure

   Example

   ```
   RP/0/RP0/CPU0:P2B_DT_02#configure
   ```

2. Remove the entire running configuration.

   commit replace

   Example

   ```
   RP/0/RP0/CPU0:ios(config)#commit replace
   Fri Sep 15 11:47:31.746 UTC

   This commit will replace or remove the entire running configuration. This
   operation can be service affecting.
   Do you wish to proceed? [no]: yes
   RP/0/RP0/CPU0:ios(config)#
   ```

3. Remove all the ZTP logs and saved settings.

   ztp clean

   Example

   ```
   RP/0/RP0/CPU0:ios#ztp clean
   Fri Sep 15 11:48:13.669 UTC
   This would remove all ZTP temporary files.
   Would you like to proceed? [no]: yes
   All ZTP operation files have been removed.
   ZTP logs are present in /var/log/ztp*.log for logrotate.
   Please remove manually if needed.
   If you now wish ZTP to run again from boot, do 'conf t/commit replace' followed
    by reload.
   RP/0/RP0/CPU0:ios#
   ```

4. Reboot the Cisco NCS 1014 system.

   reload

   After the node comes up, you can see that the ZTP is initiated and the configuration is restored successfully.

   Example

   ```
   RP/0/RP0/CPU0:ios#reload
   Fri Apr 29 06:50:12.312 UTC
   Proceed with reload? [confirm]

   RP/0/RP0/CPU0:ios#
   Preparing system for backup. This may take a few minutes especially for large
    configurations.
    Status report: node0_RP0_CPU0: BACKUP INPROGRESS
   ```

```
 Status report: node0_RP0_CPU0: BACKUP HAS COMPLETED SUCCESSFULLY
[Done]
```

```
RP/0/RP0/CPU0:Sep 25 17:29:19.875 UTC: pyztp2[325]: %INFRA-ZTP-4-START : ZTP
 has started. Interfaces might be brought up if they are shutdown
RP/0/RP0/CPU0:Sep 25 17:30:33.286 UTC: pyztp2[325]:
%INFRA-ZTP-6-DISCOVERY_COMPLETED : Discovery successful on MgmtDhcp4Fetcher.
 Will proceed with fetching.
RP/0/RP0/CPU0:Sep 25 17:30:47.362 UTC: pyztp2[325]:
%INFRA-ZTP-6-FETCHING_COMPLETED : Provisioning file fetched successfully
RP/0/RP0/CPU0:Sep 25 17:31:30.889 UTC: pyztp2[325]:
%INFRA-ZTP-4-CONFIG_INITIATED : ZTP has initiated config load and commit
operations
RP/0/RP0/CPU0:Sep 25 17:32:36.100 UTC: pyztp2[325]:
%INFRA-ZTP-4-CONFIG_FINISHED : ZTP has finished config load and commit
operations
RP/0/RP0/CPU0:Sep 25 17:32:41.059 UTC: pyztp2[325]:
%INFRA-ZTP-6-CFG_TAMP_SAVE_HASH : Config hash saved after ztp Config is:
(643013d9a43a3d2576012a24eb9745a8f960480d0053d06ed81146cb3c3d54c5).
RP/0/RP0/CPU0:Sep 25 17:32:44.089 UTC: pyztp2[325]:
%INFRA-ZTP-4-PROVISIONING_COMPLETED : ZTP has successfully completed the
provisioning
RP/0/RP0/CPU0:Sep 25 17:32:52.909 UTC: pyztp2[325]: %INFRA-ZTP-4-EXITED : ZTP
 exited
User Access Verification

Username: cisco
Password:
ios con0/RP0/CPU0 is now available
```

The ZTP process is initiated on Cisco NCS 1014, the configuration is restored, and the device returns to a fully provisioned state.

Log files for ZTP operations on Cisco NCS 1014

Lists the location where the ZTP process records its operation on Cisco NCS 1014 and shows a sample log file produced when ZTP downloads and runs a script from a data interface.

ZTP records all provisioning activities on the flash file system in the `/disk0:/ztp/` directory. The log captures every transaction with the DHCP server and tracks all state transitions throughout the provisioning process.

This is a sample excerpt from a ZTP log obtained during the execution of a simple configuration script that was downloaded from a data interface. The script unshuts all system interfaces and sets a load interval of 30 seconds on each interface.

```
2023-09-25 17:37:31,693  28136 [Engine       ] DEB: ZAdmin, current
state:active. Processing work: Sending standby sync message. done = False
2023-09-25 17:37:31,716  28136 [Engine       ] DEB: ZAdmin, current
state:active. Processing work: [privileged] getting engine status. done = False
2023-09-25 17:37:31,717  28136 [Engine       ] DEB: ZAdmin, current
state:active. Processing work: Fetching provisioning data. done = False
2023-09-25 17:37:31,718  28136 [Engine       ] INF: ZAdmin, current
state:active: state tag changed to fetch
2023-09-25 17:37:31,721  28136 [Xr           ] INF: Downloading the file to
/tmp/ztp.script
2023-09-25 17:37:31,948  28136 [ReportBootz  ] INF: User script downloaded
successfully. Provisioning in progress.
2023-09-25 17:37:31,950  28136 [Engine       ] DEB: ZAdmin, current
```

```
state:active. Processing work: Config device work for ZAdmin. done = False
2023-09-25 17:37:31,951  28136 [ZtpHelpers   ] DEB: Executing: source
/pkg/bin/ztp_helper.sh && echo -ne  | xrcmd "show version"
2023-09-25 17:37:32,956  28136 [ZAdmin       ] DEB: Proceeding to provision
the router
2023-09-25 17:37:32,958  28136 [Engine       ] DEB: ZAdmin, current
state:active. Processing work: ZAdmin: Apply configuration. done = False
2023-09-25 17:37:32,959  28136 [Engine       ] INF: ZAdmin, current
state:active: state tag changed to provision
2023-09-25 17:37:32,975  28136 [Env          ] DEB: No MTU configs detected
2023-09-25 17:37:32,977  28136 [Engine       ] DEB: ZAdmin, current
state:active. Processing work: ZAdmin: Apply configuration. done = False
2023-09-25 17:37:33,021  28136 [Xr           ] DEB: Will apply the following
config: /disk0:/ztp/customer/config.candidate
2023-09-25 17:37:33,022  28136 [Xr          ] INF: Applying user configurations
2023-09-25 17:37:33,023  28136 [Configuration] INF: Provisioning via config
replace
2023-09-25 17:38:14,445  28136 [Configuration] INF: Configuration has been
applied
2023-09-25 17:38:14,447  28136 [Env          ] DEB:
cfg::createRefOnConfigCommit: called
2023-09-25 17:38:15,778  28136 [Env          ] DEB: cfg:: Generating hash for
 File name: /disk0:/ztp/customer/config.inithash_tmp
2023-09-25 17:38:15,780  28136 [Env          ] DEB:
cfg::_generateCfgAndSaveHash:: HASH :
643013d9a43a3d2576012a24eb9745a8f960480d0053d06ed81146cb3c3d54c5, type : 1
2023-09-25 17:38:17,743  28136 [Env         ] DEB: cfg::getRefOnConfigCommit:
 called
2023-09-25 17:38:17,818  28136 [Env          ] DEB: cfg::getRefOnConfigCommit
 :: ret : data :
643013d9a43a3d2576012a24eb9745a8f960480d0053d06ed81146cb3c3d54c5, len: 64
2023-09-25 17:38:17,819  28136 [Env          ] INF:
Env::getConfigRefHashOnCommit: get data from tam :
success:b'643013d9a43a3d2576012a24eb9745a8f960480d0053d06ed81146cb3c3d54c5'
2023-09-25 17:38:17,821  28136 [Engine       ] DEB: ZAdmin, current
state:active. Processing work: Sending standby sync message. done = False
2023-09-25 17:38:17,836  28136 [Engine       ] DEB: ZAdmin, current
state:active. Processing work: [privileged] getting engine status. done = False
2023-09-25 17:38:17,837  28136 [Engine       ] DEB: ZAdmin, current
state:active. Processing work: ZAdmin: Execute post-configuration script. done
 = False
2023-09-25 17:38:17,873  28136 [Env         ] INF: Env::cleanup, success:True,
 exiting:False
2023-09-25 17:38:17,876  28136 [ZtpHelpers   ] DEB: Executing: source
/pkg/bin/ztp_helper.sh && echo -ne  | xrcmd "show running-config"
2023-09-25 17:38:19,582  28136 [Env          ] INF: Executing command ip netns
 exec vrf-default /sbin/dhclient -4 -cf /etc/dhcp/dhclient.conf.ztp -lf
/var/lib/dhcp/dhclient.leases.ztp -sf /etc/dhcp/dhclient-script.ztp2 -r
Mg0_RP0_CPU0_0 to release IP
2023-09-25 17:38:20,695  28136 [Xr           ] INF: Removing linux route with
 ip 10.105.57.107
2023-09-25 17:38:20,731  28136 [Xr           ] INF: Failed to remove default
route to to_xr via 10.105.57.107 with error: Error: RTNETLINK answers: No such
 process encountered while executing command: ip netns exec vrf-default ip
route del default dev to_xr src 10.105.57.107 metric 512
2023-09-25 17:38:20,736  28136 [Engine       ] INF: ZAdmin, current
state:active, exit code:success
2023-09-25 17:38:20,737  28136 [Engine      ] INF: ZAdmin, current state:final,
 exit code:success: state changed to final
2023-09-25 17:38:22,846  28136 [Engine      ] DEB: ZAdmin, current state:final,
 exit code:success. Processing work: Sending standby sync message. done = False
2023-09-25 17:38:22,847  28136 [Engine      ] WAR: ZAdmin, current state:final,
```

```
 exit code:success: work is ignored: work=<desc='Sending standby sync message'
 done=False priv=False>
2023-09-25 17:38:22,848  28136 [Engine      ] DEB: ZAdmin, current state:final,
 exit code:success. Processing work: [privileged] getting engine status. done
 = False
2023-09-25 17:38:27,853  28136 [__main__     ] DEB: Moved to final state
2023-09-25 17:38:27,854  28136 [__main__     ] DEB: ZTP completed successfully
2023-09-25 17:38:27,855  28136 [__main__     ] INF: Exiting SUCCESSFULLY
```

Tech support information for ZTP on Cisco NCS 1014

Lists the command that gathers ZTP debugging information on Cisco NCS 1014 and shows the location of the generated archive that you share with the Cisco technical support representative.

Tech support information for Zero Touch Provisioning on Cisco NCS 1014

Command to gather ZTP debugging information

Run the **show tech-support ztp** command to collect all debugging information of the ZTP process.

```
RP/0/RP0/CPU0:ios#show tech-support ztp
Thu Jul 28 08:33:27.531 UTC
++ Show tech start time: 2022-Jul-28.083327.UTC ++
Thu Jul 28 08:33:28 UTC 2022 Waiting for gathering to complete
..
Thu Jul 28 08:33:34 UTC 2022 Compressing show tech output
Show tech output available at 0/RP0/CPU0 :
/harddisk:/showtech/showtech-R1-ZTP-2022-Jul-28.083327.UTC.tgz
++ Show tech end time: 2022-Jul-28.083334.UTC ++
RP/0/RP0/CPU0:ios#
```

In this example, the tech support information is saved as a `.tgz` file in the specified location. You can share this information with the Cisco technical support representative for troubleshooting the ZTP process.

Configure a management interface

Configure an IP address, subnet mask, and static route on the Cisco NCS 1014 management interface for system management and remote communication.

Use this task to configure the management interface on Cisco NCS 1014 when the chassis is not booted using ZTP.

The management interface enables system management and remote communication. Assigning an IP address and subnet mask allows for system management, while configuring a static route enables communication with devices on other networks.

Before you begin

- Obtain the IP addresses and subnet mask for the management interface from your network administrator.
- Verify that the management interface is connected to the management network.

Procedure

1. Enter IOS XR configuration mode.

Example

```
RP/0/RP0/CPU0:ios#configure
```

2. Enter interface configuration mode for the management interface using the command **interface mgmtEth** *rack/slot/instance/port*

   Example

   ```
   RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/0
   ```

3. Assign an IP address and a subnet mask to the management interface using the command **ipv4 address** *ipv4-address subnet-mask*.

   Example

   ```
   RP/0/RP0/CPU0:ios(config-if)#ipv4 address 10.1.1.1 255.0.0.0
   ```

4. Place the management interface in an up state.

   Example

   ```
   RP/0/RP0/CPU0:ios(config-if)#no shutdown
   ```

5. Exit the management interface configuration mode.

   Example

   ```
   RP/0/RP0/CPU0:ios(config-if)#exit
   ```

6. Specify the IP address of the default gateway to configure a static route using the command **router static address-family ipv4 unicast** *0.0.0.0/0 default-gateway*. This IP address is used for communication with devices on other networks.

   Example

   ```
   RP/0/RP0/CPU0:ios(config)#router static address-family ipv4 unicast 0.0.0.0/0
    10.25.0.1
   ```

7. Save the configuration changes by using the **commit** or **end** command.

The management interface of Cisco NCS 1014 is configured with an IP address, subnet mask, and a static route to the default gateway.

What to do next

Connect the management interface to the Ethernet network. Establish an SSH or Telnet connection to the management interface using its IP address.

Boot the Cisco NCS 1014 through the console port

Configure the console connection and boot process required to start Cisco NCS 1014 and create the root-system username.

Boot the Cisco NCS 1014 system and complete the initial setup through the console port.

Use the console port to connect to Cisco NCS 1014. By default, the console port connects to XR mode. Once the management port is configured, you can use it for subsequent connections.

Procedure

1. Connect a terminal to the console port of the RP.
2. Start the terminal emulation program on your workstation.

   The console settings are 115200 bps for NCS1K14-CNTLR-K9, 9600 bps for NCS1K14-CTLR-B-K9, 8 data bits, 1 stop bit, and no parity.
3. Power on the Cisco NCS 1014.

   To power on the shelves, install the AC or DC power supplies and cables. As Cisco NCS 1014 boots up, you can view the boot process details at the console of the terminal emulation program.
4. Press **Enter**.

   The boot process is complete when the system prompts you to enter the root-system username. If the prompt does not appear, wait for a while to give Cisco NCS 1014 more time to complete the initial boot procedure; then press **Enter**.

   > **Note**
   >
   > If the boot process fails, it may be because the preinstalled image on the Cisco NCS 1014 is corrupt. In this case, you can boot Cisco NCS 1014 using an external bootable USB drive.

Cisco NCS 1014 completes the boot process and prompts you to create the root-system username for logging on to the XR console.

Boot Cisco NCS 1014 using a USB drive

Reimages Cisco NCS 1014 for system upgrade or recovery from boot failure by preparing a bootable USB drive and booting the chassis from it.

Boot Cisco NCS 1014 using a bootable USB drive to reimage the system for upgrade or recovery.

A bootable USB drive allows you to reimage Cisco NCS 1014 for a system upgrade or to recover from a boot failure. The drive is prepared by copying a compressed boot file onto it and extracting the contents. Preparation steps can be performed with Windows, Linux, or Mac operating systems.

Before you begin

- Obtain a USB drive with a minimum storage capacity of 4 GB.

- Make sure the USB drive has a single partition.
- Copy the compressed boot file from the software download page at Cisco.com to your local machine. The filename for the compressed boot file is in the format *ncs1010-x64-usb-(release_number).zip*.

Procedure

1. Prepare a bootable USB drive on your local machine:
   a) Connect the USB drive to your local machine and format it with the FAT32 file system.
   b) Copy the compressed boot file to the USB drive.
   c) Verify that the copy operation is successful.

      To verify, compare the file size at source and destination. Also, verify the MD5 checksum value.

   d) Extract the content of the compressed boot file by unzipping it in the USB drive.

      Extracting the content makes the USB drive a bootable drive.

      Note

      You must extract the contents of the zipped file ("EFI" and "boot" directories) directly in the root folder of the USB drive. If the unzipping application places the extracted files in a new folder, move the "EFI" and "boot" directories to the root folder of the USB drive.

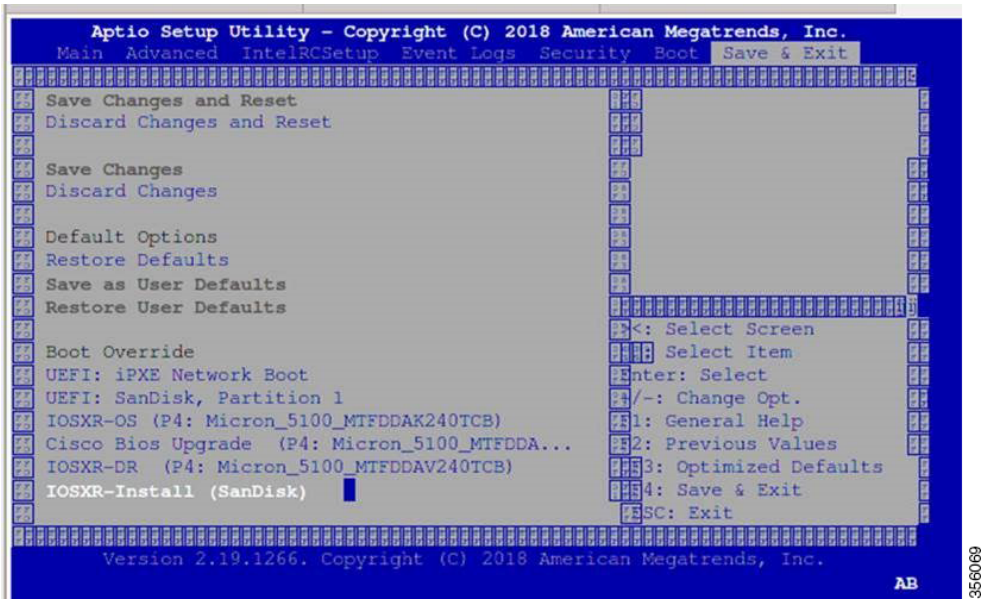
2. Boot Cisco NCS 1014 from the bootable USB drive.
   a) Insert the USB drive in one of the USB ports of Cisco NCS 1014.
   b) Reboot Cisco NCS 1014 using power cycle or console.

      Note

      Use the **reload bootmedia usb noprompt** command to boot the Cisco NCS 1014 from the USB. If you are using the **reload bootmedia usb noprompt** command, then you can skip the remaining steps.

   c) Press **Esc** to enter BIOS.
   d) Select the **Save & Exit** tab of BIOS.

      Figure 2: BIOS page



e) Select **IOS -XR Install**.

The BIOS UI displays the USB drive vendor in the brackets, in this case, SMART USB 1084.

The system detects USB and boots the image from USB.

```
Booting from USB..
Loading Kernel..
Verifying (loop)/boot/bzImage...
(loop)/boot/bzImage verified using attached signature.
Loading initrd..
Verifying (loop)/boot/initrd.img...
```

f) Remove the USB drive after the `Rebooting the system after installation` message is displayed.

Cisco NCS 1014 reboots automatically.

> **Note**
>
> The USB must be removed only after the image is loaded successfully.

Cisco NCS 1014 boots from the USB drive and starts the IOS XR installation.

iPXE boot methods

Explains the iPXE Preboot Execution Environment included in management interface network cards and its role in reimaging Cisco NCS 1014 or booting the chassis when the bootable partition is missing or has failed.

A iPXE boot method is a system firmware process that

- operates at the UEFI level using the network card of management interfaces,
- enables you to reimage the Cisco NCS 1014 or boot the chassis in case of missing or failed bootable partitions, and

- downloads and installs the ISO image, then bootstraps the system into the new installation.

iPXE acts as a bootloader and provides the flexibility to choose the image that the system will boot based on the Platform Identifier (PID), the Serial Number, or the management mac-address. You must define iPXE in the DHCP server configuration file.

iPXE boot methods

> **Note**
>
> The time taken for iPXE to download the ISO image depends on the network speed. Ensure that the network speed is sufficient to complete the image download in less than 10 minutes. The chassis reloads if the image is not downloaded by 10 minutes.

To start the iPXE boot process, use one of these methods.

- **Preferred method:** Run either the **reload bootmedia network location all** command or the **reload bootmedia network location 0/RP0** command.
- **Alternative method:** Power cycle the Cisco NCS 1014 chassis and start the iPXE boot process in the BIOS interface.

> **Note**
>
> Software installation using iPXE boot with IPv6 is not supported.

Configure the DHCP server

Configure a DHCP server with the required network information for iPXE to identify the chassis and download the ISO image, script, and provisioning configuration file.

Set up a DHCP server for IPv4, IPv6, or both protocols so that iPXE can boot Cisco NCS 1014.

A DHCP server is essential for the iPXE boot process and must be configured to support either IPv4, IPv6, or both. This configuration allows the network client to obtain the necessary boot and provisioning files automatically.

Note

For DHCPv6, a routing advertisement (RA) message must be sent to all nodes in the network to indicate which method is to be used to obtain the IPv6 address. Configure the Router-advertise-daemon (radvd, install using **yum install radvd**) to allow the client to send the DHCP request. For example:

```
interface eth3
{
        AdvSendAdvert on;
        MinRtrAdvInterval 60;
        MaxRtrAdvInterval 180;
        AdvManagedFlag on;
        AdvOtherConfigFlag on;
        prefix 2001:1851:c622:1::/64
        {
                AdvOnLink on;
                AdvAutonomous on;
                AdvRouterAddr off;
        };
};
```

Procedure

1. Create the `dhcpd.conf` file (for IPv4, IPv6, or both communication protocols), the `dhcpv6.conf` file (for IPv6), or both, in the `/etc/` directory.

   This configuration file stores the network information such as the path to the script, the location of the ISO install file, the location of the provisioning configuration file, the serial number, and the MAC address of the chassis.

2. Test the server when the DHCP server is running.

   For IPv4, you can test the server by using either the MAC address or the serial number of the chassis:

   - Use the MAC address of the chassis and ensure that the configuration is successful.
   - Use the serial number of the chassis. The serial number of the chassis is derived from the BIOS and is used as an identifier.

   ```
   host demo {
   option dhcp-client-identifier "<chassis-serial-number>";
     filename "http://<IP-address>/<hardware-platform>-mini-x.iso";
     fixed-address <IP-address>;
   }
   ```

   Example

   ```
   host 10.89.205.202 {
          hardware ethernet 40:55:39:xx:xx:xx;
           option dhcp-client-identifier "<FCB2437B066>";
             if exists user-class and option user-class = "iPXE" {
             filename "http://10.89.205.127/box1/ncs1010-x64.iso";
          } else {
              filename "http://10.89.205.127/box1/StartupConfig.cfg";
          }
   ```

```
        fixed-address 10.89.205.202;
    }
```

The DHCP server is configured and is ready to respond to iPXE boot requests from Cisco NCS 1014.

Boot the Cisco NCS 1014 using iPXE

Configure the Cisco NCS 1014 to boot using iPXE, either through the IOS XR CLI or via the BIOS interface.

Enable network boot and reimage of the Cisco NCS 1014 using iPXE.

Use this task to boot the Cisco NCS 1014 from the network with iPXE for reimaging, using either the command-line interface or BIOS selections.

Before you begin

Before you perform the iPXE boot, ensure that:

- The DHCP server is set and running.
- The management port of the Cisco NCS 1014 chassis is in the UP state.

Procedure

1. Run the **reload bootmedia network location all** command or the **reload bootmedia network location 0/RP0** command to initiate the iPXE boot process and reimage the chassis.

   Example

```
RP/0/RP0/CPU0:ios#reload bootmedia network location all
Mon Dec  4 09:49:14.220 UTC
Proceed with reload? [confirm]
Preparing system for backup. This may take a few minutes especially for large
 configurations.
 Status report: node0_RP0_CPU0: BACKUP INPROGRESS
 Status report: node0_RP0_CPU0: BACKUP HAS COMPLETED SUCCESSFULLY
[Done]

[ 3490.493853] systemd-shutdown[1]: Could not detach DM /dev/dm-9: Device or
 resource busy
[ 3490.601094] systemd-shutdown[1]: Could not detach DM /dev/dm-8: Device or
 resource busy
[ 3490.710401] systemd-shutdown[1]: Could not detach DM /dev/dm-7: Device or
 resource busy
[ 3490.849417] systemd-shutdown[1]: Failed to finalize  DM devices, ignoring
[ 3492.144874] Unsupported TPM Send Cmd! tpm_tag=8001,tpm_ordinal=0145
[ 3492.229149] tpm tpm0: tpm_try_transmit: send(): error -11
[ 3492.307885] reboot: Restarting system
Shelf Assembly Reset
NCS1014, Initializing Devices
Booting from Primary Flash
Aldrin: Programmed MI 4
Continue boot...
Version 2.19.1266. Copyright (C) 2023 American Megatrends, Inc.
BIOS Date: 10/06/2023 16:47:27 Ver: 0ACHI0480
Press <DEL> or <ESC> to enter setup.
TAM: Chip DB Verified
```

```
Software Boot OK, Validated
iPXE initialising devices...ok

iPXE 1.0.0+ (8b3e3) -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS HTTP HTTPS TFTP VLAN EFI ISO9660 ISO9660_grub Menu
Trying net0-2051 and net0-2052...
net0-2051: 40:14:82:ba:d1:42 using NII on NII-0000:06:00.0 (open)
  [Link:down, TX:0 TXE:0 RX:0 RXE:0]
  [Link status: Unknown (http://ipxe.org/1a086194)]
```

Example

```
RP/0/RP0/CPU0:sanity_176#reload bootmedia network location 0/RP0
Mon Feb 10 16:34:42.282 IST
Proceed with reload? [confirm]

RP/0/RP0/CPU0:Feb 10 16:34:44.277 IST: shelfmgr_exec_cli[65652]:
%PLATFORM-SHELFMGR-6-USER_OP : User mols requested 'PXE reimage' of 0/RP0
RP/0/RP0/CPU0:Feb 10 16:34:44.284 IST: processmgr[51]: %OS-SYSMGR-6-INFO :
Received request for graceful go-down from shelfmgr. Reload Reason:User
initiated card reimage-network. Timeout 55

Preparing system for backup. This may take a few minutes especially for large
 configurations.
RP/0/RP0/CPU0:Feb 10 16:34:44.284 IST: processmgr[51]: %OS-SYSMGR-6-INFO :
Prepared RMF to reboot
 Status report: node0_RP0_CPU0: BACKUP INPROGRESS
RP/0/RP0/CPU0:sanity_176# Status report: node0_RP0_CPU0: BACKUP HAS COMPLETED
 SUCCESSFULLY
[Done]
RP/0/RP0/CPU0:Feb 10 16:34:45.974 IST: obflmgr[317]: %PLATFORM-OBFL-6-INFO :
 Unmounted OBFL directory for 0/Rack
RP/0/RP0/CPU0:Feb 10 16:34:46.225 IST: obflmgr[317]: %PLATFORM-OBFL-6-INFO :
 Unmounted OBFL directory for 0/RP0/CPU0
RP/0/RP0/CPU0:Feb 10 16:34:46.555 IST: processmgr[51]:
%MGBL-SCONBKUP-6-INTERNAL_INFO : Reload debug script successfully spawned
--
--
[438336.926067] reboot: Restarting system


..         *** Cisco ***
System Initializing..
..


Initializing Devices

Booting from Primary Flash
MI: Skipping reprogram
Version 2.19.1266. Copyright (C) 2024 American Megatrends, Inc.
BIOS Date: 10/28/2024 12:47:04 Ver: 0ACHI0500
Press <DEL> or <ESC> to enter setup.
TAM: Chip DB Verified

Software Boot OK, Validated

iPXE initialising devices...ok

iPXE 1.0.0+ (8b3e3) -- Open Source Network Boot Firmware -- http://ipxe.org
```

```
Features: DNS HTTP HTTPS TFTP VLAN EFI ISO9660 ISO9660_grub Menu
Trying net0-2051 and net0-2052...
net0-2051: 9c:38:18:88:a4:1d using NII on NII-0000:06:00.0 (open)
  [Link:down, TX:0 TXE:0 RX:0 RXE:0]
  [Link status: Unknown (http://ipxe.org/1a086194)]
Configuring (net0-2051 9c:38:18:88:a4:1d)...... ok
--
--
net0-2052: fe80::9e38:18ff:fe88:a41e/64
Filename: http://4.25.30.34/Kepler/25_1_1_Images/ncs1010-x64.iso
http://4.25.30.34/Kepler/25_1_1_Images/ncs1010-x64.iso... ok
ncs1010-x64.iso : 1810485248 bytes
Booting /EFI/BOOT/bootx64.efi
Welcome to GRUB!
```

2. Aternatively, initiate the boot process using the BIOS interface.
   a) Reboot Cisco NCS 1014 using power cycle or console.
   b) Press **Esc** to enter BIOS.
   c) Select the **Save & Exit** tab of BIOS.
   d) Choose **UEFI: iPXE Network Boot**.

   Example

```
TAM: Chip DB Verified
Software Boot OK, Validated
iPXE initialising devices...ok
iPXE 1.0.0+ (8b3e3) -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS HTTP HTTPS TFTP VLAN EFI ISO9660 ISO9660_grub Menu
Trying net0-2051 and net0-2052...
net0-2051: 9c:38:18:88:a4:1d using NII on NII-0000:06:00.0 (open)
[Link:down, TX:0 TXE:0 RX:0 RXE:0]
[Link status: Unknown (http://ipxe.org/1a086194)]
Configuring (net0-2051 9c:38:18:88:a4:1d)...... ok
net0: fe80::9e38:18ff:fe88:a41d/64
net1: fe80::9e38:18ff:fe88:a420/64 (inaccessible)
net2: fe80::9e38:18ff:fe88:a421/64 (inaccessible)
net3: fe80::9e38:18ff:fe88:a422/64 (inaccessible)
net0-2051: 4.25.9.3/255.255.0.0 gw 4.25.0.1
net0-2051: fe80::9e38:18ff:fe88:a41d/64
net0-2052: fe80::9e38:18ff:fe88:a41e/64
Filename: http://4.25.30.34/Kepler/25_1_1_Images/ncs1010-x64.iso
http://4.25.30.34/Kepler/25_1_1_Images/ncs1010-x64.iso... ok
ncs1010-x64.iso : 1810485248 bytes
Booting /EFI/BOOT/bootx64.efi
Welcome to GRUB!
/EndEntire
error: no such device: ((cd0)/EFI/BOOT)/EFI/BOOT/grub.cfg.
Verifying (cd0)/EFI/BOOT/grub.cfg...
(cd0)/EFI/BOOT/grub.cfg verified using Pkcs7 signature.
```

The Cisco NCS 1014 successfully initiates the iPXE network boot process, loading the designated image and displaying boot confirmation messages. Once complete, the system is ready for further installation or configuration.

Insert and verify operation of a Cisco NCS 1014 line card

Configure, verify, and prepare a Cisco NCS 1014 line card for operation, including physical installation, operational checks, environmental verification, and FPD upgrade.

Install the Cisco NCS 1014 line card and ensure it becomes fully operational within the chassis.

Use this task whenever you add or replace a Cisco NCS 1014 line card. These steps confirm physical installation, operating status, environmental health, and that firmware is current.

Procedure

1. Insert the line card into the slot.
2. Wait until the line card LED turns green.
3. Check that the PID is in **OPERATIONAL** status using the **show platform** command.

Example

CCMD-16-C and CCM-16-L line cards:

```
RP/0/RP0/CPU0:ios#show platform
Fri Sep 22 06:56:28.653 UTC
Node               Type                      State                     Config
state
--------------------------------------------------------------------------------
0/RP0/CPU0       NCS1K14-CNTLR-K9(Active) IOS XR RUN              NSHUT,NMON
0/PM0            NCS1K4-AC-PSU           OPERATIONAL              NSHUT,NMON
0/FT0            NCS1K14-FAN             OPERATIONAL              NSHUT,NMON
0/FT1            NCS1K14-FAN             OPERATIONAL              NSHUT,NMON
0/FT2            NCS1K14-FAN             OPERATIONAL              NSHUT,NMON
0/0/NXR0         NCS1K14-CCMD-16-L       OPERATIONAL              NSHUT,NMON
0/2/NXR0         NCS1K14-CCMD-16-C       OPERATIONAL              NSHUT,NMON
0/3/NXR0         NCS1K14-CCMD-16-C       OPERATIONAL              NSHUT,NMON
```

Example

2.4T line card:

```
RP/0/RP0/CPU0:ios#show platform
Fri Sep 22 06:56:28.653 UTC
Node               Type                      State                     Config
state
--------------------------------------------------------------------------------
0/RP0/CPU0       NCS1K14-CNTLR-K9(Active) IOS XR RUN              NSHUT,NMON
0/PM0            NCS1K4-AC-PSU           OPERATIONAL              NSHUT,NMON
0/FT0            NCS1K14-FAN             OPERATIONAL              NSHUT,NMON
0/FT1            NCS1K14-FAN             OPERATIONAL              NSHUT,NMON
0/FT2            NCS1K14-FAN             OPERATIONAL              NSHUT,NMON
0/2/NXR0         NCS1K14-2.4T-K9         OPERATIONAL              NSHUT,NMON
0/3/NXR0         NCS1K14-BLANK             PRESENT                NSHUT,NMON
```

Example

2.4TX line card:

```
RP/0/RP0/CPU0:ios#show platform
Node               Type                       State                     Config
state
--------------------------------------------------------------------------------
0/RP0/CPU0       NCS1K14-CNTLR-K9(Active) IOS XR RUN              NSHUT,NMON
0/PM0            NCS1K4-AC-PSU-2           OPERATIONAL            NSHUT,NMON
0/FT0            NCS1K14-FAN              OPERATIONAL             NSHUT,NMON
0/FT1            NCS1K14-FAN              OPERATIONAL             NSHUT,NMON
0/FT2            NCS1K14-FAN              OPERATIONAL             NSHUT,NMON
0/0/NXR0         NCS1K14-2.4T-X-K9        OPERATIONAL             NSHUT,NMON
0/1/NXR0         NCS1K14-2.4T-X-K9        OPERATIONAL             NSHUT,NMON
0/2/NXR0         NCS1K14-2.4T-X-K9        OPERATIONAL             NSHUT,NMON
0/3/NXR0         NCS1K14-2.4T-X-K9        OPERATIONAL             NSHUT,NMON
```

Example

EDFA2 line card:

```
RP/0/RP0/CPU0:sanity_176#show platform
Node               Type                       State                     Config
state
--------------------------------------------------------------------------------
0/RP0/CPU0       NCS1K14-CNTLR-K9(Active) IOS XR RUN              NSHUT,NMON
0/PM0            NCS1K4-AC-PSU-2          OPERATIONAL             NSHUT,NMON
0/PM1            NCS1K4-AC-PSU-2          OPERATIONAL             NSHUT,NMON
0/FT0            NCS1K14-FAN              OPERATIONAL             NSHUT,NMON
0/FT1            NCS1K14-FAN              OPERATIONAL             NSHUT,NMON
0/FT2            NCS1K14-FAN              OPERATIONAL             NSHUT,NMON
0/0/NXR0         NCS1K14-BLANK             PRESENT                NSHUT,NMON
0/1/NXR0         NCS1K4-1.2T-K9           OPERATIONAL             NSHUT,NMON
0/2/NXR0         NCS1K14-BLANK             PRESENT                NSHUT,NMON
0/3/NXR0         NCS1K14-EDFA2            OPERATIONAL             NSHUT,NMON
0/4              NCS1K-MD-32O-CE          OPERATIONAL             NSHUT,NMON
0/5              NCS1K-MD-32E-CE          OPERATIONAL             NSHUT,NMON
```

Example

2.4TA line card:

```
RP/0/RP0/CPU0:ios#show platform
Mon Jan 12 11:36:27.652 UTC
Node               Type                       State                     Config
state
--------------------------------------------------------------------------------
0/RP0/CPU0       NCS1K14-CNT-B-K9(Active) IOS XR RUN              NSHUT,NMON
0/PM1            NCS1K4-AC-PSU            OPERATIONAL             NSHUT,NMON
0/FT0            NCS1K14-FAN              OPERATIONAL             NSHUT,NMON
0/FT1            NCS1K14-FAN              OPERATIONAL             NSHUT,NMON
0/FT2            NCS1K14-FAN              OPERATIONAL             NSHUT,NMON
0/0/NXR0         NCS1K14-BLANK             PRESENT                NSHUT,NMON
0/1/NXR0         NCS1K14-2.4T-A-K9        OPERATIONAL             NSHUT,NMON
0/2/NXR0         NCS1K14-2.4T-A-K9        OPERATIONAL             NSHUT,NMON
0/3/NXR0         NCS1K14-BLANK             PRESENT                NSHUT,NMON
```

4. Check the line card environment parameters using the **show environment [power | voltage | current | temperature] [location** *location*] command.

   Example

```
RP/0/RP0/CPU0:ios#show environment power
===============================================================================
CHASSIS LEVEL POWER INFO: 0
===============================================================================

   Total output power capacity (Group 0 + Group 1) :    2500W +     0W
   Total output power required                     :    1636W
   Total power input                               :     637W
   Total power output                              :     568W

Power Group 0:
===============================================================================

   Power       Supply          ------Input----   ------Output---     Status
   Module      Type             Volts     Amps    Volts     Amps
===============================================================================

   0/PM1       NCS1K4-AC-PSU-2 227.5      2.8     12.1      47.0     OK

Total of Group 0:              637W/2.8A         568W/47.0A


===============================================================================
Location     Card Type         Power                Power              Status
                               Allocated            Used
                               Watts                Watts
===============================================================================
0/RP0/CPU0  NCS1K14-CNTLR-K9   73                    20                  ON
0/FT0       NCS1K14-FAN        170                   167                 ON
0/FT1       NCS1K14-FAN        170                   85                  ON
0/FT2       NCS1K14-FAN        170                   159                 ON
0/0/NXR0    NCS1K14-2.4T-L-K9  460                   38                  ON
0/2/NXR0    NCS1K14-2.4T-X-K9  410                   0                   ON
0/3/NXR0    NCS1K14-CCMD-16-C  110                   16                  ON
0/Rack      NCS1014            73                    14                  ON
```

> **Note**
>
> - Insert a filler into any unused slot to ensure proper airflow and optimal system temperature.
> - Insert a clip into any unused port to maintain optimal card temperature.
> - Tighten the top and bottom screws to secure the line card in the chassis.

5. Upgrade the FPDs of the line card depending on the output of the **show hw-module location 0/***line-card-slot* **fpd** command.

The Cisco NCS 1014 line card is physically installed, operational, has passed environment checks, and has updated firmware levels as necessary.

NTP servers

Explains how Network Time Protocol (NTP) servers synchronize device clocks across a network to ensure accurate and consistent timekeeping.

A Network Time Protocol (NTP) server is a network device that

- provides a common, authoritative time source for other devices,
- synchronizes network clocks using the User Datagram Protocol (UDP) and Coordinated Universal Time (UTC), and
- distributes accurate time across the network to maintain consistency and support management, security, and troubleshooting operations.

Importance of time synchronization: In modern and large networks, time synchronization is critical because every aspect of managing, securing, planning, and debugging a network depends on accurate timestamps for the occurrence of events.

Table 2: Feature History

Feature Name	Release Information	Feature Description
NTP Support	Cisco IOS XR Release 7.11.1	Network Time Protocol (NTP) allows devices to synchronize clocks with the NTP servers, maintaining the most accurate time. NCS 1010 now supports time synchronization. In modern and large networks, time synchronization is critical because every aspect of managing, securing, planning, and debugging a network depends on the time of occurrence of events. Commands added: • **ntp server** • **show ntp associations** • **show ntp status**

NTP key features and operation

NTP uses the User Datagram Protocol (UDP) as its transport protocol.

- NTP communication is always based on Coordinated Universal Time (UTC).
- NTP networks usually receive their time from an authoritative source, such as a radio clock or atomic clock attached directly to a time server. NTP distributes this time across the network.
- The "stratum" concept describes the number of NTP hops a machine is from an authoritative time source:
  - A stratum 1 server is directly connected to an authoritative source (for example, a radio or atomic clock, or a GPS time source).
  - A stratum 2 server receives its time from a stratum 1 server via NTP, and so forth.

- Machines running NTP, called "associations," are usually statically configured with the IP addresses of machines with which they should form associations. Accurate timekeeping is achieved by exchanging NTP messages between each pair of associated machines.
- An NTP broadcast client listens for broadcast messages sent by an NTP broadcast server at a designated IPv4 address. The client synchronizes the local clock using the first received broadcast message.

Synchronize the clock with an NTP server

Configure the IOS XR system clock on Cisco NCS 1014 to synchronize with a specified NTP server using IPv4 or IPv6, ensuring accurate system time.

Synchronize the IOS XR system clock with an NTP server to prevent time deviation and maintain precise system operation.

IOS XR features an independent system clock, which must be periodically synchronized with an NTP server to avoid drifting from true time. Accurate system time is essential for proper logging, scheduling, and network functionality.

Procedure

1. Enter configuration mode.

   Example

   ```
   RP/0/RP0/CPU0:ios# configure
   ```

2. Configure the NTP server using the command **ntp server** *ntp-server-ip-address* **version** *version-number* **prefer iburst**.

   Example

   IPv4:

   ```
   RP/0/RP0/CPU0:ios(config)#ntp server 10.127.60.137 version 4 prefer iburst
   ```

   IPv6:

   ```
   RP/0/RP0/CPU0:ios(config)#ntp server 2001:DB8::1 version 4 prefer iburst
   ```

   > **Note**
   >
   > The NTP server can also be reached through a VRF if the management interface is in a VRF.

3. Commit the configuration.

   Example

   ```
   RP/0/RP0/CPU0:ios(config-ntp)#commit
   ```

4. Verify NTP associations using the command **show ntp associations**

Example

```
RP/0/RP0/CPU0:ios#show ntp associations
Tue Oct 17 06:32:54.389 UTC

      address         ref clock     st  when  poll reach  delay  offset
disp
 ~10.0.0.1       10.64.58.50       2    15    64    1    1.95  -0.062  1937.7

 ~10.127.60.137    .STEP.           16     -    64    0    0.00   0.000
15937
 * sys_peer, # selected, + candidate, - outlayer, x falseticker, ~ configured
```

5. Verify that the clock is synchronized using the command **show ntp status**.

Example

```
RP/0/RP0/CPU0:ios#show ntp status
Tue Oct 17 06:33:41.125 UTC
Clock is synchronized, stratum 3, reference is 10.0.0.1
nominal freq is 1000000000.0000 Hz, actual freq is 31762818.6272 Hz, precision
 is 2**24
reference time is E8D8A944.A4F2AEF1 (06:33:40.644 UTC Tue Oct 17 2023)
clock offset is -0.592 msec, root delay is 2.434 msec
root dispersion is 939.64 msec, peer dispersion is 938.03 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.0000304834
s/s
system poll interval is 64, last update was 1 sec ago
authenticate is disabled, panic handling is disabled,
hostname resolution retry interval is 1440 minutes.
```

The IOS XR system clock on Cisco NCS 1014 is synchronized with the NTP server, ensuring accurate system time.

Verify external reference clock status

Verifies the synchronization status of the IOS XR system clock and NTP associations on Cisco NCS 1014 with the external NTP reference clock.

Confirm that the Cisco NCS 1014 system clock is synchronized with an external NTP server and check the status of all relevant NTP associations.

Use this task to ensure accurate system time and network synchronization for the Cisco NCS 1014, which relies on an external reference clock via NTP. The commands described can be entered in any order.

Procedure

---

1. Run the command **show ntp associations [detail] [location** *node-id***]** to view NTP associations.

Example

```
RP/0/RP0/CPU0:ios#show ntp associations
Sun Nov 5 15:14:44.128 UTC

address ref clock st when poll reach delay offset disp
*~10.0.0.1 10.64.58.50 2 81 128 377 1.84 7.802 2.129
```

```
* sys_peer, # selected, + candidate, - outlayer, x falseticker, ~ configured
```

Displays the status of NTP associations.

Example

```
RP/0/RP0/CPU0:ios#show ntp associations detail
Sun Nov 5 15:14:48.763 UTC

10.0.0.1 configured, our_master, stratum 2
ref ID 10.64.58.50, time E8F22BB9.79D4A841 (14:56:57.475 UTC Sun Nov 5 2023)
our mode client, peer mode server, our poll intvl 128, peer poll intvl 128
root delay 0.6866 msec, root disp 1.04, reach 377, sync dist 6.2590
delay 1.84 msec, offset 7.802 msec, dispersion 2.129
precision 2**23, version 4
org time E8F22F92.B647E8FC (15:13:22.712 UTC Sun Nov 5 2023)
rcv time E8F22F92.B88F303C (15:13:22.720 UTC Sun Nov 5 2023)
xmt time E8F22F92.B88F303C (15:13:22.720 UTC Sun Nov 5 2023)
filtdelay = 1.844 1.772 1.983 1.954 1.945 2.000 1.902 1.778
filtoffset = 7.857 7.802 8.065 8.063 8.332 8.397 8.664 8.684
filterror = 0.000 0.060 1.995 2.055 4.050 4.110 6.060 6.120
```

Example

```
RP/0/RP0/CPU0:ios#show ntp associations detail location 0/RP0/CPU0
Sun Nov 5 15:38:15.744 UTC

10.0.0.1 configured, our_master, stratum 2
ref ID 10.64.58.50, time E8F233C0.5606A159 (15:31:12.336 UTC Sun Nov 5 2023)
our mode client, peer mode server, our poll intvl 128, peer poll intvl 128
root delay 0.7019 msec, root disp 0.47, reach 377, sync dist 5.6762
delay 2.01 msec, offset 7.226 msec, dispersion 3.856
precision 2**23, version 4
org time E8F23563.DE5D42D5 (15:38:11.868 UTC Sun Nov 5 2023)
rcv time E8F23563.E07C296D (15:38:11.876 UTC Sun Nov 5 2023)
xmt time E8F23563.E07C296D (15:38:11.876 UTC Sun Nov 5 2023)
filtdelay = 2.006 1.865 1.936 1.762 1.932 1.875 1.881 2.011
filtoffset = 7.210 7.305 7.372 7.226 7.298 7.258 7.251 7.224
filterror = 0.000 2.025 2.085 4.035 4.095 6.060 6.120 8.070
```

2. Run the command **show ntp status [location** *node-id***]** to verify the system clock's synchronization status.

Example

```
RP/0/RP0/CPU0:ios#show ntp status
Sun Nov 5 15:14:36.949 UTC

Clock is synchronized, stratum 3, reference is 10.0.0.1
nominal freq is 1000000000.0000 Hz, actual freq is 44881851.3383 Hz, precision
 is 2**24
reference time is E8F22D7A.AB020D97 (15:04:26.668 UTC Sun Nov 5 2023)
clock offset is 9.690 msec, root delay is 2.553 msec
root dispersion is 24.15 msec, peer dispersion is 2.13 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.0000212807
s/s
system poll interval is 128, last update was 610 sec ago
authenticate is disabled, panic handling is disabled,
hostname resolution retry interval is 1440 minutes.
```

Verifies that the clock is synchronized with the NTP server.

NTP troubleshooting information

Points you to a Cisco support article that contains troubleshooting information for Network Time Protocol (NTP).

For NTP troubleshooting information, see Troubleshooting Network Time Protocol.

2 Perform Preliminary Checks

Topics:

- Preliminary checks for Cisco NCS 1014
- Verify hardware components status
- Verify the software version
- Verify environmental parameters
- Verify management interface status
- Verify firmware version
- Display alarms on Cisco NCS 1014
- Verify core dump context information
- Verify core files

Provides comprehensive procedures for the initial bring-up and configuration of the Cisco NCS 1014. It covers system booting, ZTP, management interface setup, line card installation, and NTP synchronization to ensure efficient deployment and operation within service provider infrastructures.

Preliminary checks for Cisco NCS 1014

Describes the verification checks that you must perform after you log in to the Cisco NCS 1014 console so that you can confirm the default setup and correct any issue before further configuration.

A preliminary check for Cisco NCS 1014 is a post-login verification process that

- involves running a set of commands to confirm the system has come up correctly,
- checks the hardware inventory, the operational state of each card, and software image and firmware versions, and
- verifies environmental parameters, management connectivity, active alarms, core dumps, and core files.

Sample output and references

Note

The sample output in these procedures is not from the latest software release. The output changes for any explicit reference to the current release.

Note

- For details about proactive system monitoring that you can use to prevent potential issues, see System Health Check on page 107.

Inventory support in Cisco NCS 1014

Describes the hardware inventory model of Cisco NCS 1014, the supported cards and pluggables, and the unique device identifier (UDI) that the system reports for each component.

The Cisco NCS 1014 inventory model is the set of hardware components that the system reports when you query its inventory. It consists of:

- one Cisco NCS 1014 controller card,
- one Cisco NCS 1014 chassis,
- two AC or DC power supply units (PSU) of 2 KW and 2.5 KW,
- three FAN trays, and
- four line cards.

Table 3: Feature History

Feature Name	Release Information	Description
Inventory support	Cisco IOS XR Release 26.1.1	The NCS 1014 system enables inventory support for these new line cards. • NCS1K14-2.4T-A-K9 • NCS1K14-2.4TAL-K9
Inventory support	Cisco IOS XR Release 25.1.1	Inventory support and pluggable optics support are enabled in NCS 1014 system. • NCS1K-MD-32O-CE and NCS1K-MD-32E-CE • ONS-QSFP-OTDR • DP01QSDD-ZT5-A1 • ONS-SC-PTP-1510
Inventory support	Cisco IOS XR Release 7.11.1	Inventory support and pluggable optics support for QSFP28,QSFP-DD and Coherent Interface Module (CIM 8) are enabled in NCS 1014 system.

Card name mapping

The short names used in this guide map to the following Cisco line card product IDs (PIDs):

- CCMD-16-C: refers to the NCS1K14-CCMD-16-C card.
- CCMD-16-L: refers to the NCS1K14-CCMD-16-L card.
- 1.2T: refers to the NCS1K4-1.2T-K9 C-band card.
- 2.4T: refers to the NCS1K4-2.4T-K9 DWDM transponder and muxponder card.
- 2.4TA: refers to the NCS1K14-2.4T-A-K9 DWDM transponder and muxponder card.
- EDFA2: refers to the NCS1K4-EDFA2 C-band card.
- NCS1K-MD-32O-CE: refers to the NCS 1000 32-channel Odd Mux/Demux 150 GHz C-band Enhanced card.
- NCS1K-MD-32E-CE: refers to the NCS 1000 32-channel Even Mux/Demux 150 GHz C-band Enhanced card.

Unique device identifier (UDI) reported by show inventory

The **show inventory** command retrieves and displays inventory information for each Cisco product in the form of a unique device identifier (UDI). The UDI is a combination of three data elements:

- Product identifier (PID): The name by which the product is ordered. The PID is also known as the product name or part number.
- Version identifier (VID): The version of the product. The VID is incremented whenever an upgrade changes the product.
- Serial number (SN): The vendor-assigned unique serial number that identifies a specific product instance.

Verify the hardware inventory on Cisco NCS 1014 device

Verify the hardware inventory of Cisco NCS 1014 by running the **show inventory** command with the keyword that matches the information that you want to retrieve.

Use this task to confirm that the physical entities and modules installed in your Cisco NCS 1014 system are correctly identified, so that you can use the inventory data for operational and troubleshooting needs.

The **show inventory** command displays the inventory details of all the physical entities that are installed on Cisco NCS 1014. You can use specific keywords with the command to filter the inventory information you want to retrieve, such as all, details, fan, power, vendor-type, raw, chassis, and location.

Procedure

1. Run the **show inventory** command with the keyword that matches the inventory information that you want to retrieve.

   For sample outputs of each keyword variant, see Sample outputs of the show inventory command.

2. (Optional) Restrict the output to a specific node by appending the **location** *location* keyword to the command.

   Example

   ```
   RP/0/RP0/CPU0:ios#show inventory location 0/1/NXR0
   ```

Cisco NCS 1014 displays the inventory information that corresponds to the **show inventory** keyword that you ran.

Sample outputs of the show inventory command

Lists the sample outputs of the **show inventory** command for each keyword variant supported on Cisco NCS 1014.

The following sections show representative outputs for each keyword variant of the **show inventory** command. Use these samples together with the Verify the hardware inventory of Cisco NCS 1014 task.

show inventory

The following sample output shows the inventory summary of the chassis, controller, FAN modules, PSUs, and line cards.

```
RP/0/RP0/CPU0:ios#show inventory

NAME: "Rack 0", DESCR: "Network Convergence System 1014 chassis with timing
support"
PID: NCS1014           , VID: V00, SN: FCB2652B1XA

NAME: "0/RP0/CPU0", DESCR: "Network Convergence System 1014 Controller"
PID: NCS1K14-CNTLR-K9  , VID: V00, SN: FCB2717B1GB

NAME: "0/FT0", DESCR: "Network Convergence System 1014 FAN Module"
PID: NCS1K14-FAN       , VID: V00, SN: FCB2704B0P7

NAME: "0/FT1", DESCR: "Network Convergence System 1014 FAN Module"
PID: NCS1K14-FAN       , VID: V00, SN: FCB2720B18D

NAME: "0/FT2", DESCR: "Network Convergence System 1014 FAN Module"
PID: NCS1K14-FAN       , VID: V00, SN: FCB2704B0PA

NAME: "0/PM0", DESCR: "Network Convergence System 1004 AC Power Supply Unit
2.5KW"
PID: NCS1K4-AC-PSU-2   , VID: V00, SN: POG26300N0V
```

```
NAME: "0/PM1", DESCR: "Network Convergence System 1004 AC Power Supply Unit
2.5KW"
PID: NCS1K4-AC-PSU-2   , VID: V00, SN: POG26300N1H

NAME: "0/0/NXR0", DESCR: "NCS 1014 2.4T Line Card Ver 2"
PID: NCS1K14-2.4T-A-K9 , VID: V00, SN: FCB2921B2BM

NAME: "0/1/NXR0", DESCR: "NCS 1014 2.4T Licensed card-Ver2"
PID: NCS1K14-2.4TAL-K9 , VID: V00, SN: FCB2942B0JC

NAME: "0/2/NXR0", DESCR: "NCS 1014 2.4T licensed Card- crossconnect modes"
PID: NCS1K14-2.4TXL-K9 , VID: V01, SN: FCB2921B2WC

NAME: "0/3/NXR0", DESCR: "NCS 1014 2.4T Line Card Ver 2"
PID: NCS1K14-2.4T-A-K9 , VID: V00, SN: FCB2942B0J3
RP/0/RP0/CPU0:ios#
```

show inventory details

The following sample output shows the detailed inventory information, including the manufacturer name, SNMP index, hardware version, and part number for each physical entity.

```
RP/0/RP0/CPU0:ios#show inventory details

NAME: "Rack 0", DESCR: "Network Convergence System 1014 chassis with timing
support"
PID: NCS1014           , VID: V00, SN: FCB2652B1XA
MFG_NAME: Cisco Systems, Inc., SNMP_IDX: 8384513   , Type: Rack
PN: 800-111211-01, HW Ver: 0.1

NAME: "0/RP0/CPU0", DESCR: "Network Convergence System 1014 Controller"
PID: NCS1K14-CNTLR-K9  , VID: V00, SN: FCB2717B1GB
MFG_NAME: Cisco Systems, Inc., SNMP_IDX: 16385     , Type: Module
PN: 800-111209-01, HW Ver: 0.2

NAME: "0/FT0", DESCR: "Network Convergence System 1014 FAN Module"
PID: NCS1K14-FAN       , VID: V00, SN: FCB2704B0P7
MFG_NAME: Cisco Systems, Inc., SNMP_IDX: 28673     , Type: Fantray
PN: 800-111210-01, HW Ver: 0.1

NAME: "0/PM0", DESCR: "Network Convergence System 1004 AC Power Supply Unit
2.5KW"
PID: NCS1K4-AC-PSU-2   , VID: V00, SN: POG26300N0V
MFG_NAME: Cisco Systems, Inc., SNMP_IDX: 20481     , Type: Power Supply
PN: 341-101364-01, HW Ver: 0.1

NAME: "0/0/NXR0", DESCR: "NCS 1014 2.4T Licensed card-Ver2"
PID: NCS1K14-2.4T-A-K9 , VID: V00, SN: FCB2921B2BM
MFG_NAME: Cisco Systems, Inc., SNMP_IDX: 1         , Type: Module
PN: 800-52136-01, HW Ver: 0.1
RP/0/RP0/CPU0:ios#
```

show inventory all

The following sample output shows the inventory information for all the physical entities in the chassis, including line cards and pluggable optics modules.

```
RP/0/RP0/CPU0:ios#show inventory all
Mon Nov 27 11:01:53.452 UTC
```

```
NAME: "Rack 0", DESCR: "Network Convergence System 1014 chassis with timing
support"
PID: NCS1014           , VID: V00, SN: FCB2717B13C

NAME: "0/RP0/CPU0", DESCR: "Network Convergence System 1014 Controller"
PID: NCS1K14-CNTLR-K9  , VID: V00, SN: FCB2723B0CX

NAME: "0/0/NXR0", DESCR: "Network Convergence System 1014 Filler"
PID: NCS1K14-BLANK     , VID: V01, SN: N/A

NAME: "0/1/NXR0", DESCR: "Network Convergence System 1014 2.4T Line Card"
PID: NCS1K14-2.4T-K9   , VID: V00, SN: FCB2710B0L5

NAME: "Optics0/1/0/0", DESCR: "Cisco CIM8 C K9 Pluggable Optics Module"
PID: CIM8-C-K9         , VID: VES1, SN: SIM-AX12-SW

NAME: "Optics0/1/0/1", DESCR: "Cisco 100G QSFP28 SR4-S Pluggable Optics Module"
PID: QSFP-100G-SR4-S   , VID: ES1 , SN: AVF1933G18C
RP/0/RP0/CPU0:ios#
```

show inventory location

The following sample output shows the inventory information for a specific line card location, including all pluggable optics modules installed on that line card.

```
RP/0/RP0/CPU0:ios#show inventory location all
Thu Sep 10 18:34:00.350 IST

NAME: "Rack 0", DESCR: "Network Convergence System 1014 chassis with timing
support"
PID: NCS1014           , VID: V02 , SN: FCB2832B0PN

NAME: "0/RP0/CPU0", DESCR: "Network Convergence System 1014 Controller"
PID: NCS1K14-CNTLR-K9  , VID: V01, SN: FCB2831B0RE

NAME: "0/FT0", DESCR: "Network Convergence System 1014 FAN Module"
PID: NCS1K14-FAN       , VID: V01, SN: FCB2834B2HT

NAME: "0/FT1", DESCR: "Network Convergence System 1014 FAN Module"
PID: NCS1K14-FAN       , VID: V01, SN: FCB2834B2LC

NAME: "0/FT2", DESCR: "Network Convergence System 1014 FAN Module"
PID: NCS1K14-FAN       , VID: V01, SN: FCB2834B2L4

NAME: "0/PM0", DESCR: "Network Convergence System 1004 AC Power Supply Unit
2.5KW"
PID: NCS1K4-AC-PSU-2   , VID: V01, SN: POG28010N6A

NAME: "0/PM1", DESCR: "Network Convergence System 1004 AC Power Supply Unit
2.5KW"
PID: NCS1K4-AC-PSU-2   , VID: V00, SN: POG25510N0H

NAME: "0/0/NXR0", DESCR: "NCS 1014 2.4T Line Card-crossconnect modes"
PID: NCS1K14-2.4T-X-K9 , VID: V01, SN: FCB2805B2BT

NAME: "0/1/NXR0", DESCR: "NCS 1014 2.4T Licensed card-Ver2"
PID: NCS1K14-2.4TAL-K9 , VID: V00, SN: FCB2943B2RP

NAME: "0/2/NXR0", DESCR: "Network Convergence System 1014 2.4T Line Card"
```

```
PID: NCS1K14-2.4T-K9   , VID: V00, SN: FCB2726B06K

NAME: "0/3/NXR0", DESCR: "NCS1004 3.2T QSFP-DD DCO Transponder"
PID: NCS1K4-QXP-K9     , VID: V00, SN: FCB2551B1AQ

NAME: "Optics0/0/0/0", DESCR: "Cisco CIM8 C K9 Pluggable Optics Module"
PID: CIM8-C-K9         , VID: V02, SN: ACA291500SE

NAME: "Optics0/0/0/1", DESCR: "Cisco QSFP DD 400G DR4 S Pluggable Optics Module"
PID: QDD-400G-DR4-S    , VID: V02, SN: INL26250QEC

NAME: "Optics0/0/0/4", DESCR: "Cisco QSFP DD 4X100G FR S Pluggable Optics
Module"
PID: QDD-4X100G-FR-S   , VID: V01, SN: INL2529A4GR

NAME: "Optics0/0/0/7", DESCR: "Cisco CIM8 enhanced C-Band Pluggable Optics
Module"
PID: CIM8-CE-K9        , VID: VES1, SN: N/A

NAME: "Optics0/1/0/0", DESCR: "Cisco CIM8 C K9 Pluggable Optics Module"
PID: CIM8-C-K9         , VID: V02, SN: ACA2909004N

NAME: "Optics0/1/0/3", DESCR: "Cisco QSFP DD PSM Pluggable Optics Module"
PID: ONS-QSFP-PSM2     , VID: VES2, SN: ACW3014YL27

NAME: "Optics0/1/0/4", DESCR: "Cisco QSFP DD 400G DR4 S Pluggable Optics Module"
PID: QDD-400G-DR4-S    , VID: V02, SN: INL280303NZ

NAME: "Optics0/1/0/7", DESCR: "Cisco CIM8 enhanced L-Band Pluggable Optics
Module"
PID: CIM8-LE-K9        , VID: VES1, SN: ACA282500FA

NAME: "Optics0/2/0/0", DESCR: "Cisco CIM8 C K9 Pluggable Optics Module"
PID: CIM8-C-K9         , VID: VES1, SN: ACA274900AQ

NAME: "Optics0/2/0/1", DESCR: "Cisco QSFP DD 400G AOC 3M Pluggable Optics
Module"
PID: QDD-400-AOC3M     , VID: V01, SN: INL2748042E-A

NAME: "Optics0/2/0/6", DESCR: "Cisco QSFP DD 400G LR8 S Pluggable Optics Module"
PID: QDD-400G-LR8-S    , VID: V01, SN: FNS26260LXE

NAME: "Optics0/2/0/7", DESCR: "Cisco CIM8 C K9 Pluggable Optics Module"
PID: CIM8-C-K9         , VID: VES1, SN: ACA280600NP

NAME: "Optics0/3/0/5", DESCR: "Cisco QSFP DD PSM Pluggable Optics Module"
PID: ONS-QSFP-PSM2     , VID: VES2, SN: ACW3014YL09

NAME: "Optics0/3/0/14", DESCR: "Cisco QSFP DD HK9 Pluggable Optics Module"
PID: DP04QSDD-HK9      , VID: ES01, SN: ACA265101NQ

NAME: "Optics0/3/0/15", DESCR: "Cisco QSFP DD 400G DR4 S Pluggable Optics
Module"
PID: QDD-400G-DR4-S    , VID: V02, SN: INL27320001
RP/0/RP0/CPU0:ios#
```

show inventory chassis

The following sample output shows the chassis-only inventory information.

```
RP/0/RP0/CPU0:ios#show inventory chassis
Mon Nov 27 11:02:05.083 UTC

NAME: "Rack 0", DESCR: "Network Convergence System 1014 chassis with timing
support"
PID: NCS1014           , VID: V00, SN: FCB2717B13C
RP/0/RP0/CPU0:ios#
```

show inventory fan

The following sample output shows the inventory information for all the FAN modules installed on the chassis.

```
RP/0/RP0/CPU0:ios#show inventory fan
Mon Nov 27 11:02:39.811 UTC

NAME: "0/FT0", DESCR: "Network Convergence System 1014 FAN Module"
PID: NCS1K14-FAN       , VID: V00, SN: FCB2720B192

NAME: "0/FT1", DESCR: "Network Convergence System 1014 FAN Module"
PID: NCS1K14-FAN       , VID: V00, SN: FCB2720B197

NAME: "0/FT2", DESCR: "Network Convergence System 1014 FAN Module"
PID: NCS1K14-FAN       , VID: V00, SN: FCB2720B19U
RP/0/RP0/CPU0:ios#
```

show inventory power

The following sample output shows the inventory information for the power supply units (PSU) installed on the chassis.

```
RP/0/RP0/CPU0:ios#sh inventory power
Mon Nov 27 11:03:04.566 UTC

NAME: "0/PM0", DESCR: "Network Convergence System 1004 AC Power Supply Unit"
PID: NCS1K4-AC-PSU     , VID: V01, SN: POG2727CLP6
```

Verify hardware components status

Verify the operational state of every hardware component on Cisco NCS 1014 by running the **show platform** and **show inventory** commands.

Use this task to confirm that every hardware component that is installed on Cisco NCS 1014 is operational, including the controller, the power supply units (PSU), the FAN modules, the line cards, and the pluggable optics.

The **show platform** command displays the operational state of each node, and the **show inventory** command displays the product identifier (PID), version identifier (VID), and serial number (SN) of each physical entity along with the details of the QSFP pluggables.

Before you begin

Ensure that all the required hardware components are installed on Cisco NCS 1014. For installation details, see the *Cisco Network Convergence System 1014 Hardware Installation Guide*.

Procedure

1. Run the **show platform** command in Cisco IOS XR EXEC mode, to display the operational state of each node and the configuration state of each node.

   Example

```
RP/0/RP0/CPU0:ios#show platform
Node               Type                         State                     Config
state
--------------------------------------------------------------------------------
0/RP0/CPU0       NCS1K14-CNTLR-K9(Active) IOS XR RUN              NSHUT,NMON
0/PM0            NCS1K4-AC-PSU            OPERATIONAL             NSHUT,NMON
0/PM1            NCS1K4-AC-PSU            OPERATIONAL             NSHUT,NMON
0/FT0            NCS1K14-FAN              OPERATIONAL             NSHUT,NMON
0/FT1            NCS1K14-FAN              OPERATIONAL             NSHUT,NMON
0/FT2            NCS1K14-FAN              OPERATIONAL             NSHUT,NMON
0/0/NXR0         NCS1K4-1.2T-K9           OPERATIONAL             NSHUT,NMON
0/1/NXR0         NCS1K14-2.4T-K9          OPERATIONAL             NSHUT,NMON
0/2/NXR0         NCS1K4-1.2T-K9           OPERATIONAL             NSHUT,NMON
0/3/NXR0         NCS1K4-1.2T-K9           OPERATIONAL             NSHUT,NMON
```

```
RP/0/RP0/CPU0:RINode1#show platform
Mon Feb 24 14:13:59.742 IST
Node               Type                         State                     Config
state
--------------------------------------------------------------------------------
0/RP0/CPU0       NCS1K14-CNTLR-K9(Active) IOS XR RUN              NSHUT,NMON
0/PM0            NCS1K4-AC-PSU-2          OPERATIONAL             NSHUT,NMON
0/FT0            NCS1K14-FAN              OPERATIONAL             NSHUT,NMON
0/FT1            NCS1K14-FAN              OPERATIONAL             NSHUT,NMON
0/FT2            NCS1K14-FAN              OPERATIONAL             NSHUT,NMON
0/0/NXR0         NCS1K14-EDFA2            OPERATIONAL             NSHUT,NMON
0/1/NXR0         NCS1K14-BLANK            PRESENT                 NSHUT,NMON
0/2/NXR0         NCS1K14-BLANK            PRESENT                 NSHUT,NMON
0/3/NXR0         NCS1K4-1.2T-K9           OPERATIONAL             NSHUT,NMON
0/4              NCS1K-MD-32O-CE          OPERATIONAL             NSHUT,NMON
0/5              NCS1K-MD-32E-CE          OPERATIONAL             NSHUT,NMON
```

2. Run the **show inventory** command to display the details of each physical entity along with the details of the QSFP pluggables.

   Example

```
RP/0/RP0/CPU0:ios#show inventory

NAME: "Rack 0", DESCR: "Network Convergence System 1014 chassis with timing
support"
PID: NCS1014           , VID: V00, SN: FCB2726B0AR

NAME: "0/RP0/CPU0", DESCR: "Network Convergence System 1014 Controller"
PID: NCS1K14-CNTLR-K9  , VID: V00, SN: FCB2726B0LR

NAME: "0/0/NXR0", DESCR: "NCS1K4 12x QSFP28 2 Trunk C-Band DWDM card"
PID: NCS1K4-1.2T-K9    , VID: V00, SN: CAT2250B0C4

NAME: "Optics0/0/0/9", DESCR: "Cisco 100G QSFP28 FR-S Pluggable Optics Module"
```

```
PID: QSFP-100G-FR-S    , VID: V02 , SN: FBN2638A153

NAME: "Optics0/0/0/10", DESCR: "Cisco 100G QSFP28 FR-S Pluggable Optics Module"
PID: QSFP-100G-FR-S    , VID: V02 , SN: FBN2638A6J0

NAME: "Optics0/0/0/11", DESCR: "Cisco 100G QSFP28 FR-S Pluggable Optics Module"
PID: QSFP-100G-FR-S    , VID: V02 , SN: FBN2638A73V

NAME: "Optics0/0/0/12", DESCR: "Cisco 100G QSFP28 FR-S Pluggable Optics Module"
PID: QSFP-100G-FR-S    , VID: V02 , SN: FBN2544A687

NAME: "Optics0/0/0/13", DESCR: "Cisco 100G QSFP28 FR-S Pluggable Optics Module"
PID: QSFP-100G-FR-S    , VID: V02 , SN: FBN2638A2D4

NAME: "Optics0/0/0/2", DESCR: "Cisco 100G QSFP28 LR4 Pluggable Optics Module"
PID: ONS-QSFP28-LR4    , VID: V01 , SN: FNS2333080H

NAME: "Optics0/0/0/3", DESCR: "Cisco 100G QSFP28 FR-S Pluggable Optics Module"
PID: QSFP-100G-FR-S    , VID: V02 , SN: FBN2638A6J2

NAME: "Optics0/0/0/4", DESCR: "Cisco 100G QSFP28 FR-S Pluggable Optics Module"
PID: QSFP-100G-FR-S    , VID: V01 , SN: INL23243050

NAME: "Optics0/0/0/5", DESCR: "Cisco 100G QSFP28 LR4 Pluggable Optics Module"
PID: ONS-QSFP28-LR4    , VID: V01 , SN: FNS23320DZF

NAME: "Optics0/0/0/6", DESCR: "Cisco 100G QSFP28 LR4 Pluggable Optics Module"
PID: ONS-QSFP28-LR4    , VID: V01 , SN: FNS26140JRK

NAME: "Optics0/0/0/7", DESCR: "Cisco 100G QSFP28 LR4-S Pluggable Optics Module"
PID: QSFP-100G-LR4-S   , VID: V02 , SN: FNS263509QL

NAME: "Optics0/0/0/8", DESCR: "Cisco 100G QSFP28 FR-S Pluggable Optics Module"
PID: QSFP-100G-FR-S    , VID: V02 , SN: FBN2638A5G5

NAME: "0/1/NXR0", DESCR: "Network Convergence System 1014 2.4T Line Card"
PID: NCS1K14-2.4T-K9   , VID: V00, SN: FCB2726B072
```

Verify the software version

Verify the Cisco IOS XR software version that is installed on Cisco NCS 1014 by running the **show version** command.

Use this task to confirm the version of the Cisco IOS XR software that is preinstalled on Cisco NCS 1014.

Cisco NCS 1014 is shipped with the Cisco IOS XR software preinstalled. Verifying the software version helps confirm that your device is running the latest and most stable software.

Procedure

Run the **show version** command to display the software version, the build information, and details such as the system uptime.

Example

```
RP/0/RP0/CPU0:ios#show version
Cisco IOS XR Software, Version 7.11.1.49I LNT
```

```
Copyright (c) 2013-2023 by Cisco Systems, Inc.

Build Information:
 Built By     : sajshah
 Built On     : Sun Nov 19 20:31:06 UTC 2023
 Build Host   : iox-ucs-077
 Workspace    :
/auto/ioxdepot6/GISO/giso_build_lindt/giso_eng_create/yshivapp_2023-11-20_04-28-49_UTC

 Version      : 7.11.1.49I
 Label        : 7.11.1.49I-Weekly

cisco NCS1010 (C3758R @ 2.40GHz)
cisco NCS1014 (C3758R @ 2.40GHz) processor with 32GB of memory
KEPLER_PF6 uptime is 1 hour, 40 minutes
NCS 1014 - Chassis
```

```
RP/0/RP0/CPU0:sanity_176#show version
Cisco IOS XR Software, Version 25.1.1.34I LNT
Copyright (c) 2013-2024 by Cisco Systems, Inc.
Build Information:
Built By : cisco
Built On : Sun Jan 26 23:09:31 UTC 2025
Build Host : iox-ucs-037
Workspace : /auto/iox-ucs-037-san2/prod/25.1.1.34I.SIT_IMAGE/ncs1010/ws/
Version : 25.1.1.34I
Label : 25.1.1.34I
cisco NCS1010 (C3758R @ 2.40GHz)
cisco NCS1014 (C3758R @ 2.40GHz) processor with 32GB of memory
sanity_176 uptime is 3 days, 23 hours, 59 minutes
NCS 1014 - Chassis
```

Verify environmental parameters

Verify the environmental parameters of the Cisco NCS 1014 chassis, including fan speed, power, temperature, voltage, and current.

Use this task to ensure the Cisco NCS 1014 operates within safe environmental limits by checking real-time metrics.

Monitoring environmental parameters helps you proactively detect potential issues with power, cooling, or component health.

show environment all altitude fan power voltage current temperature location *location*

The following keywords are available with the **show environment** command:

Before you begin

Log in to the Cisco NCS 1014 console in Cisco IOS XR EXEC mode.

Procedure

1. Run the **show environment** command with the keyword that matches the environmental parameter that you want to retrieve.

   To view specific parameters, append the relevant keyword, such as `fan`, `power`, `voltage`, `current`, or `temperature`

- **all**: Displays all environmental parameters in a single output.
- **altitude**: Displays the altitude information of the chassis.
- **fan**: Displays the fan speed of the FAN modules and the power supply units (PSU).
- **power**: Displays the power consumption details of the chassis and its modules.
- **voltage**: Displays the voltage values of the voltage sensors.
- **current**: Displays the current values of the current sensors.
- **temperature**: Displays the temperature values of the temperature sensors.
- **location** *location*: Restricts the output to the specified node location.

> **Note**
>
> From Release 25.4.1, the **show environment all temperature location** *lc-location* command displays the **TEMP_MODULE** sensor to indicate the temperature of the pluggable modules. **!TEMP_MODULE** indicates that the temperature has crossed its threshold limit and that an alarm is raised for this sensor.

For sample command outputs, refer to the Sample outputs of the show environment command.

2. (Optional) Restrict the output to a specific node by appending the **location** *location* keyword to the command.

Example

```
RP/0/RP0/CPU0:ios#show environment temperature location 0/RP0/CPU0
```

You have verified the environmental parameters for the Cisco NCS 1014, confirming that the device is operating within safe limits.

Sample outputs of the show environment command

Lists the sample outputs of the **show environment** command for each keyword variant supported on Cisco NCS 1014.

The following sections show representative outputs for each keyword variant of the **show environment** command. Use these samples together with the Verify environmental parameters task.

show environment fan

The following sample output shows the fan speed, in revolutions per minute (RPM), of each FAN module and PSU.

```
RP/0/RP0/CPU0:RINode1#show environment fan
Mon Feb 24 14:15:40.158 IST
==============================================================
                                              Fan speed (rpm)
Location      FRU Type                        FAN_0    FAN_1
--------------------------------------------------------------

0/PM0         NCS1K4-AC-PSU-2                   5696    5696
0/FT0         NCS1K14-FAN                       7560    6780
```

```
0/FT1        NCS1K14-FAN                          7560    6720
0/FT2        NCS1K14-FAN                          7560    6780
```

show environment power

The following sample output shows the power allocated and used for a specific line card.

```
RP/0/RP0/CPU0:ios#show environment power location 0/1/NXR0
Mon Jan 12 12:08:20.031 UTC
==================================================
   Location     Card Type               Power       Power        Status
                                        Allocated   Used
                                        Watts       Watts
==================================================
   0/1/NXR0     NCS1K14-2.4T-A-K9       460         31           ON
RP/0/RP0/CPU0:235#
```

The following sample output shows the chassis-level power summary for all groups, modules, and cards.

```
RP/0/RP0/CPU0:ios#show environment power
Mon Feb 24 14:16:12.970 IST
================================================================================
CHASSIS LEVEL POWER INFO: 0
================================================================================

   Total output power capacity (Group 0 + Group 1) :    2500W +        0W
   Total output power required                     :    1136W
   Total power input                               :     368W
   Total power output                              :     304W

Power Group 0:
================================================================================

   Power       Supply          ------Input----   ------Output---     Status
   Module      Type             Volts     Amps    Volts     Amps

================================================================================

   0/PM0       NCS1K4-AC-PSU-2 230.5     1.6     12.1      25.2     OK

Total of Group 0:              368W/1.6A         304W/25.2A

================================================================================

   Location     Card Type               Power       Power        Status
                                        Allocated   Used
                                        Watts       Watts
================================================================================

   0/RP0/CPU0   NCS1K14-CNTLR-K9        73          24           ON
   0/FT0        NCS1K14-FAN             170         26           ON
   0/FT1        NCS1K14-FAN             170         27           ON
   0/FT2        NCS1K14-FAN             170         27           ON
   0/0/NXR0     NCS1K14-EDFA2           220         56           ON
   0/3/NXR0     NCS1K4-1.2T-K9          260         112          ON
   0/Rack       NCS1014                 73          15           ON
```

show environment temperature

The following sample output shows the temperature sensor readings for a specific line card.

```
RP/0/RP0/CPU0:ios#show environment temperature location 0/1/NXR0
Mon Jan 12 12:05:12.173 UTC
==============================================================
Location  TEMPERATURE                          Value     Crit    Major    Minor
    Minor    Major    Crit
          Sensor                               (deg C)     (Lo)     (Lo)
(Lo)     (Hi)     (Hi)    (Hi)
--------------------------------------------------------------------------------------------
0/1/NXR0
          TEMP_DX_PCB_0                          34      -10       -5
0       80       85       95
          TEMP_DX_ZYNQ                           35      -10       -5
0       90       95      100
          TEMP_DX_PCB_1                          29      -10       -5
0       80       85       95
          TEMP_DX_META0_DIE                      38      -10       -5
0       95      100      105
          TEMP_DX_PCB_2                          30      -10       -5
0       80       85       95
          TEMP_DX_META1_DIE                      35      -10       -5
0       95      100      105
          TEMP_DX_META0_0                        31      -10       -5
0      115      120      125
          TEMP_DX_META0_1                        31      -10       -5
0      115      120      125
          TEMP_DX_META0_2                        31      -10       -5
0      115      120      125
          TEMP_DX_META0_3                        30      -10       -5
0      115      120      125
          TEMP_DX_META1_0                        33      -10       -5
0      115      120      125
          TEMP_DX_META1_1                        34      -10       -5
0      115      120      125
          TEMP_DX_META1_2                        33      -10       -5
0      115      120      125
          TEMP_DX_META1_3                        33      -10       -5
0      115      120      125
```

The following sample output shows the temperature sensor readings for the route processor (RP).

Note

From Release 25.4.1, the **show environment all temperature location** *lc-location* command displays the **TEMP_MODULE** sensor to indicate the temperature of the pluggable modules. **!TEMP_MODULE** indicates that the temperature has crossed its threshold limit and that an alarm is raised for this sensor.

```
RP/0/RP0/CPU0:ios#show environment temperature location 0/2
Wed Dec 17 10:09:48.417 UTC
================================================================================
Location  TEMPERATURE                            Value      Crit     Major
Minor     Minor     Major     Crit
           Sensor                               (deg C)      (Lo)      (Lo)
(Lo)      (Hi)      (Hi)      (Hi)
--------------------------------------------------------------------------------
0/2/NXR0
          TEMP_DX_PCB_0                           28       -10       -5
  0       80        85        95
.
.
.
output snipped
.
.
.
          TEMP_DX_PCB_3                           24       -10       -5
  0       80        85        95
          TEMP_DX_META2_DIE                       26       -10       -5
  0       95        100       105
Optics0/2/0/0
          TEMP_MODULE                             26       -10        0
  5       95        100       105
Optics0/2/0/7
          TEMP_MODULE                             24       -10        0
  5       95        100       105
Optics0/2/0/1
          TEMP_MODULE                             25        -5        0
  5       65        70        75
Optics0/2/0/2
          TEMP_MODULE                             23        -5        0
  5       65        70        75
Optics0/2/0/3
          TEMP_MODULE                             21        -5        0
  5       65        70        75
Optics0/2/0/4
          TEMP_MODULE                             23        -5        0
  5       65        70        75
```

show environment voltage

The following sample output shows the voltage sensor readings for a specific line card.

```
RP/0/RP0/CPU0:ios#show environment voltage location 0/1/NXR0
Mon Jan 12 11:52:56.416 UTC
================================================================
Location  VOLTAGE                          Value      Crit    Minor    Minor
     Crit
          Sensor                           (mV)       (Lo)    (Lo)     (Hi)
```

```
      (Hi)
-------------------------------------------------------------------------------
0/1/NXR0
          VMON_3_3V                          3303     3135     3201     3399
     3465
          VMON_1_8V_PS_MGT                   1801     1710     1746     1854
     1890
          VMON_1_8V                          1793     1710     1746     1854
     1890
          VMON_1_2V_PS_MGT                   1202     1140     1164     1236
     1260
          VMON_1_2V                          1202     1140     1164     1236
     1260
          VMON_2_5V                          2503     2375     2425     2575
     2625
          VMON_0_85V_PS_MGT                   852      808      825      876
      893
          VMON_0_90V_PS_MGT                   901      855      873      927
      945
          VMON_0_85V_PS                       845      808      825      876
      893
          META0_PGOOD                        3003     2400     2700     3300
     3600
          META1_PGOOD                        3003     2400     2700     3300
     3600
          VMON_1_05V_PEX                     1054      998     1019     1082
     1103
          META_1_8V                          1803     1710     1746     1854
     1890
          VIN_7_0V_ST                        7029     6650     6790     7210
     7350
          VAUX_3_3V_ST                       3304     3135     3201     3399
     3465
          AVD_1_2V_META0                     1210     1140     1164     1236
     1260
          AVD_1_2V_META1                     1211     1140     1164     1236
     1260
          VIN_0_META0_12V                   12062     9600    10800    12600
    12840
          VOUT_0_META0_CORE                   760      724      739      785
      800
          VOUT_1_META0_CORE                   760      724      739      785
      800
          VIN_2_META0_12V                   12062     9600    10800    12600
    12840
          VOUT_2_META0_CORE                   761      724      739      785
      800
          VOUT_AVD_META0                      754      713      728      773
      788
          VIN_0_META1_12V                   12078     9600    10800    12600
    12840
          VOUT_0_META1_CORE                   761      724      739      785
      800
          VOUT_1_META1_CORE                   761      724      739      785
      800
          VIN_2_META1_12V                   12046     9600    10800    12600
    12840
          VOUT_2_META1_CORE                   761      724      739      785
      800
          VOUT_AVD_META1                      753      713      728      773
      788
```

The following sample output shows the voltage sensor readings for the route processor (RP).

show environment current

The following sample output shows the current sensor readings for a specific line card.

```
RP/0/RP0/CPU0:ios#show environment current location 0/1/NXR0
Mon Jan 12 11:54:56.298 UTC
==========================================
Location  CURRENT                                 Value
          Sensor                                  (mA)
-------------------------------------------------------------------------------
0/1/NXR0
          IMON_CTLPL                             1042
          IMON_CLI                                  0
          IMON_META0_IN0                          255
          IMON_META0_CORE_IOUT0                  2421
          IMON_META0_CORE_IOUT1                   975
          IMON_META0_IN2                          354
          IMON_META0_CORE_IOUT2                  2335
          IMON_META0_AVD_IOUT                    2718
          IMON_META1_IN0                          101
          IMON_META1_CORE_IOUT0                  1080
          IMON_META1_CORE_IOUT1                   267
          IMON_META1_IN2                          155
          IMON_META1_CORE_IOUT2                   681
          IMON_META1_AVD_IOUT                    1300
```

The following sample output shows the chassis-wide current sensor readings.

show environment all

The following sample output shows the consolidated environmental information that includes temperature, voltage, current, and power readings for the chassis, route processor, line cards, FAN modules, and PSUs.

```
RP/0/RP0/CPU0:sanity_176#sh environment all
=============================================================================================
Location  TEMPERATURE                             Value     Crit    Major    Minor
    Minor    Major    Crit
          Sensor                                 (deg C)     (Lo)     (Lo)
(Lo)     (Hi)     (Hi)     (Hi)
---------------------------------------------------------------------------------------------
0/RP0/CPU0
          RP_TEMP_PCB                               31      -10       -5
0       80       85       90
          RP_TEMP_HOT_SPOT                          32      -10       -5
0       80       85       90
          RP_TEMP_LTM4638_0                         32      -10       -5
0       85       90       95
          RP_TEMP_LTM4644_0                         30      -10       -5
0       85       90       95
          RP_TEMP_LTM4644_1                         32      -10       -5
0       85       90       95
          RP_TEMP_LTM4638_1                         31      -10       -5
0       80       90       95
          RP_TEMP_LTM4644_2                         30      -10       -5
0       85       90       95
          RP_TEMP_LTM4638_2                         31      -10       -5
0       80       90       95
          TEMP_CPU_DIE                              36      -10       -5
0       80       85       90
          TEMP_DDR_DIMM                             33      -10       -5
0       80       85       90
          TEMP_CPU_SSD                              44      -10       -5
```

```
0       70       75       80
          TEMP_CHASSIS_SSD                        31      -10       -5
0       70       75       80
0/PM0
          Air Inlet Temperature                   30      -10       -5
0       62       67       72
          Air Outlet Temperature                  32      -10       -5
0       82       87       92
          Oring MOSFET                            35      -10       -5
0       77       82       87
0/PM1
          Air Inlet Temperature                   31      -10       -5
0       62       67       72
          Air Outlet Temperature                  32      -10       -5
0       82       87       92
          Oring MOSFET                            34      -10       -5
0       77       82       87
0/1/NXR0
          TEMP_DX_PCB                              -      -10       -5
0       95      100      105
          TEMP_DX_ZYNQ                             -      -10       -5
0       95      100      105
          SA_TEMP_OUTLET_LC                        -      -10       -5
0       75       85       90
0/3/NXR0
          TEMP_DX_PCB                             24      -10       -8
-5       85       90       95
          TEMP_DX_ZYNQ                            27      -10       -8
-5       95       98      100
          SA_TEMP_OUTLET_LC                       25      -10       -5
0       75       85       90
0/Rack
          SA_TEMP_CHASSIS_INLET0                  21      -10       -5
0       45       55       60
          SA_TEMP_CHASSIS_INLET1                  21      -10       -5
0       45       55       60
          SA_TEMP_LTM4647_HOTSPOT                 32      -10       -5
0       95      105      110
          SA_TEMP_ALDRIN_HOTSPOT                  26      -10       -5
0       80       85       90
          SA_TEMP_INLET_PSU0                      30      -10       -5
0       80       85       90
          SA_TEMP_INLET_PSU1                      30      -10       -5
0       80       85       90
          SA_TEMP_INLET_FAN0                      25      -10       -5
0       80       85       90
          SA_TEMP_INLET_FAN1                      20      -10       -5
0       80       85       90
==============================================================================
Location  VOLTAGE                            Value      Crit    Minor    Minor
     Crit
          Sensor                              (mV)       (Lo)    (Lo)     (Hi)
      (Hi)
------------------------------------------------------------------------------
0/RP0/CPU0
          RP_ADM1266_12V0                    12050    10800    11280    12720
    13200
          RP_ADM1266_3V3_STAND_BY             3304     3070     3200     3400
     3530
          RP_ADM1266_5V0                      5056     4650     4850     5150
     5350
          RP_ADM1266_3V3                      3334     3070     3200     3400
```

```
    3530
         RP_ADM1266_2V5_PLL                  2515     2330     2430     2580
    2680
         RP_ADM1266_2V5_FPGA                 2511     2330     2430     2580
    2680
         RP_ADM1266_1V2_FPGA                 1202     1120     1160     1240
    1280
         RP_ADM1266_3V3_CPU                   3332     307
```

Verify management interface status

Verify the configuration and operational status of the management Ethernet interface on Cisco NCS 1014 using the appropriate show commands.

Use this task to confirm that the management Ethernet interface of Cisco NCS 1014 is up and to view its configuration and traffic counters.

Checking the management Ethernet interface status helps ensure the device is reachable for management and operating as expected. You can also view detailed configuration and traffic counters during troubleshooting.

Procedure

1. Run the **show interfaces mgmtEth** *instance* command to display the configuration of the management Ethernet interface, including the IP address, the MTU, the bandwidth, the duplex mode, and the traffic counters.

   Example

```
RP/0/RP0/CPU0:ios#show interfaces MgmtEth 0/RP0/CPU0/0
MgmtEth0/RP0/CPU0/0 is up, line protocol is up
  Interface state transitions: 3
  Hardware is Management Ethernet, address is 4014.82ba.d26e (bia
4014.82ba.d26e)
  Internet address is 10.105.57.37/25
  MTU 1514 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation ARPA,
  Full-duplex, 1000Mb/s, CX, link type is autonegotiation
  loopback not set,
  Last link flapped 00:09:12
  ARP type ARPA, ARP timeout 04:00:00
  Last input 00:00:00, output 00:00:00
  Last clearing of "show interface" counters never
  5 minute input rate 1000 bits/sec, 2 packets/sec
  5 minute output rate 5000 bits/sec, 1 packets/sec
     6715 packets input, 640515 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 2213 broadcast packets, 4430 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     944 packets output, 355004 bytes, 0 total output drops
     Output 94 broadcast packets, 114 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     3 carrier transitions
```

```
RP/0/RP0/CPU0:ios#show interfaces MgmtEth 0/RP0/CPU0/0
Mon Feb 24 14:25:17.125 IST
```

```
MgmtEth0/RP0/CPU0/0 is up, line protocol is up
  Interface state transitions: 1
  Hardware is Management Ethernet, address is 9c38.1888.a418 (bia
9c38.1888.a418)
  Description: mgmt0
  Internet address is 10.127.126.174/27
  MTU 1514 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation ARPA,
  Full-duplex, 1000Mb/s, 1000BASE-T, link type is autonegotiation
  loopback not set,
  Last link flapped 2d23h
  ARP type ARPA, ARP timeout 04:00:00
  Last input 00:00:00, output 00:04:50
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 1 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     202558 packets input, 14206120 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 64032 broadcast packets, 137089 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     330 packets output, 14428 bytes, 0 total output drops
     Output 0 broadcast packets, 4 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions
```

In the previous output, the management interface is administratively up.

You can also run the **show interfaces summary** and **show interfaces brief** commands in Cisco IOS XR EXEC mode to verify the management interface status.

2. Optionally, run the **show interfaces summary** and **show interfaces brief** commands in Cisco IOS XR EXEC mode to verify the management interface status.

   The following example shows sample output from the **show interfaces summary** command:

```
RP/0/RP0/CPU0:ios#show interfaces summary
Interface Type          Total    UP       Down     Admin Down
--------------          -----    --       ----     ----------
ALL TYPES               5        2        0        3
--------------
IFT_ETHERNET            2        1        0        1
IFT_NULL                1        1        0        0
IFT_PTP_ETHERNET        2        0        0        2
```

```
RP/0/RP0/CPU0:ios#show interfaces summary
Mon Feb 24 14:26:00.615 IST
Interface Type          Total    UP       Down     Admin Down
--------------          -----    --       ----     ----------
ALL TYPES               6        5        0        1
--------------
IFT_GETHERNET           1        1        0        0
IFT_LOOPBACK            1        1        0        0
IFT_ETHERNET            2        2        0        0
IFT_NULL                1        1        0        0
IFT_PTP_ETHERNET        1        0        0        1
```

The following example shows sample output from the **show interfaces brief** command:

```
RP/0/RP0/CPU0:ios#show interfaces brief

                Intf       Intf        LineP              Encap  MTU
BW
                Name       State       State               Type (byte)
(Kbps)
--------------------------------------------------------------------------------

                 Nu0          up          up               Null  1500
  0
    Mg0/RP0/CPU0/0          up          up              ARPA  1514    1000000

    Mg0/RP0/CPU0/1  admin-down  admin-down              ARPA  1514    1000000

    PT0/RP0/CPU0/0  admin-down  admin-down              ARPA  1514    1000000

    PT0/RP0/CPU0/1  admin-down  admin-down              ARPA  1514    1000000
```

```
RP/0/RP0/CPU0:ios#show interfaces brief
Mon Feb 24 14:26:32.302 IST

                Intf       Intf        LineP              Encap  MTU
BW
                Name       State       State               Type (byte)
(Kbps)
--------------------------------------------------------------------------------

                 Lo0          up          up           Loopback  1500
  0
                 Nu0          up          up               Null  1500
  0
        Gi0/0/0/5          up          up              ARPA  1514    1000000

    Mg0/RP0/CPU0/0          up          up              ARPA  1514    1000000

    Mg0/RP0/CPU0/1          up          up              ARPA  1514    1000000

    PT0/RP0/CPU0/1  admin-down  admin-down              ARPA  1514    1000000
```

You can confirm whether the management Ethernet interface is administratively and operationally up and review its configuration and traffic counters.

What to do next

If the management interface is administratively down, perform the following actions:

- Check the Ethernet cable connection.
- Verify the IP configuration of the management interface. For details about configuring the management interface, see Configure a management interface on page 23.
- Verify that the management interface is in the no shut state by running the **show running-config interface mgmtEth** command.

The following example shows sample output from the **show running-config interface mgmtEth** command:

```
RP/0/RP0/CPU0:ios#show running-config interface mgmtEth 0/RP0/CPU0/0
interface MgmtEth0/RP0/CPU0/0
 ipv4 address 10.105.57.37 255.255.255.128
!
```

Verify firmware version

Verify the field-programmable device (FPD) firmware version running on each hardware component of Cisco NCS 1014 using the show hw-module fpd and show fpd package commands.

Use this task to confirm that the firmware that is running on each hardware component of Cisco NCS 1014 is compatible with the installed Cisco IOS XR image.

Cisco NCS 1014 hardware components require compatible firmware for proper operation. Using the **show hw-module fpd** command displays the current firmware version for each FPD, while show fpd package shows the firmware version bundled with the installed Cisco IOS XR software image.

Procedure

1. Run the **show hw-module fpd** command.

> **Note**
>
> From R26.2.1, the solid-state drive (SSD) FPD name that is displayed in the output of the **show hw-module fpd** command changes from a vendor- or model-specific name to a generic format: **CPU-SSD** for RP locations and **CHASSIS-SSD** for rack locations.

Example

```
RP/0/RP0/CPU0:ios#show hw-module fpd
Mon Aug 24 08:37:21.376 UTC

Auto-upgrade:Enabled,PM excluded
Attribute codes: B golden, P protect, S secure, A Anti Theft aware
                                                                                FPD
 Versions

==============
Location   Card type            HWver FPD device       ATR Status    Running
  Programd   Reload Loc
-------------------------------------------------------------------------------
0/RP0/CPU0 NCS1K14-CNTLR-K9      1.0   ADM-DB              CURRENT      2.10
     2.10         NOT REQ
0/RP0/CPU0 NCS1K14-CNTLR-K9      1.0   ADM-MB              CURRENT      2.30
     2.30         NOT REQ
0/RP0/CPU0 NCS1K14-CNTLR-K9      1.0   BIOS            S   CURRENT      6.10
     6.10           0/RP0
0/RP0/CPU0 NCS1K14-CNTLR-K9      1.0   BIOS-Golden      BS  CURRENT
      4.70           0/RP0
0/RP0/CPU0 NCS1K14-CNTLR-K9      1.0   CPU-SSD          S   CURRENT      1.30
```

```
     1.30          0/RP0
0/RP0/CPU0 NCS1K14-CNTLR-K9      1.0   CpuFpga          S   CURRENT      1.17
     1.17          0/RP0
0/RP0/CPU0 NCS1K14-CNTLR-K9      1.0   CpuFpgaGolden    BSP CURRENT
      1.09          0/RP0
0/RP0/CPU0 NCS1K14-CNTLR-K9      1.0   TamFw            S   CURRENT      9.04
     9.04          0/RP0
0/RP0/CPU0 NCS1K14-CNTLR-K9      1.0   TamFwGolden      BSP CURRENT
      9.04          0/RP0
0/PM0      NCS1K4-AC-PSU-2       1.1   PO-PriMCU            CURRENT      1.03
     1.03         NOT REQ
0/PM0      NCS1K4-AC-PSU-2       1.1   PO-SecMCU            CURRENT      1.05
     1.05         NOT REQ
0/PM1      NCS1K4-AC-PSU-2       1.1   PO-PriMCU            CURRENT      1.03
     1.03         NOT REQ
0/PM1      NCS1K4-AC-PSU-2       1.1   PO-SecMCU            CURRENT      1.05
     1.05         NOT REQ
0/0/NXR0   NCS1K14-2.4T-X-K9     1.0   CpuModFw         S   CURRENT    263.100
  263.100       NOT REQ
0/0/NXR0   NCS1K14-2.4T-X-K9      0.0   OpticsFw_Port_0  S   CURRENT
80.15016 80.15016     NOT REQ
0/0/NXR0   NCS1K14-2.4T-X-K9      0.0   OpticsFw_Port_7  S   CURRENT
80.14015 80.14015     NOT REQ
0/2/NXR0   NCS1K14-2.4T-K9       0.1   CpuModFw         S   CURRENT    263.100
  263.100       NOT REQ
0/2/NXR0   NCS1K14-2.4T-K9       0.0   OpticsFw_Port_0  S   CURRENT
80.15016 80.15016     NOT REQ
0/2/NXR0   NCS1K14-2.4T-K9       0.0   OpticsFw_Port_2  S   CURRENT      1.11
     1.11         NOT REQ
0/2/NXR0   NCS1K14-2.4T-K9       0.0   OpticsFw_Port_7  S   CURRENT
80.15016 80.15016     NOT REQ
0/3/NXR0   NCS1K4-QXP-K9         0.2   CpuModFw         S   CURRENT    263.100
  263.100       NOT REQ
0/3/NXR0   NCS1K4-QXP-K9         0.0   OpticsFw_Port_0  S   CURRENT      1.11
     1.11         NOT REQ
0/3/NXR0   NCS1K4-QXP-K9        0.0   OpticsFw_Port_14 S   CURRENT    61.2416
  61.2416       NOT REQ
0/Rack     NCS1014               1.1   ADM-CHASSIS          CURRENT      0.21
     0.21         NOT REQ
0/Rack     NCS1014               1.1   CHASSIS-SSD      S   CURRENT      1.30
     1.30          0/Rack
0/Rack     NCS1014               1.1   IoFpga           S   CURRENT      2.28
     2.28         NOT REQ
0/Rack     NCS1014               1.1   IoFpgaGolden     BS  CURRENT
      1.05         NOT REQ
```

- Status: Upgrade status of the firmware. The status can be one of the following values:
  - CURRENT: The firmware version is the latest version.
  - NOT READY: The firmware of the FPD is not ready for upgrade.
  - NEED UPGD: A newer firmware version is available in the installed image. Cisco recommends that you perform an upgrade.
  - UPGD PREP: The firmware of the FPD is preparing for upgrade.
  - RLOAD REQ: The upgrade is complete, and the card requires a reload.
  - UPGD DONE: The firmware upgrade is successful.
  - UPGD FAIL: The firmware upgrade has failed.

- UPGD SKIP: The upgrade is skipped because the installed firmware version is higher than the version available in the image.
- Running: The current version of the firmware that is running on the FPD.

2. Run the **show fpd package** command to display the FPD image version that is available with the installed software release for each hardware component.

Example

```
RP/0/RP0/CPU0:ios#show fpd package
Fri Sep 11 10:58:07.423 IST

================================
================================================
                                        Field Programmable Device Package

================================================
                                                Req     SW      Min Req   Min
 Req
Card Type            FPD Description           Reload  Ver     SW Ver  Board
 Ver
==================== ========================== ====== ======= ========
=========

-------------------------------------------------------------------------------
NCS1K14-2.4T-A-K9    CIM8CEK9ACA                NO     80.14015  80.14015
0.0
                     CIM8CK9ACA                 NO     80.14015  80.14015
0.0
                     CIM8LEK9ACA                NO     80.14015  80.14015
0.0
                     CIM8LK9ACA                 NO     80.14015  80.14015
0.0
                     CpuModFw                   NO    263.100   263.100
0.0
                     ONSQSFPPSM2ACC             NO      1.11      1.11
0.0
-------------------------------------------------------------------------------
```

What to do next

Upgrade all the FPDs by running the **upgrade hw-module location all fpd all** command in Cisco IOS XR EXEC mode. After the upgrade is complete, the Status column shows **RLOAD REQ** if the software requires a reload.

If a reload is required:

If the FPGA location is 0/RP0, run the **admin hw-module location 0/RP0 reload** command. This command reboots only the CPU. As a result, traffic is not impacted. If the FPGA location is 0/0, run the **admin hw-module location all reload** command. This command reboots the chassis. As a result, traffic is impacted. After the reload is complete, the new FPGA runs the current version.

If the firmware upgrade fails:

If the firmware upgrade fails, run the **show logging** command to view the details and upgrade the firmware again by using the preceding commands.

> **Note**
>
> You can upgrade the firmware version of the power modules only when both the power modules are present and powered on.

Display alarms on Cisco NCS 1014

Verify the alarms active on Cisco NCS 1014 by displaying them with the appropriate show alarms command.

Use this task to review and check active alarms on Cisco NCS 1014 to monitor system health and diagnose issues.

The show alarms command displays current and historical alarm details on Cisco NCS 1014, allowing you to assess the system status in either brief or detailed formats.

The **show alarms** command supports the following syntax:

Procedure

Run the **show alarms brief card location** *location* **active** command to display the active alarms for the card at the specified location in brief format.

Example

```
RP/0/RP0/CPU0:ios#show alarms brief card location 0/RP0/CPU0 active

------------------------------------------------------------------------------------
Active Alarms
------------------------------------------------------------------------------------
Location        Severity     Group            Set Time
Description
------------------------------------------------------------------------------------
0/0             Major        Ethernet         11/21/2023 11:11:35 UTC
HundredGigECtrlr0/0/0/2 - Carrier Loss On The LAN
0/3             Major        Ethernet         11/21/2023 11:11:37 UTC
HundredGigECtrlr0/3/0/2 - Remote Fault
0/0             Major        Ethernet         11/21/2023 11:11:38 UTC
HundredGigECtrlr0/0/0/4 - Local Fault
0/3             Major        Ethernet         11/21/2023 11:11:38 UTC
HundredGigECtrlr0/3/0/3 - Local Fault
0/3             Major        Ethernet         11/21/2023 11:11:38 UTC
HundredGigECtrlr0/3/0/4 - Local Fault
0/0             Major        Ethernet         11/21/2023 11:11:38 UTC
HundredGigECtrlr0/0/0/5 - Local Fault
0/0             Major        Ethernet         11/21/2023 11:11:38 UTC
HundredGigECtrlr0/0/0/6 - Local Fault
0/0             Major        Ethernet         11/21/2023 11:11:38 UTC
HundredGigECtrlr0/0/0/7 - Local Fault
0/3             Major        Ethernet         11/21/2023 11:11:39 UTC
HundredGigECtrlr0/3/0/5 - Local Fault
0/3             Major        Ethernet         11/21/2023 11:11:39 UTC
HundredGigECtrlr0/3/0/6 - Local Fault
0/3             Major        Ethernet         11/21/2023 11:11:38 UTC
HundredGigECtrlr0/3/0/7 - Local Fault
0/0             Major        Ethernet         11/21/2023 11:11:43 UTC
```

```
HundredGigECtrlr0/0/0/3 - Local Fault
0/3             Major        Ethernet         11/21/2023 11:11:57 UTC
HundredGigECtrlr0/3/0/8 - Local Fault
0/3             Major        Ethernet         11/21/2023 11:11:57 UTC
HundredGigECtrlr0/3/0/9 - Local Fault
0/3             Major        Ethernet         11/21/2023 11:11:59 UTC
HundredGigECtrlr0/3/0/12 - Local Fault
0/3             Major        Ethernet         11/21/2023 11:11:59 UTC
HundredGigECtrlr0/3/0/13 - Local Fault
0/0             Major        Ethernet         11/21/2023 11:12:03 UTC
HundredGigECtrlr0/0/0/9 - Local Fault
0/0             Major        Ethernet         11/21/2023 11:12:04 UTC
HundredGigECtrlr0/0/0/8 - Local Fault
0/0             Major        Ethernet         11/21/2023 11:12:04 UTC
HundredGigECtrlr0/0/0/10 - Local Fault
0/0             Major        Ethernet         11/21/2023 11:12:04 UTC
HundredGigECtrlr0/0/0/11 - Local Fault
0/0             Major        Ethernet         11/21/2023 11:12:04 UTC
HundredGigECtrlr0/0/0/12 - Local Fault
0/0             Major        Ethernet         11/21/2023 11:12:04 UTC
HundredGigECtrlr0/0/0/13 - Local Fault
0/3             Major        Ethernet         11/21/2023 11:12:04 UTC
HundredGigECtrlr0/3/0/11 - Local Fault
0/3             Major        Ethernet         11/21/2023 11:12:05 UTC
HundredGigECtrlr0/3/0/10 - Local Fault
```

```
RP/0/RP0/CPU0:RINode1#show alarms brief card location 0/RP0/CPU0 active
Mon Feb 24 14:27:49.802 IST

------------------------------------------------------------------------------------
Active Alarms
------------------------------------------------------------------------------------
Location        Severity     Group            Set Time
Description
------------------------------------------------------------------------------------
0               Major        Environ          02/21/2025 14:29:21 IST    Power
 Group Redundancy lost
0/3             Minor        Environ          02/21/2025 14:29:46 IST
TEMP_DX_PCB: Invalid sensor read error.
0/3             Minor        Environ          02/21/2025 14:29:46 IST
TEMP_DX_ZYNQ: temperature alarm
0/0/NXR0        Minor        Software         02/21/2025 14:29:58 IST
Ots0/0/0/0 - Neighbour not found
0/0             Minor        Controller       02/21/2025 14:29:58 IST
Ots0/0/0/0 - OTDR Reflectance Exceeded Absolute Threshold - Receive Direction
0/RP0/CPU0      Minor        Software         02/21/2025 14:30:30 IST    SW
Upgrade is still allowed as SIA Grace Period is remaining
0/3             Critical     Environ          02/21/2025 14:31:55 IST
VMON_0_85V: low voltage alarm
0/3             Critical     Environ          02/21/2025 14:31:55 IST
VMON_1_2V: low voltage alarm
0/3             Critical     Environ          02/21/2025 14:31:55 IST
VMON_1_8V: low voltage alarm
0/3             Critical     Environ          02/21/2025 14:31:55 IST
VMON_2_5V: low voltage alarm
0/3             Critical     Environ          02/21/2025 14:31:55 IST
VMON_3_3V: low voltage alarm
0/3             Critical     Environ          02/21/2025 14:31:55 IST
VMON_0_85V_MGT: low voltage alarm
0/3             Critical     Environ          02/21/2025 14:31:55 IST
VMON_0_9V_MGT: low voltage alarm
0/3             Critical     Environ          02/21/2025 14:31:55 IST
```

```
VMON_1_2V_MGT: low voltage alarm
0/3             Critical     Environ          02/21/2025 14:31:55 IST
VMON_1_8V_MGT: low voltage alarm
0/3             Critical     Environ          02/21/2025 14:31:55 IST
PGOOD_VTT: low voltage alarm
0/3             Critical     Environ          02/21/2025 14:31:55 IST
VIN_7_0V_ST: low voltage alarm
0/3             Critical     Environ          02/21/2025 14:31:55 IST
VAUX_3_3V_ST: low voltage alarm
0/3             Critical     Environ          02/21/2025 14:31:55 IST
VIN_12V_MOD: low voltage alarm
0/3             Critical     Environ          02/21/2025 14:31:55 IST
VMON_1_15V_CDR: low voltage alarm
```

Note

In the maintenance mode, all the alarms are suppressed and the **show alarms** command does not display the alarm details. Run the **show controllers** *controller-type R/S/I/P* command to view the client and trunk alarms.

The system displays all active alarms for the specified card location. You can use this information to verify the system's operational status and diagnose issues.

Verify core dump context information

Verify the latest core dump context information for any process on Cisco NCS 1014 using the `show context` command.

Use this task to check up to ten recent core dump instances for any process on Cisco NCS 1014 for troubleshooting and analysis.

Core dump context information includes details such as core location, build information, process state, register data, and backtrace. This aids in analyzing system crashes or failures on Cisco NCS 1014.

Procedure

Run the **show context** command to display the core dump context information of any process on the system, up to the last ten instances.

Example

```
RP/0/RP0/CPU0:RINode1#show context
Mon Feb 24 14:28:32.450 IST

Disclaimer: if exception path is not configured, cores may be found in
/misc/disk1/coredumps

node: node0_RP0_CPU0
------------------------------------------------------------------
No context
```

Verify core files

Verify whether any process has generated compressed core dump files on the hard disk of Cisco NCS 1014.

Use this task to confirm if the Cisco NCS 1014 device has generated any core dump files by listing compressed core dumps stored on its hard disk.

The **dir harddisk:/*core.gz** command lists the compressed core dump files located at the root of the hard disk on Cisco NCS 1014. This helps you monitor for unexpected process failures.

Procedure

Run the **dir harddisk:/*core.gz** command to display each core dump file that matches the ***core.gz** pattern under the **harddisk:/** directory.

Example

```
RP/0/RP0/CPU0:ios#dir harddisk:/*core.gz
Wed Dec 6 04:54:16.336 UTC

Directory of harddisk:/*core.gz
2476 -rw-r--r--. 1 8120038 Oct 30 15:08
cma_server_41264.by.6.20231030-150817.node0_RP0_CPU0.502a7.core.gz
```

3 Perform System Upgrade and Install Feature Packages

Topics:

- Software and firmware compatibility matrix
- Upgrade the software on NCS 1014
- FPD upgrades
- Factory reset

Provides details about the procedures and components used for upgrading system software and installing feature packages on the NCS 1014.

This reference topic provides the chapter-level entry point for system upgrade, feature package installation, FPD upgrade, software compatibility, and factory reset topics.

For details about the chapter topics, see these references:

- Installing system upgrades and feature packages
- Software and firmware compatibility matrix
- FPD upgrades
- Factory reset

Software and firmware compatibility matrix

Describes the FPGA and optics firmware versions that are compatible with supported NCS 1014 hardware and Cisco IOS XR software releases.

These tables provide the compatibility of FPGA firmware versions for each hardware type and supported software release.

Table 4: FPGA firmware compatibility

Hardware Type	FPGA	R7.11.1	R24.1.1	R24.2.11	R24.3.1	R24.4.1	R25.1.1	R25.2.1	R25.3.1	R25.4.1	R26.1.1	R26.2.1	R26.3.1
NCS1K14-24T-K9	CpuModFw	234.10	241.10	242.10	243.10	244.100	252100	252100	263100	264100	261100	262100	263100
NCS1K14-24TX-K9	CpuModFw	NA	241.10	242.10	243.10	244.10	251100	252100	263100	264100	261100	262100	263100
NCS1K4-QXP-K9	CpuModFw	NA	241.10	242.10	243.10	244.10	251100	252100	263100	264100	261100	262100	263100
NCS1K14-QDD-16C	CpuModFw	234.10	241.10	242.10	243.10	244.10	251100	252100	263100	264100	261100	262100	263100
	OptModFw	18.03	20.02	20.02	20.02	20.02	20.02	20.02	20.02	264100	20.02	20.04	20.06
NCS1K14-EDFA2	CpuModFw	NA	NA	NA	NA	NA	251100	252100	263100	264100	261100	262100	263100
	OptModFw	NA	NA	NA	NA	NA	2.04	2.04	2.08	2.08	2.10	2.14	2.14
	CpuModFw	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	263100
NCS1014	ADM-CHASSIS	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
	IoFpga	1.10	1.10	1.10	1.19	1.19	1.19	2.26	2.26	1.19	2.26	2.28	2.28
	IoFpgaGolden	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05
	SsdIntelSC2KB	1.20	1.20	1.20	1.20	1.20	1.30	1.30	1.30	1.30	1.30	1.30	1.30

Table 5: FPGA firmware compatibility for NCS1K4-AC-PSU-2 power supply unit

FPGA	R7.11.1	R24.1.1	R24.2.11	R24.3.1	R24.4.1	R25.1.1	R25.2.1	R25.3.1	R25.4.1	R26.1.1	R26.2.1	R26.3.1
PO-PriMCU	1.03	1.03	1.01	1.03	1.03	1.03	1.03	1.03	1.03	1.03	1.03	1.03
PO-SecMCU	1.05	1.05	1.01	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05	1.05

Table 6: FPGA firmware compatibility for NCS1K14-CNTLR-K9 controller card

FPGA	R7.11.1	R24.1.1	R24.2.11	R24.3.1	R24.4.1	R25.1.1	R25.2.1	R25.3.1	R25.4.1	R26.1.1	R26.2.1	R26.3.1
ADM-DB	2.10	2.10	2.10	2.10	2.10	2.10	2.10	2.10	2.10	2.10	2.10	2.10
ADM-MB	2.30	2.30	2.30	2.30	2.30	2.30	2.30	2.10	2.30	2.30	2.30	2.30
BIOS	4.70	4.80	4.80	4.80	4.80	5.00	5.00	5.00	5.80	6.10	6.10	6.20
BIOS-Golden	4.50	4.50	1.72	4.50	4.70	4.70	4.70	4.70	4.70	4.70	4.70	4.70
CpuFpga	1.09	1.09	1.09	1.17	1.17	1.17	1.17	1.17	1.17	1.17	1.17	1.17

CpuFpgaGolden	1.03	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09	1.09
SsdMicron5300	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	1.01	1.01
TamFw	9.04	9.04	9.04	9.04	9.04	9.04	9.04	9.04	9.04	9.04	9.04	9.04
TamFwGolden	9.04	9.04	9.04	9.04	9.04	9.04	9.04	9.04	9.04	9.04	9.04	9.04

Table 7: Optics firmware compatibility

Hardware type	FPGA	1.3 R
NCS1K14-2.4T-X-K9 (CIM8-C-K9)	OpticsFw_Port_0	4 0 3
NCS1K14-2.4TAL-K9 (CIM8-LE-K9)		4 0 3
NCS1K14-2.4T-K9 (CIM8-CE-K9)		4 0 3
NCS1K4-QXP-K9 (QDD-400G-ZRP-S)		6 2.6
NCS1K4-QXP-K9 (DP04QSDD-E26-28B)	OpticsFw_Port_8	8 2.7
NCS1K4-QXP-K9 (DP04QSDD-HE0)	OpticsFw_Port_14	4 2.0 7
NCS1K4-QXP-K9 (DP04QSDD-HK9)	OpticsFw_Port_2	1 8.0 7
NCS1K14-EDFA2 (ONS-QSFP-OTDR)	OpticsFw_Port_6	2 0 . . 1
NCS1K4-QXP-K9 (DP04QSDD-HK9)	OpticsFw_Port_2	1 8.0 7
NCS1K4-2.4T-X-K9 (ONS-QSFP-PSM2)	OpticsFw_Port_0	1 1 .-1
NCS1K4-QXP-K9 (ONS-QSFP-PSM2)	OpticsFw_Port_4	1 1 .-1
NCS1K4-2.4T-K9 (ONS-QSFP-PSM2)	OpticsFw_Port_4	1 1 .-1
NCS1K14-OXP2-K9 (DP08SFP8-ZRB-A2)	OpticsFw_Port_0	5 4.9 1
NCS1K14-EDFA2 (DP04QSDD-E30-441)	OpticsFw_Port_7	0 5.6

Upgrade the software on NCS 1014

Configure and install a new Cisco IOS XR software version on the NCS 1014 platform.

Upgrade the Cisco IOS XR operating system to keep features current and maintain platform reliability.

NCS 1014 includes Cisco IOS XR by default, but software upgrades are available to add features or address issues. Upgrades are performed using an ISO image in XR mode. The NCS 1014 and related platforms use the same packaging image. For example, ISO images are named `ncs1010-x64-[sw-rel-ver].iso`.

Note

Upgrading from R7.11.1 to either R24.2.1 or R24.3.1 will raise the *DISASTER_RECOVERY_UNAVAILABLE_ALARM*. Postupgrade, this alarm clears automatically. For more information on the alarm, see *Troubleshooting Guide for Cisco NCS 1014*.

Before you begin

- Configure the management interface on NCS 1014.
- Copy the ISO image you want to install to the NCS 1014 hard disk or a network server accessible from NCS 1014.

Procedure

1. Install the new ISO image:
   a. To install from the hard disk, run **install replace /harddisk:**/*iso-image-name*
   b. To install from a network server (FTP, HTTP, HTTPS), run **install package replace** *<ftp or http or https protocol>/package_path filename1 filename2 ...*

   Note

   The **install package replace** command upgrades the ISO image but doesn't reload the RP automatically. But the **install replace** command upgrades the ISO image and reloads the RP.

   Installs the new ISO image from the harddisk or from the network server. The install operation takes 20–40 minutes to complete.

   Example

```
RP/0/RP0/CPU0:ios#install replace /harddisk:/ncs1010-x64-7.11.1.iso
Wed Nov 15 09:44:44.491 UTC
Once the packaging dependencies have been determined, the install operation
may have to reload the system.
If you want to control the timing of system reload, you must not continue,
but use the 'install package replace' command instead, followed by 'install
apply'.
Continue? [yes/no]:[yes]
Install replace operation 1.1 has started
Install operation will continue in the background
.....................
.....................
ios con0/RP0/CPU0 is now available
```

> **Note**
>
> Boot time FPD upgrade happens before XR boot. All the FPDs belonging to the RP location are upgraded during the boot time FPD upgrade.

> **Note**
>
> Automatic Field Programmable Device(FPD) upgrade is enabled by default.. When the automatic FPD upgrade is enabled, the install operation also upgrades the FPDs (except the Golden FPDs and Power modules) that need to be upgraded.

2. Check install status during the operation using the command **show install request**.

   Example

```
RP/0/RP0/CPU0:ios#show install request
Wed Nov 15 10:00:35.713 UTC
User request: install replace /harddisk://1010-x64-release.iso
Operation ID: 1.1
State:In progress since 2023-11-15 09:50:23 UTC
Current activity:    Package add or other package operation
Next activity:       Apply
Time started:        2023-11-15 09:55:24 UTC
Timeout in:          84m 43s
Locations responded: 0/1
Location          Packaging operation stage Notification Phase Clients
responded
----------------- ------------------------- ------------------
-----------------
0/RP0/CPU0        Package operations        None in progress
 N/A
```

   When the install operation completes successfully, the device automatically reloads.

> **Note**
>
> In case of the **install package replace** command, you'll be prompted to enter the next command (**install apply reload** command).

3. Commit the installed ISO image.

   Example

```
RP/0/RP0/CPU0:ios#install commit
Wed Nov 15 10:38:00.592 UTC
```

```
Install commit operation 1 has started
Install operation will continue in the background
```

> **Note**
>
> It is the mandatory to commit the install successfully to upgrade the software, missing this step followed by any controller reload/restart/power cycle will result in rollback to previously installed committed software/RPM package version.

The *DISASTER_RECOVERY_UNAVAILABLE_ALARM* clears upon completion of the upgrade from R7.11.1 to R24.2.1 or R24.3.1.

4. Verify the committed package information using the command **show install committed**.

Example

```
RP/0/RP0/CPU0:ios#show install committed
Wed Nov 15 10:41:20.454 UTC
Software Hash: ec69dcceb81c0da69b297aa7de1d00f56b8aef52403c5e0ffe6e5db098bd83b8
Package
Version
--------------------------------------------------------------------------------
xr-aaa
7.11.1.48Iv1.0.0-1
xr-acl
7.11.1.48Iv1.0.0-1
xr-apphosting
7.11.1.48Iv1.0.0-1
xr-appmgr
7.11.1.48Iv1.0.0-1
xr-bcdl
7.11.1.48Iv1.0.0-1
xr-bfd
7.11.1.48Iv1.0.0-1
xr-bgp
7.11.1.48Iv1.0.0-1
xr-bgputil
7.11.1.48Iv1.0.0-1
xr-bng-stubs
7.11.1.48Iv1.0.0-1
xr-bundles
7.11.1.48Iv1.0.0-1
```

: The NCS 1014 platform runs the updated Cisco IOS XR software. The device reloads as needed, relevant alarms (such as DISASTER_RECOVERY_UNAVAILABLE_ALARM) clear, and required FPDs upgrade automatically.

Software upgrade and downgrade matrix

This table lists the upgrade and downgrade paths supported for Cisco NCS 1014.

Upgrade Path			Downgrade Path		
Source Release	**Destination Release**	**Bridge SMU**	**Source Release**	**Destination Release**	**Bridge SMU**
• R7.11.1 • R24.1.1 • R24.2.11 • R24.3.1 • R24.4.1 • R25.1.1	R25.2.1	No	R25.2.1	• R25.1.1 • R24.4.1 • R24.3.1 • R24.2.11 • R24.1.1 • R7.11.1	No

Note

- Downgrading the software from version 24.4.1 to a lower version with loopback enabled is not supported and will affect traffic if attempted.
- Before upgrading to R24.4.1 or a later version, you must manually configure the wavelength or frequency to ensure a non-traffic-impacting software upgrade.

Supported software upgrade and downgrade versions

Provides information on Cisco IOS XR software upgrade and downgrade paths, including supported versions, prerequisites, limitations, health checks, and configuration options.

The Cisco chassis is preinstalled with IOS XR software. You can upgrade the software to access new features and fixes or downgrade if needed. To ensure compatibility and a successful upgrade or downgrade, it is important to choose versions aligned with Cisco-certified paths.

This reference provides answers to common questions:

- What upgrade or downgrade releases are supported for the current software version?
- Can the chassis be upgraded from Release X to Release Y?
- Are there any bridging Software Maintenance Updates (SMUs) required before upgrading?

To determine if the current release supports an upgrade to a target release, run the validation task at the start of your upgrade or downgrade using the **install replace** command. If validation fails, the upgrade is blocked and the system notifies you of the reason. You may use the `force` keyword with the command to bypass automatic validation. In this case, the system shows warning messages if the upgrade fails but does not block it.

The feature provides the following reference information to help understand the prerequisites or limitations for software upgrades or downgrades:

- Required bridging SMU RPMs
- Blocking SMU RPMs
- Unsupported hardware
- Caveats or restrictions

To view support information, use these **show** commands:

Command	Description
show install upgrade-matrix running	Displays all supported software upgrades from the current version according to the support data installed on the running system
show install upgrade-matrix iso *path-to-ISO*	Displays details about the software upgrade from the current version to the version of the target ISO according to the support data in both the running system and the ISO image
show install upgrade-matrix iso *path-to-ISO* **all**	Displays all supported software upgrades from any version according to the support data in the target ISO image
show install upgrade-matrix iso *path-to-ISO* **from-running**	Displays details about the software upgrade from the current version to the version of ISO according to the support matrices in both the running system and the target ISO image

This example shows all supported releases for upgrade from the current version 24.1.1 on the chassis:

```
RP/0/RP0/CPU0:ios#show install upgrade-matrix running
Fri Mar 15 12:53:23.715 IST
Matrix: XR version: 24.1.1, File version: 1.0, Version: N/A

The upgrade matrix indicates that the following system upgrades are supported
 from the current XR version:

From       To         Restrictions
---------- ---------- ------------
24.1.1     7.11.1     -

Add the from and to versions to the end of the CLI command, for data on versions
 with additional restrictions

For example, to display restrictions for the 24.1.1->7.11.1 upgrade, use
        'show install upgrade-matrix running 24.1.1 7.11.1'
```

Installation health checks (pre and post-upgrade)

Pre and post-upgrade installation health checks ensure system readiness and integrity. The existing client-server framework notifies subscribed clients to perform precheck functionality. The system health check infrastructure is plugged into the install precheck and postcheck phases during system upgrades, including other existing install checks.

Upgrade precheck:

- If single command upgrade is triggered either with a force option or is configured to skip checks, then health check is bypassed and a syslog entry added.
- When single command upgrade is triggered, install infra performs install specific prechecks. If the install prechecks pass, the system health check infra plug-in is invoked to check the overall system health.
- The health check infrastructure returns the health status during the installation.
- Single command upgrade continues on if the prechecks completes with no errors.
- If any errors are detected, then single command upgrade continues or terminates depending on the option that is selected for abort-on-precheck-failure.
- Single command upgrade postchecks before autocommit triggers based on the user selected level information.

Upgrade post check:

- Post checks are bypassed if force or config option is selected for single command upgrade.
- If install specific postchecks are completed sucessfully, then the system health check infra plug-in is invoked. If no errors are reported then the autocommit triggers.
- If any errors are detected, the abort-on option that is saved before the upgrade reload is used to either abort the single command upgrade or continue. This depends on the severity of the errors that are detected during post check.
- Summary of the pre and posthealth check is appended to the single command upgrade operation log.

Installation Profile Creation

Installation Profile is created to choose and alternate installation behavior. One default profile is created involving pre and postchecks. You can edit the install behavior to choose cases like terminate installation if precheck fails or revert after post installation check. You can also choose to continue installation despite failure in pre checks.

You can configure “enable or disable” options to run pre or post installation checks or “abort-on-failure” for pre checks, or "warn-on-failure" and “restore-to-v1” on post checks. To configure the Install profile, use the following commands:

config

install profile *profile_name* **pre-check***metric-name* **[enable | disable] [abort-on-failure | continue-on-failure | revert-on-failure]**

end

This is a sample to display metric settings in the install profile.

```
RP/0/RP0/CPU0:ios#show install profile default
Fri Mar 15 11:29:35.381 IST
Profile Name : default
State : Enabled

Prechecks : Enabled
        communication-timeout   : Enabled       [ warn-on-failure ]
        config-inconsistency    : Enabled       [ error-on-failure ]
        process-resource        : Enabled       [ warn-on-failure ]
        process-status          : Enabled       [ warn-on-failure ]
        system-clock            : Enabled       [ warn-on-failure ]
        hw-monitoring           : Enabled       [ warn-on-failure ]
        lc-monitoring           : Enabled       [ warn-on-failure ]
        pci-monitoring          : Enabled       [ warn-on-failure ]
        wd-monitoring           : Enabled       [ warn-on-failure ]
        disk-space              : Enabled       [ error-on-failure ]
        upgrade_matrix          : Enabled       [ error-on-failure ]
        core-cleanup            : Disabled      [ NA ]
        file-cleanup            : Disabled      [ NA ]

Postchecks : Enabled
        communication-timeout   : Enabled       [ error-on-failure ]
        config-inconsistency    : Enabled       [ error-on-failure ]
        process-resource        : Enabled       [ error-on-failure ]
        process-status          : Enabled       [ error-on-failure ]
        system-clock            : Enabled       [ error-on-failure ]
        hw-monitoring           : Enabled       [ error-on-failure ]
        lc-monitoring           : Enabled       [ error-on-failure ]
        pci-monitoring          : Enabled       [ error-on-failure ]
        wd-monitoring           : Enabled       [ error-on-failure ]
```

Use this configuration to report health check:

config

grpc local-connection

Netconf-yang agent

commit

The following is a sample to display health check states:

```
RP/0/RP0/CPU0:ios#show healthcheck internal states
Fri Mar 15 12:55:54.739 IST

 Internal Structure INFO

 Current state: Disabled

 Reason: Success

 Netconf Config State: Enabled

 Grpc Config State: Disabled

 Nosi state: Not ready

 Appmgr conn state: Invalid

 Nosi lib state: Not ready

 Nosi client: Valid client
```

FPD upgrades

Explains the FPD types, Golden FPD behavior, and status values that you use to evaluate firmware state before upgrading NCS 1014 hardware.

A Field Programmable Device (FPD) is a programmable hardware component that

- incorporates a Field Programmable Gate Array (FPGA) for flexible operation,
- requires verification and periodic upgrades to keep line card firmware current for chassis operation, and
- functions in multiple modules, including Route Processor, Power Modules, Line Cards, and Rack components of NCS 1014 systems.

FPD details and status values

> **Note**
>
> During the software upgrade, when the SSD is upgraded, the FPD goes into the RELOAD_REQ state, as displayed by the **show hw-module fpd** command. This behavior is expected because the updated SSD firmware can only be activated after reloading the specific SSD location mentioned in the **show hw-module fpd** output.

This table lists the NCS 1014 FPDs that are distributed across Route Processor (RP), Power Modules (PM), Line Cards (LC), and Rack.

Table 8: NCS 1014 FPDs

Location	FPDs
RP	• ADM-DB • ADM-MB • BIOS • BIOS-Golden • CpuFpga • CpuFpgaGolden • SsdIntelS4510 • SsdIntelSC2KB • SsdMicron5300 • TamFw • TamFwGolden
PM0 and PM1	• PO-PriMCU • PO-SecMCU
LC	• CpuModFw • OptModFw
Rack	• ADM-CHASSIS • IoFpga • IoFpgaGolden • SsdIntelSC2KB

Golden FPDs

Golden FPDs serve as backup FPDs for the primary FPDs. For example, **BIOS-Golden** is the backup Golden FPD for the **BIOS** primary FPD. If a primary FPD is corrupted, NCS 1014 boots with the corresponding Golden FPD. The Golden FPDs cannot be upgraded.

Retrieve FPD information

There are multiple types of FPDs for each type of module. The **show hw-module fpd** command provides information about each FPD.

Note

From R26.2.1, the SSD FPD name displayed in the output of the **show hw-module fpd** command is changed from a vendor or model-specific name to a generic format: **CPU-SSD** for RP locations and **CHASSIS-SSD** for rack locations.

```
RP/0/RP0/CPU0:235#show hw-module fpd
Mon Jan 12 11:44:18.438 UTC

Auto-upgrade:Enabled,PM excluded
Attribute codes: B golden, P protect, S secure, A Anti Theft aware
                                                                                FPD
Versions

==============
Location   Card type              HWver FPD device       ATR Status    Running
  Programd   Reload Loc
-------------------------------------------------------------------------------
0/RP0/CPU0 NCS1K14-CNT-B-K9       0.2   ADM-DB               CURRENT      2.10
     2.10         NOT REQ
0/RP0/CPU0 NCS1K14-CNT-B-K9       0.2   ADM-MB               CURRENT      2.30
     2.30         NOT REQ
0/RP0/CPU0 NCS1K14-CNT-B-K9       0.2   BIOS             S   CURRENT      6.10
     6.10           0/RP0
0/RP0/CPU0 NCS1K14-CNT-B-K9       0.2   BIOS-Golden      BS  CURRENT
     4.50           0/RP0
0/RP0/CPU0 NCS1K14-CNT-B-K9       0.2   CpuFpga          S   CURRENT      1.17
     1.17           0/RP0
0/RP0/CPU0 NCS1K14-CNT-B-K9       0.2   CpuFpgaGolden    BS  CURRENT
     1.03           0/RP0
0/RP0/CPU0 NCS1K14-CNT-B-K9       0.2   SsdMicron5300    S   CURRENT      0.01
     0.01           0/RP0
0/RP0/CPU0 NCS1K14-CNT-B-K9       0.2   TamFw            S   CURRENT      9.04
     9.04           0/RP0
0/RP0/CPU0 NCS1K14-CNT-B-K9       0.2   TamFwGolden      BS  CURRENT
     9.04           0/RP0
0/PM1      NCS1K4-AC-PSU          0.1   PO-PriMCU            CURRENT      2.04
     2.04         NOT REQ
0/PM1      NCS1K4-AC-PSU          0.1   PO-SecMCU            CURRENT      2.06
     2.06         NOT REQ
0/1/NXR0   NCS1K14-2.4T-A-K9     0.1   CpuModFw         S   CURRENT     261.100
   261.100          NOT REQ
0/2/NXR0   NCS1K14-2.4T-A-K9     0.1   CpuModFw         S   CURRENT     261.100
   261.100          NOT REQ
0/2/NXR0   NCS1K14-2.4T-A-K9     0.0   OpticsFw_Port_0  S   CURRENT    80.14014
 80.14014       NOT REQ
0/2/NXR0   NCS1K14-2.4T-A-K9     0.0   OpticsFw_Port_7  S   CURRENT    80.14014
 80.14014       NOT REQ
0/Rack     NCS1014                0.1   ADM-CHASSIS          CURRENT      0.21
     0.21         NOT REQ
0/Rack     NCS1014                0.1   IoFpga           S   CURRENT      2.26
     2.26         NOT REQ
0/Rack     NCS1014                0.1   IoFpgaGolden     BS  CURRENT
     1.05         NOT REQ
0/Rack     NCS1014                0.1   SsdIntelSC2KB    S   CURRENT      1.30
```

```
     1.30         0/Rack
RP/0/RP0/CPU0:235#
```

```
RP/0/RP0/CPU0:ios#show hw-module fpd
Wed Nov 15 19:29:37.061 UTC

Auto-upgrade:Enabled
Attribute codes: B golden, P protect, S secure, A Anti Theft aware
                                                                    FPD Versions


==============
Location    Card type       HWver   FPD device     ATR Status   Running Programd
 Reload Loc
-------------------------------------------------------------------------------
0/RP0/CPU0 NCS1K14-CNTLR-K9  0.2   ADM-DB              CURRENT    2.10  2.10
   NOT REQ
0/RP0/CPU0 NCS1K14-CNTLR-K9  0.2   ADM-MB              CURRENT    2.30  2.30
   NOT REQ
0/RP0/CPU0 NCS1K14-CNTLR-K9  0.2   BIOS S              CURRENT    4.70  4.70
   0/RP0
0/RP0/CPU0 NCS1K14-CNTLR-K9  0.2   BIOS-Golden      BS CURRENT          4.70
   0/RP0
0/RP0/CPU0 NCS1K14-CNTLR-K9  0.2   CpuFpga           S CURRENT    1.09  1.09
   0/RP0
0/RP0/CPU0 NCS1K14-CNTLR-K9  0.2   CpuFpgaGolden    BS CURRENT          1.09
   0/RP0
0/RP0/CPU0 NCS1K14-CNTLR-K9  0.2   SsdMicron5300     S CURRENT    0.01  0.01
   0/RP0
0/RP0/CPU0 NCS1K14-CNTLR-K9  0.2   TamFw             S CURRENT    9.04  9.04
   0/RP0
0/RP0/CPU0 NCS1K14-CNTLR-K9  0.2   TamFwGolden      BS CURRENT          9.04
   0/RP0
0/PM0      NCS1K4-AC-PSU     0.1   PO-PriMCU           CURRENT    2.04  2.04
   NOT REQ
0/PM0      NCS1K4-AC-PSU     0.1   PO-SecMCU           CURRENT    2.06  2.06
   NOT REQ
0/PM1      NCS1K4-AC-PSU     0.1   PO-PriMCU           CURRENT    2.04  2.04
   NOT REQ
0/PM1      NCS1K4-AC-PSU     0.1   PO-SecMCU           CURRENT    2.06  2.06
   NOT REQ
0/0/NXR0   NCS1K4-1.2T-K9    0.1   CpuModFw          S CURRENT   234.10 234.10
  NOT REQ
0/0/NXR0   NCS1K4-1.2T-K9    0.1   OptModFw           S CURRENT    1.38  1.38
   NOT REQ
0/1/NXR0   NCS1K14-2.4T-K9   0.1   CpuModFw          S CURRENT   234.10 234.10
  NOT REQ
0/2/NXR0   NCS1K14-CCMD-16-C 0.1   CpuModFw          S CURRENT   234.10 234.10
  NOT REQ
0/2/NXR0   NCS1K14-CCMD-16-C 0.1   OptModFw           S CURRENT     1.38 1.38
   NOT REQ
0/3/NXR0   NCS1K4-1.2T-K9    0.1   CpuModFw          S CURRENT   234.10 234.10
  NOT REQ
0/3/NXR0   NCS1K4-1.2T-K9    0.1   OptModFw           S CURRENT     1.38 1.38
   NOT REQ
0/Rack     NCS1014           0.1   ADM-CHASSIS          CURRENT     0.21 0.21
   NOT REQ
0/Rack     NCS1014           0.1   IoFpga             S CURRENT     1.10 1.10
   NOT REQ
0/Rack     NCS1014           0.1   IoFpgaGolden     BS CURRENT          1.05
   NOT REQ
```

```
0/Rack     NCS1014           0.1   SsdIntelSC2KB     S CURRENT     1.20 1.20
   0/Rack
```

```
RP/0/RP0/CPU0:RINode1#show hw-module fpd
Mon Feb 24 14:30:20.742 IST

Auto-upgrade:Enabled,PM excluded
Attribute codes: B golden, P protect, S secure, A Anti Theft aware
                                                                                FPD
Versions

==============
Location   Card type              HWver FPD device       ATR Status    Running
  Programd   Reload Loc
-----------------------------------------------------------------------------------
0/RP0/CPU0 NCS1K14-CNTLR-K9       1.0   ADM-DB               CURRENT      2.10
     2.10         NOT REQ
0/RP0/CPU0 NCS1K14-CNTLR-K9       1.0   ADM-MB               CURRENT      2.30
     2.30         NOT REQ
0/RP0/CPU0 NCS1K14-CNTLR-K9       1.0   BIOS             S   CURRENT      5.00
     5.00           0/RP0
0/RP0/CPU0 NCS1K14-CNTLR-K9       1.0   BIOS-Golden      BS  CURRENT
     4.70           0/RP0
0/RP0/CPU0 NCS1K14-CNTLR-K9       1.0   CpuFpga          S   CURRENT      1.17
     1.17           0/RP0
0/RP0/CPU0 NCS1K14-CNTLR-K9       1.0   CpuFpgaGolden    BS  CURRENT
     1.09           0/RP0
0/RP0/CPU0 NCS1K14-CNTLR-K9       1.0   SsdMicron5300    S   CURRENT      0.01
     0.01           0/RP0
0/RP0/CPU0 NCS1K14-CNTLR-K9       1.0   TamFw            S   CURRENT      9.04
     9.04           0/RP0
0/RP0/CPU0 NCS1K14-CNTLR-K9       1.0   TamFwGolden      BS  CURRENT
     9.04           0/RP0
0/PM0      NCS1K4-AC-PSU-2        1.1   PO-PriMCU            CURRENT      1.03
     1.03         NOT REQ
0/PM0      NCS1K4-AC-PSU-2        1.1   PO-SecMCU            CURRENT      1.05
     1.05         NOT REQ
0/0/NXR0   NCS1K14-EDFA2          0.1   CpuModFw         S   CURRENT    252.06
   252.06         NOT REQ
0/0/NXR0   NCS1K14-EDFA2          0.1   OptModFw         S   CURRENT      2.02
     2.02         NOT REQ
0/3/NXR0   NCS1K4-1.2T-K9         0.1   CpuModFw         S   CURRENT    252.06
   252.06         NOT REQ
0/3/NXR0   NCS1K4-1.2T-K9         0.1   OptModFw         S   CURRENT      1.38
     1.38         NOT REQ
0/Rack     NCS1014                1.1   ADM-CHASSIS          CURRENT      0.21
     0.21         NOT REQ
0/Rack     NCS1014                1.1   IoFpga           S   CURRENT      1.23
     1.23         NOT REQ
0/Rack     NCS1014                1.1   IoFpgaGolden     BS  CURRENT
     1.05         NOT REQ
0/Rack     NCS1014                1.1   SsdIntelSC2KB    S   CURRENT      1.30
     1.30          0/Rack
0/4        NCS1K-MD-32O-CE        0.2   MD-32-LUM        S   CURRENT      2.20
     2.20         NOT REQ
0/5        NCS1K-MD-32E-CE        12.1  MD-32-ACC        S   CURRENT      1.12
     1.12         NOT REQ
```

The following table describes the significant fields in the output of the **show hw-module fpd** command.

Table 9: Description of Fields in show hw-module fpd Command

Field	Description
Location	Location of the FPD.
Card type	PID of the modules such as chassis, card, CPU, and PSU.
HWver	Hardware version where the FPD resides.
FPD device	Name of the FPD.
ATR	Attribute codes. The possible values are: • B - Golden Image • S - Secure Image • P - Protect Image The attribute code of the primary FPDs is S and the Golden FPDs is BS.
Status	Status of the FPD. See Table 10: Description of FPD Status Values in show hw-module fpd Command Output on page 83.
Running	FPD image version that has been activated and currently running in the FPD device.
Programd	FPD image version that has been programmed into the FPD device, but might not be activated.
Reload Loc	Indicates whether reload of the location is required or not.

The following table describes the possible values of the **Status** field in the output of the **show hw-module fpd** command.

Table 10: Description of FPD Status Values in show hw-module fpd Command Output

FPD Status	Description
NOT READY	The driver that owns the FPD device has not initialized the FPD client to handle this device.
CURRENT	FPD version is up-to-date and upgrade is not required.
NEED UPGD	Upgrade is required for this FPD. Check the output of the **show fpd package** command to determine the recommended FPD version.
UPGD PREP	FPD is preparing for upgrade.
IN QUEUE	Upgrade of this FPD is in queue.
UPGD SKIP	FPD upgrade is not required. For example, • FPD version is up-to-date and compatible. • FPD image is protected.

FPD Status	Description
UPGRADING	FPD upgrade has started and the driver has not reported the upgrade progress information yet.
%UPGD	Percentage of FPD upgrade completion.
RLOAD REQ	FPD upgrade is successful and the FPD must be reloaded for the new version to take effect.
UPGD FAIL	FPD upgrade has failed. Check the syslog for any timeout messages or any failure reported by the driver.
UPGD DONE	FPD upgrade is successful.

Restriction

The NCS 1014 does not support trunk FPD upgrade on the QXP card.

FPD upgrade verification fields

Describes the show hw-module fpd fields and status values that indicate whether an FPD is current, queued, failed, or ready for upgrade.

Use the **show hw-module fpd** command to check whether all the FPDs are in the **CURRENT** state. If the status of any FPD is **NEED UPGD**, then the upgrade is required for that FPD.

The following table describes the significant fields in the output of the **show hw-module fpd** command.

Table 11: Description of Fields in show hw-module fpd Command

Field	Description
Location	Location of the FPD.
Card type	PID of the modules such as chassis, card, CPU, and PSU.
HWver	Hardware version where the FPD resides.
FPD device	Name of the FPD.
ATR	Attribute codes. The possible values are: • B - Golden Image • S - Secure Image • P - Protect Image The attribute code of the primary FPDs is S and the Golden FPDs is BS.
Status	Status of the FPD. See Table 12: Description of FPD Status Values in show hw-module fpd Command on page 85
Running	FPD image version that has been activated and currently running in the FPD device.

Field	Description
Programd	FPD image version that has been programmed into the FPD device, but might not be activated.
Reload Loc	Indicates whether a reload of the location is required or not.

The following table describes the possible values of the **Status** field in the output of the **show hw-module fpd** command.

Table 12: Description of FPD Status Values in show hw-module fpd Command

FPD Status	Description
NOT READY	The driver that owns the FPD device has not initialized the FPD client to handle this device.
CURRENT	FPD version is up-to-date and upgrade is not required.
NEED UPGD	Upgrade is required for this FPD.
UPGD PREP	FPD is preparing for upgrade.
IN QUEUE	Upgrade of this FPD is in queue.
UPGD SKIP	FPD upgrade is not required. For example, • FPD version is up-to-date and compatible. • FPD image is protected.
UPGRADING	FPD upgrade has started and the driver has not reported the upgrade progress information yet.
%UPGD	Percentage of FPD upgrade completion.
RLOAD REQ	FPD upgrade is successful and the FPD must be reloaded for the new version to take effect.
UPGD FAIL	FPD upgrade has failed. Check the syslog for any timeout messages or any failure reported by the driver.
UPGD DONE	FPD upgrade is successful.

FPD can be upgraded using two methods:

- Upgrade FPDs manually on page 92
- Configure automatic FPD upgrade on page 93

Verify whether an FPD upgrade is required

Teaches you how to use show hw-module fpd and show fpd package outputs to determine whether an FPD upgrade is required.

Use this task to verify FPD status and identify supported FPD package versions before you upgrade firmware.

Table 13: Feature History

Feature Name	Release Information	Feature Description
Automatic FPD Upgrade Support for Coherent Interconnect Module 8	Cisco IOS XR Release 24.3.1	The Coherent Interconnect Module 8 (CIM8) FPD is automatically upgraded to the latest qualified version when the line card FPD is upgraded. This ensures that both the line card and CIM8 operate with optimized performance and improved interoperability. Supported line cards are: • NCS1K14-2.4T-X-K9 • NCS1K4-2.4T-K9

Procedure

1. Use the **show hw-module fpd** command to check whether all the FPDs are in the **CURRENT** state.

   If the status of any FPD is **NEED UPGD**, then the upgrade is required for that FPD.

2. Use the **show fpd package** command to determine the FPDs that are supported with the current software release and the minimum hardware requirements for each FPD.

   The Coherent Interconnect Module 8 (CIM8) FPD is automatically upgraded to the latest qualified version while the line card FPD is being upgraded. If the CIM 8 has a higher version than the line card's CIM 8 FPD version, the CIM 8 remains at the higher version and does not downgrade.

   - NCS1K14-2.4T-X-K9
   - NCS1K4-2.4T-K9
   - NCS1K14-2.4T-A-K9
   - NCS1K14-2.4TAL-K9

   You can see the CIM 8 FPD version in the **show fpd package** command output.

```
RP/0/RP0/CPU0:ios#show fpd package
Mon Jan 12 11:48:15.912 UTC

================================
================================================
                                          Field Programmable Device Package

================================================
                                                Req     SW      Min Req   Min
 Req
Card Type            FPD Description           Reload  Ver     SW Ver  Board
 Ver
==================== ========================== ====== ======= ========
=========
--------------------------------------------------------------------------------
NCS1014              ADM-CHASSIS                 NO      0.21      0.21
0.0
```

```
                      IoFpga                      NO      1.19      1.19
0.0
                      IoFpgaGolden                NO      1.05      1.05
0.0
                      SsdIntelSC2KB               YES     1.20      1.20
0.0
                      SsdMicron5400               YES     0.02      0.02
0.0
--------------------------------------------------------------------------------
NCS1K14-2.4T-A-K9     CIM8CEK9ACA                 NO     80.14014  80.14014
0.0
                      CIM8CK9ACA                  NO     80.14014  80.14014
0.0
                      CIM8LEK9ACA                 NO     80.14014  80.14014
0.0
                      CIM8LK9ACA                  NO     80.14014  80.14014
0.0
                      CpuModFw                    NO     261.100   261.100
 0.0
--------------------------------------------------------------------------------
NCS1K14-2.4TAL-K9     CIM8CEK9ACA                 NO     80.14014  80.14014
0.0
                      CIM8CK9ACA                  NO     80.14014  80.14014
0.0
                      CIM8LEK9ACA                 NO     80.14014  80.14014
0.0
                      CIM8LK9ACA                  NO     80.14014  80.14014
0.0
                      CpuModFw                    NO     261.100   261.100
 0.0
--------------------------------------------------------------------------------
```

```
RP/0/RP0/CPU0:ios#show fpd package
Wed Jul 17 11:44:07.258 IST

================================
================================================
                                        Field Programmable Device Package

================================================
                                                Req     SW      Min Req   Min
 Req
Card Type            FPD Description            Reload  Ver     SW Ver  Board
 Ver
==================== ========================== ====== ======= ========
=========
--------------------------------------------------------------------------------
NCS1014              ADM-CHASSIS                 NO      0.21      0.21
0.0
                     IoFpga                      NO      1.19      1.19
0.0
                     IoFpgaGolden                NO      1.05      1.05
0.0
                     SsdIntelSC2KB               YES     1.20      1.20
0.0
                     SsdMicron5400               YES     0.02      0.02
0.0
--------------------------------------------------------------------------------
NCS1K-MD-32E-C       MD-32-ACC                   NO      2.18      2.18
0.0
```

```
                     MD-32-NEO                  NO      2.02      2.02
0.0
.
.
.
--------------------------------------------------------------------------------
NCS1K14-2.4T-K9      CIMFw                      NO    180.13019 180.13019
0.0
                     CpuModFw                   NO     43.27     43.27
0.0
--------------------------------------------------------------------------------
NCS1K14-2.4T-L-K9    CIMFw                      NO    180.13019 180.13019
0.0
                     CpuModFw                   NO     43.27     43.27
0.0
--------------------------------------------------------------------------------
NCS1K14-2.4T-X-K9    CIMFw                      NO    180.13019 180.13019
0.0
                     CpuModFw                   NO     43.27     43.27
0.0
--------------------------------------------------------------------------------
NCS1K14-2.4TXL-K9    CIMFw                      NO    180.13019 180.13019
0.0
                     CpuModFw                   NO     43.27     43.27
0.0
--------------------------------------------------------------------------------
.
.
NCS1K14-CNTLR-K9     ADM-DB                     NO      2.10      2.10
0.2
                     ADM-MB                     NO      2.30      2.30
0.2
                     BIOS                       YES     4.80      4.80
0.0
                     BIOS-Golden                YES     4.70      0.01
0.0
```

```
RP/0/RP0/CPU0:RINode1#show fpd package
Mon Feb 24 14:31:30.361 IST

================================
================================================
                                        Field Programmable Device Package

================================================
                                             Req     SW      Min Req   Min
 Req
Card Type            FPD Description         Reload  Ver     SW Ver  Board
 Ver
==================== ======================== ====== ======= ========
=========
--------------------------------------------------------------------------------
NCS1014              ADM-CHASSIS               NO      0.21      0.21
0.0
                     IoFpga                    NO      1.19      1.19
0.0
                     IoFpgaGolden              NO      1.05      0.01
0.0
                     SsdIntelSC2KB             YES     1.30      1.30
0.0
                     SsdMicron5400             YES     0.02      0.02
```

```
0.0
                     SsdSolidigmSC2KB           YES     1.30      1.30
0.0
--------------------------------------------------------------------------------
NCS1K-MD-32E-C       MD-32-ACC                  NO      2.18      2.18
0.0
                     MD-32-NEO                  NO      2.02      2.02
0.0
--------------------------------------------------------------------------------
NCS1K-MD-32E-CE      MD-32-ACC                  NO      1.12      1.12
0.0
                     MD-32-LUM                  NO      2.20      2.20
0.0
--------------------------------------------------------------------------------
NCS1K-MD-32O-C       MD-32-ACC                  NO      2.18      2.18
0.0
                     MD-32-NEO                  NO      2.02      2.02
0.0
--------------------------------------------------------------------------------
NCS1K-MD-32O-CE      MD-32-ACC                  NO      1.12      1.12
0.0
                     MD-32-LUM                  NO      2.20      2.20
0.0
--------------------------------------------------------------------------------
NCS1K14-2.4T-K9      CIMFw                      NO     80.13021  80.13021
0.0
                     CpuModFw                   NO    252.06    252.06
0.0
--------------------------------------------------------------------------------
NCS1K14-2.4T-L-K9    CIMFw                      NO     80.13021  80.13021
0.0
                     CpuModFw                   NO    252.06    252.06
0.0
--------------------------------------------------------------------------------
NCS1K14-2.4T-X-K9    CIMFw                      NO     80.13021  80.13021
0.0
                     CpuModFw                   NO    252.06    252.06
0.0
--------------------------------------------------------------------------------
NCS1K14-2.4TXL-K9    CIMFw                      NO     80.13021  80.13021
0.0
                     CpuModFw                   NO    252.06    252.06
0.0
--------------------------------------------------------------------------------
NCS1K14-CCMD-16-C    CpuModFw                   NO    252.06    252.06
0.0
                     OptModFw                   NO     20.02     20.02
0.0
--------------------------------------------------------------------------------
NCS1K14-CCMD-16-L    CpuModFw                   NO    252.06    252.06
0.0
                     OptModFw                   NO     20.02     20.02
0.0
--------------------------------------------------------------------------------
NCS1K14-CNTLR-K9     ADM-DB                     NO      2.10      2.10
0.2
                     ADM-MB                     NO      2.30      2.30
0.2
                     BIOS                       YES     5.00      5.00
0.0
                     BIOS-Golden                YES     4.70      0.01
0.0
```

```
                     CpuFpga                   YES     1.17      1.17
0.0
                     CpuFpgaGolden             YES     1.09      0.01
0.0
                     SsdIntelS4510             YES    11.51     11.51
0.0
                     SsdIntelSC2KB             YES     1.30      1.30
0.0
                     SsdIntelSCKKBGZ           YES     1.30      1.30
0.0
                     SsdMicron5300             YES     0.01      0.01
0.0
                     SsdMicron5400             YES     0.02      0.02
0.0
                     SsdSolidigmSC2KB          YES     1.30      1.30
0.0
                     SsdSolidigmSCKKB          YES     1.30      1.30
0.0
                     SsdSRM28480GF1            YES    14.03     14.03
0.0
                     TamFw                     YES     9.04      9.04
0.0
                     TamFwGolden               YES     9.04      0.01
0.0
--------------------------------------------------------------------------------
NCS1K14-CTLR-B-K9    ADM-DB                    NO      2.10      2.10
0.2
                     ADM-MB                    NO      2.30      2.30
0.2
                     BIOS                      YES     5.00      5.00
0.0
                     BIOS-Golden               YES     4.70      0.01
0.0
                     CpuFpga                   YES     1.17      1.17
0.0
                     CpuFpgaGolden             YES     1.09      0.01
0.0
                     SsdIntelS4510             YES    11.51     11.51
0.0
                     SsdIntelSC2KB             YES     1.30      1.30
0.0
                     SsdIntelSCKKBGZ           YES     1.30      1.30
0.0
                     SsdMicron5300             YES     0.01      0.01
0.0
                     SsdMicron5400             YES     0.02      0.02
0.0
                     SsdSolidigmSC2KB          YES     1.30      1.30
0.0
                     SsdSolidigmSCKKB          YES     1.30      1.30
0.0
                     SsdSRM28480GF1            YES    14.03     14.03
0.0
                     TamFw                     YES     9.04      9.04
0.0
                     TamFwGolden               YES     9.04      0.01
0.0
--------------------------------------------------------------------------------
NCS1K14-EDFA2        CohProbFw                 NO     70.13021  70.13021
0.0
                     CpuModFw                  NO    252.06    252.06
0.0
```

```
                      DracoFW                     NO      0.09      0.09
0.1
                      DracoFW                     NO      0.09      0.09
0.2
                      OptModFw                    NO      2.02      2.02
0.0
--------------------------------------------------------------------------------
NCS1K4-1.2T-K9        CpuModFw                    NO    252.06    252.06
0.0
                      OptModFw                    NO      1.38      1.38
0.0
--------------------------------------------------------------------------------
NCS1K4-1.2T-L-K9      CpuModFw                    NO    252.06    252.06
0.0
                      OptModFw                    NO      1.38      1.38
0.0
--------------------------------------------------------------------------------
NCS1K4-2-QDD-C-K9     CpuModFw                    NO    252.06    252.06
0.0
                      OptModFw                    NO      1.38      1.38
0.0
--------------------------------------------------------------------------------
NCS1K4-AC-PSU         PO-PriMCU                   NO      2.04      2.04
0.1
                      PO-SecMCU                   NO      2.06      2.06
0.1
--------------------------------------------------------------------------------
NCS1K4-AC-PSU-2       PO-PriMCU                   NO      1.03      1.03
0.1
                      PO-SecMCU                   NO      1.05      1.05
0.1
--------------------------------------------------------------------------------
NCS1K4-QXP-K9         CpuModFw                    NO    252.06    252.06
0.0
--------------------------------------------------------------------------------
NCS1K4-QXP-L-K9       CpuModFw                    NO    252.06    252.06
0.0
```

This table describes the fields in the output of the **show fpd package** command.

Table 14: Description of Fields in show fpd package Command

Field	Description
Card Type	PID of the modules such as chassis, card, CPU, and PSU.
FPD Description	Description of the FPD.
Req Reload	Determines whether reload is required to activate the FPD image.
SW Ver	Recommended FPD software version for the associated module running the current Cisco IOS XR Software.
Min Req SW Ver	Minimum required FPD software version to operate the module.
Min Req Board Ver	Minimum required hardware version for the associated FPD. A minimum hardware requirement of version 0.0 indicates that all the hardware can support this FPD version.

FPD can be upgraded using two methods:

- Manual upgrade
- Automatic upgrade

You know whether the FPDs are current or require upgrade, and which FPD package versions are supported by the current software release.

Upgrade FPDs manually

Teaches you how to upgrade specific FPDs, reload locations when required, and run optional commands for force upgrades or concurrent FPD upgrades.

Use this task to manually upgrade one or more FPDs and reload locations when a reload is required to activate the new firmware.

Procedure

---

1. Use the **upgrade hw-module location** *[location-id]* **fpd** *[fpd name]* command to upgrade a specific FPD.

   > **Note**
   >
   > FPD upgrades are non-traffic affecting.

   Example

   ```
   RP/0/RP0/CPU0:ios#upgrade hw-module location 0/0/NXR0 fpd CPUModFw
   ```

   This is a sample configuration to upgrade the FPD of a trunk pluggable.

   ```
   RP/0/RP0/CPU0:ios#upgrade hw-module location 0/3/nXR0 fpd OpticsFw_Port_7
   ```

2. Use the **show hw-module fpd** command to display information about the completed FPD upgrade.
3. (Optional) Use the **upgrade hw-module location** *[location-id]* **fpd** *[fpd name]* **force** command to forcibly upgrade a specific FPD irrespective of whether the upgrade is required or not.

   Example

   ```
   RP/0/RP0/CPU0:ios#upgrade hw-module location 0/0/NXR0 fpd CPUModFw force
   ```

   This is a sample configuration to forcibly upgrade the FPD of a trunk pluggable.

   ```
   RP/0/RP0/CPU0:ios#upgrade hw-module location 0/3/nXR0 fpd OpticsFw_Port_7
   force
   ```

4. Use the **reload location** *location-id* to reload the FPDs belonging to a specific location with the new version.

   The **Reload Loc** field in the output of **show hw-module fpd** command indicates whether the reload is required or not.

Example

```
RP/0/RP0/CPU0:ios#reload location 0/RP0/CPU0
```

5. (Optional) Use the **upgrade hw-module location all fpd all** command to upgrade all the FPDs concurrently.

   Example

   ```
   RP/0/RP0/CPU0:ios#upgrade hw-module location all fpd all
   ```

   Note

   You cannot upgrade PSU FPD using the **location all fpd all** command. You can execute the Step 6 command to upgrade PSU FPD.

6. (Optional) Use the **upgrade hw-module [location** [*location-id* | **all**]] **fpd** [*fpd name*] | **all**] command to upgrade a specific FPD, all the FPDs, or the FPDs belonging to a specific location.

   Note

   Until Release 24.2.1, you cannot forcefully upgrade FPDs of power modules and SSDs.

   From Release 24.3.1, you can upgrade power module FPDs by using the command:

   upgrade hw-module [location [*location-id* | **all**]] **[pm** [*pm-id* | **all**]] **fpd** [*fpd name*] | **all**].

The selected FPDs are upgraded, and any required reload activates the new firmware version.

Configure automatic FPD upgrade

Teaches you how to disable automatic FPD upgrade and configure whether power module FPDs are included in the automatic upgrade workflow.

Use this task to control automatic FPD upgrade behavior on NCS 1014 and to configure whether power module FPDs are included in automatic upgrades.

Table 15: Feature History

Feature Name	Release Information	Feature Description
Automatic FPD Upgrade Support for Power Module	Cisco IOS XR Release 26.1.1	The FPD upgrade for power modules is now integrated with the NCS 1014 automatic FPD upgrade. You have the flexibility to include or exclude the power module FPD in the automatic upgrade according to your operational requirements. This option is disabled by default. Command added: • **fpd auto-upgrade {include | exclude} pm** You can also enable the automatic FPD upgrade for power modules using the OpenConfig data model Cisco-IOS-XR-openconfig-system-fpd-ext.

The automatic FPD upgrade process upgrades firmware with the **NEED UPGD** status to the **CURRENT** status automatically. Use the **show hw-module fpd** command to view the latest status after the automatic upgrade is completed.

In NCS 1014, automatic FPD upgrade is enabled by default.

Procedure

1. Disable the automatic FPD upgrade.

   Example

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#fpd auto-upgrade disable
RP/0/RP0/CPU0:ios(config)#commit
RP/0/RP0/CPU0:ios(config)#end
```

   Note

   - CpuModFw is upgraded during the automated FPD upgrade for cards NCS1K14-2.4T-K9 and NCS1K14-2.4T-L-K9.
   - OptModFw is upgraded first followed by CpuModFw during automated FPD upgrade for the cards NCS1K14-CCMD-16-C, NCS1K14-CCMD-16-L, NCS1K4-1.2T-K9, and NCS1K14-EDFA2.
   - Until R24.2.1, you cannot do an automatic upgrade for the FPD power module.

2. From R24.3.1, include or exclude automatic upgrade of power modules using the **fpd auto-upgrade {include | exclude} pm** command.

Example

```
RP/0/RP0/CPU0:ios#fpd auto-upgrade include pm
RP/0/RP0/CPU0:ios#commit
```

Example

```
RP/0/RP0/CPU0:ios#fpd auto-upgrade exclude pm
RP/0/RP0/CPU0:ios#commit
```

Automatic FPD upgrade is disabled, or power module FPD inclusion is configured according to the selected command.

Factory reset

Describes factory reset behavior, limitations, and data-removal locations that help prepare an NCS 1000 chassis for product return and replace processing.

This feature allows you to remove all data from the device before initiating a Product Return and Replace (PRR). This process ensures that all sensitive customer information is permanently deleted and cannot be recovered.

Table 16: Feature History

Feature Name	Release Information	Feature Description
Factory Reset	Cisco IOS XR Release 25.4.1	This feature allows customers to remove all data from the device before initiating a Product Return and Replace (PRR). This process ensures that all sensitive customer information is permanently deleted and cannot be recovered.

Limitations of factory reset

This feature has these limitations:

- Secure erase and "factory reset and shutdown" options are not supported.
- Factory reset is supported only when the chassis SSD is present and accessible.

Locations of data removal during factory reset

This table lists the locations from which data is removed during a factory reset.

Table 17: Locations of data removal

Location	Description
Route processor	Removes logs, configuration, gISO labels, and software.

Location	Description
Chassis SSD	Removes the configuration and checkpoint data from these locations: • `/mnt/dr_disk/data/otdr_baseline` • `/mnt/dr_disk/data/config` • `/mnt/dr_disk/data/alternate_cfg` • `/mnt/dr_disk/data/chkpt`

Perform factory reset

Teaches you how to remove chassis data with the factory-reset location all command before initiating a product return and replace workflow.

Use this task to remove all data from the chassis before initiating a RMA.

Procedure

Use the **factory-reset location all** command to perform factory reset for all locations on the chassis.

Example

```
RP/0/RP0/CPU0:ios#factory-reset reload location all
Wed Sep 10 03:35:58.803 UTC
Performing factory-reset may affect the stability of the system. Re-imaging
maybe required to recover. Continue?
[confirm]
RP/0/RP0/CPU0:Sep 10 03:37:07.029 UTC: shelfmgr[350]:
%PLATFORM-SHELFMGR-4-DISK_ERASE_START :
Started disk erase operation on 0/0/NXR0
RP/0/RP0/CPU0:Sep 11 11:39:31.626 UTC: shelfmgr[197]:
%PLATFORM-SHELFMGR-4-DISK_ERASE_DONE :
 Disk erase operation finished successfully on 0/0/NXR0
```

Data is removed from the chassis.

4 Disaster Recovery

Topics:

- Disaster Recovery
- Recovering nodes after chassis SSD replacement
- Backup ISO image health checks

Details disaster recovery behavior for NCS 1014, including CPU and chassis SSD replacement considerations, node restoration workflows, and backup ISO health checks that detect corruption and support recovery.

Disaster Recovery

Details disaster recovery behavior for NCS 1014, including CPU and chassis SSD replacement considerations, node restoration workflows, and backup ISO health checks that detect corruption and support recovery.

CPU replacement considerations

Provides considerations for replacing a CPU on Cisco NCS 1014, including headless operation, software and RPM alignment, and configuration restoration from the chassis SSD.

CPU replacement behavior

You must consider these points for CPU replacement:

- When the CPU is removed from the chassis, the NCS 1014 chassis runs in headless mode, which is non-traffic impacting.
- When the CPU is replaced with another CPU that has the same software and RPMs as the chassis SSD, the configuration is restored from the chassis SSD.
- When the CPU is replaced with another CPU that has different software and RPMs than the chassis SSD, the disaster recovery process starts. In this case, the node boots with the software from the chassis SSD, and the configuration is also restored from the chassis SSD.

Recovering nodes after chassis SSD replacement

Describes how Cisco NCS 1014 recovers after chassis SSD replacement in operational and shutdown nodes, including backup, software, RPM, and configuration behavior.

The chassis SSD (NCS1K14-SSD) in Cisco NCS 1014 is a hot-swappable module, meaning you can replace it without shutting down the system.

> ⚠ **Attention**
>
> When you remove the chassis SSD while the controller is in operation, the system raises the Disaster Recovery Unavailable alarm. This alarm clears automatically after you install the new chassis SSD.

Summary

These scenarios describe how the node recovers after a chassis SSD replacement.

The key components involved in the process are:

- Controller SSD: Maintains the system's software, RPMs, and configuration data critical for node operation.
- Replacement chassis SSD: The new SSD installed in the chassis, which could have identical, different, or blank software, RPM, and configuration content.

Workflow

These stages describe how nodes recover after chassis SSD replacement.

1. If the chassis SSD is replaced while the node is up and running, the way the node recovers depends on what is present on the replacement chassis SSD

Table 18: Replacing a chassis SSD in an operational node

When you replace the chassis SSD with...	Then the system loads only...	And then the new chassis SSD takes a backup of...
another that has the same software and RPMs as the controller SSD	the configuration from the controller SSD	the system configuration from the controller SSD.
another that has the different software and RPMs from the controller SSD	the software, RPMs, and the configuration from the controller SSD	all the contents from the controller SSD.
another received from Cisco manufacturing as a spare or through a return material authorization (RMA) process, which comes with a blank configuration	the software, RPMs, and the configuration from the controller SSD, after inserting this SSD to this system	all controller SSD contents.

2. If the chassis SSD is replaced in a node that is powered down, recovery occurs as follows after the node boots:

Table 19: Replacing a chassis SSD in a shutdown node

When you replace the chassis SSD with...	Then upon powering on...	And then the system...
another that has software and RPMs different from the contents of the controller SSD	the system triggers the disaster recovery workflow	installs the software, RPMs, and configurations from the chassis SSD.

Result

Once the replacement SSD is installed, the node automatically synchronizes and restores configuration and software as needed to ensure proper operation. If contents differ, the system recovers by copying all essential files. The system clears any recovery alarms after restoration is complete.

Backup ISO image health checks

Explains how the health check feature audits the backup ISO image on Cisco NCS 1014 to support error-free booting during disaster recovery.

Backup ISO image health checks are disaster recovery checks that

- ensure error-free booting of the NCS 1014 chassis during disaster recovery operations,
- audit the chassis SSD content against the running software in the background every 12 hours to detect corruption, and
- recover a corrupted ISO image by copying it from the backup location.

Backup ISO image storage and recovery

NCS 1014 has a partition for disaster recovery where the backup ISO image is stored. The backup ISO image is stored in the chassis SSD.

The chassis SSD content is audited against the running software by the install process in the background every 12 hours to detect corruption. If the ISO image is corrupted, the software will recover it by copying from the backup location.

If the software fails to synchronize with the chassis SSD, then the Disaster Recovery ISO Image Corruption alarm is raised. See the *Troubleshooting Guide for Cisco NCS 1014* to clear the alarm.

5 Connection Verification

Topics:

- Connection verification methods
- How OLT line card connection verifications work

Details the procedures for monitoring optical power levels via photodiodes and verifying physical connections between the NCS 1010 OLT line card and the NCS1K-CCMD-16 line card. Guides users through configuring tone patterns and executing detection commands to ensure accurate cabling during node installation.

This chapter describes the tasks to verify connection between the OLT Line Card of NCS 1010 and NCS1K14-CCMD-16-C line card.

Connection verification methods

Explains how connection verification uses a dedicated tone to validate cabling between the Optical Line Terminal line card and NCS1K-CCMD-16 line cards during node installation.

A connection verification method is a cabling validation process that

- checks the connection between the Optical Line Terminal (OLT) line card and NCS1K-CCMD-16 line cards,
- prevents miscabling during node installation, and
- uses a dedicated probe signal to verify connectivity.

Connection verification signal

You can use verification tasks to validate the connection between the Cisco NCS 1010 OLT line card and the NCS1K-CCMD-16 line card during installation. The dedicated Connection Verification Tunable Laser (CV-TL) on the OLT card generates a probe signal at a specific frequency and power, which the connected CCMD-16 line card detects. This process ensures that connections are correct and helps avoid errors in cabling.

For comprehensive details about the connection verification process, refer toCisco NCS 1010 Datapath Configuration Guide.

Power data readings

Describes the photodiode power monitors and calibrated port references that help you interpret power data for NCS1K-CCMS-16 input and output ports.

Use this reference to identify the photodiode monitors, calibrated ports, port labels, and power ranges used in connection verification.

Photodiodes (PDs) are optical power monitors available on all input ports and aggregated output ports. Tone detection is enabled on some PD monitors.

Table 20: NCS1K-CCMS-16 calibrated port references

Photodiode	Port calibrated	Port label (direction)	Minimum power (dBm)	Maximum power (dBm)	Dynamic range (dBm)
PD 21	MPO-16 input ports	(TX)	-50	30	80
PD 22	MPO-16 output ports	(RX)	-50	30	80

How OLT line card connection verifications work

Describes how Optical Line Terminal (OLT) line cards use out-of-band channels and photodiode monitors to verify connections with NCS1K-CCMD-16 line cards.

Summary

Line card connection verification ensures the integrity and correct identification of optical paths between OLTs and NCS1K-CCMD-16 line cards. This process relies on specialized channels, modulated signals, and dedicated photodiode monitors.

The key components involved in the process are:

- OLT line card: Generates the verification tone and manages the initiation of the connection test.

- Connection Verification Tunable Laser (CV-TL): Produces a modulated tone at 191.175 THz, embedding the Cable ID pattern to identify the optical path.
- Photodiode monitors (PD21 and PD22 on the CCMD-16 card): Detect and buffer the Cable ID pattern from the incoming verification signals for validation by the node controller.

Connection verification between an OLT and NCS1K-CCMD-16 line cards is vital for maintaining reliable optical communication. Embedding the Cable ID in the verification signal helps identify the path under test and validates both out-of-band (OOB) and in-band (IB) connections.

Workflow

The process involves these stages:

1. Tone generation: The OLT line card initiates the process by generating a verification tone using the CV-TL tuned at 191.175 THz.
2. Cable ID modulation: The CV-TL tone is modulated with a low-frequency pattern that includes the Cable ID of the specific connection.
3. Signal routing: The modulated CV-TL signal is transmitted via the OOB channel toward the CCMD-16 line card.
4. Path identification: For verification toward the CCMD-16 card, the CV-TL signal is routed to photodiode PD21 (for OOB verification) and PD22 (for IB verification).
5. Signal detection and buffering: PD21 or PD22 detects the verification signal and buffers the encoded Cable ID pattern.
6. Verification by node controller: The node controller reads the buffered Cable ID pattern and confirms the identity and integrity of the optical path under test.

Result

This process enables automated, accurate verification of optical connections between OLTs and NCS1K-CCMD-16 line cards, ensuring each link is correctly identified and operational.

Verify connection for a CCMD-16 line card

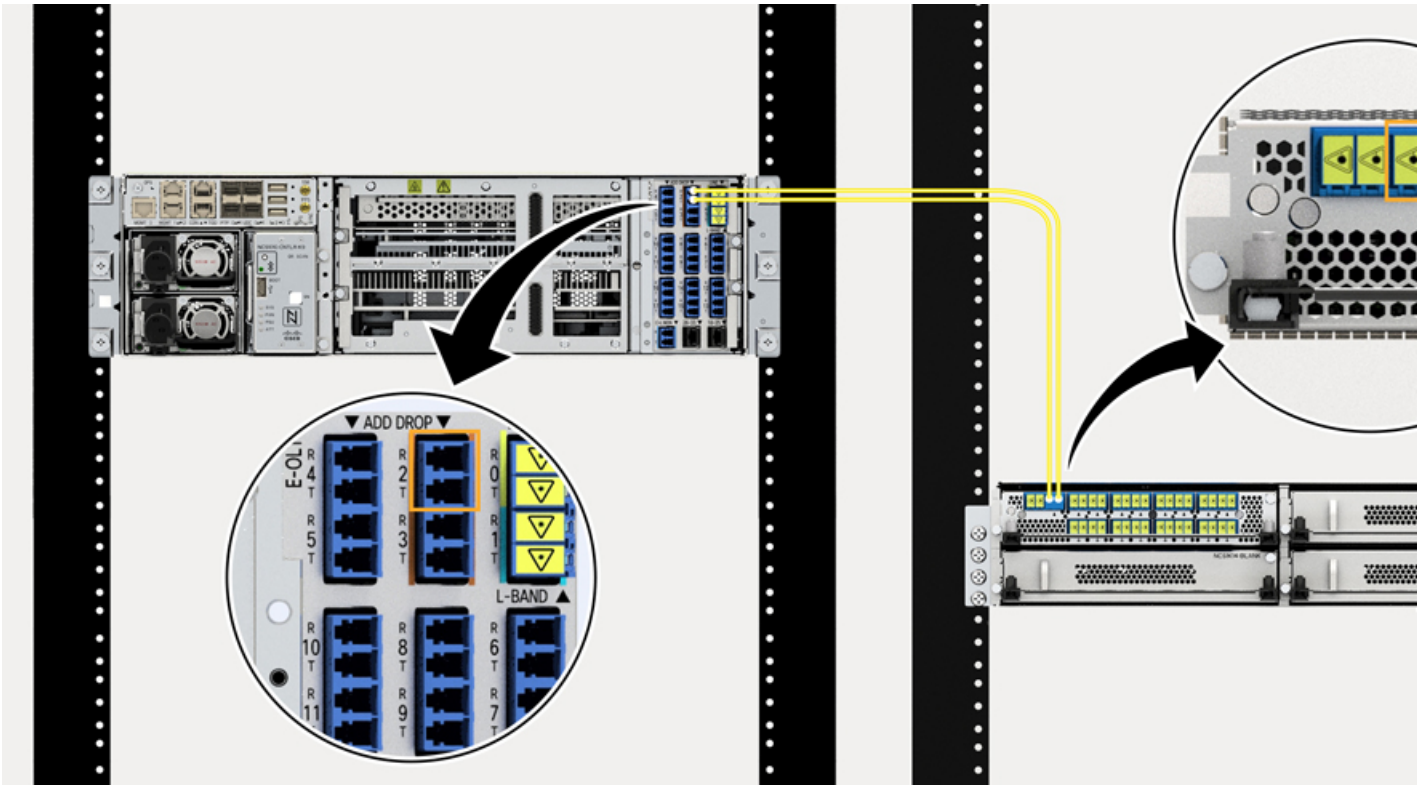
Teaches you how to verify the connection between a Cisco NCS 1010 OLT-C line card and an NCS1K-CCMD-16 line card by detecting the expected tone pattern.

Use this task to verify that the NCS 1010 OLT-C line card and the NCS1K-CCMD-16 line card are connected correctly.

The connection verification procedure checks the connection between the OLT line card and CCMD-16 line cards to match the different instances with the OLT line-card connectors.

The OLT-C line card and the NCS1K-CCMD-16 line card are connected as shown in the figure.

Figure 3: NCS 1010 and NCS1K-CCMD-16 connection



The OLT-C line card performs connection verification between the OLT-C line card and the NCS1K-CCMD-16 line card panels as described in How OLT line card connection verifications work on page 102.

The NCS1K-CCMD-16 line card is identified and verified by checking the connection verification signal at the monitors on the out-of-band (OOB) and in-band (IB) loops. PD21 monitors the OOB loop, and PD22 monitors the IB loop for the NCS1K-CCMD-16 line card.

Before you begin

Configure the Optical Transport Section (OTS) controller in Cisco NCS 1010 to generate the tone for connection verification. For details about configuring the OTS controller, see Cisco NCS 1010 Datapath Configuration Guide.

Procedure

1. Configure the Optical Multiplex Section (OMS) controller to detect the tone for connection verification.

   Example

   ```
   RP/0/RP0/CPU0:(config)#controller oms 0/1/0/0
   RP/0/RP0/CPU0:(config-Oms)#tone-rate 2
   RP/0/RP0/CPU0:(config-Oms)#tone-pattern-expected aabbccdd
   RP/0/RP0/CPU0:(config-Oms)#tone-detect-oob
   RP/0/RP0/CPU0:(config-Oms)#commit
   ```

   Configure **tone-detect-oob** on the OMS x/x/x/0 controller for NCS1K-CCMD-16.

2. Start tone-pattern detection on the OMS controller with the **tone-pattern-detect** command.

Example

This sample starts tone-pattern detection on the OMS controller.

```
RP/0/RP0/CPU0:#tone-pattern-detect controller oms 0/1/0/0 start
Tue May 10 11:38:03.775 UTC
Tone pattern detect started
```

3. Check the tone information with the **tone-info** command.

Example

This sample shows successful connection verification on the OMS controller.

```
RP/0/RP0/CPU0:#show controllers oms 0/1/0/0 tone-info
Fri Sep 22 06:04:03.787 UTC
Tone Info:
Tone Rate : 2 bits/second
Tone Pattern Expected(Hex value) : aabbccdd
Tone Pattern Received(Hex value) : aabbccdd
Tone Detected OOB :  Enabled
Detection State: Success
```

4. Stop tone-pattern detection on the OMS controller after connection verification succeeds.

Example

```
RP/0/RP0/CPU0:#tone-pattern-detect controller oms 0/1/0/0 stop
Fri Sep 22 06:23:15.165 UTC
Tone pattern detect stopped
```

The expected and received tone patterns match, and the detection state is Success.

6 System Health Check

Topics:

- System health checks
- Enable the health check
- Health check refresh cadence
- View the status of all metrics
- Metric threshold values
- View health status of an individual metric
- Disable health check

Describes how to configure and monitor the system health check service on Cisco NCS 1014, including enabling the service, changing the refresh time, customizing metric thresholds, viewing metric status, and disabling health check.

Monitoring systems in a network proactively helps prevent potential issues and take preventive actions. This chapter describes the tasks to configure and monitor system health check.

System health checks

Describes how the Cisco NCS 1014 health check service monitors physical characteristics, processing status, and resource utilization to help you assess the device condition and identify potential issues before they impact the network.

The Cisco NCS 1014 health check service is a system monitoring service that

- monitors physical characteristics, current processing status, and the currently utilized resources to assess the condition of the device at any time,
- analyzes the system health by tracking metrics that are critical for the functioning of Cisco NCS 1014, and
- is installed with the Cisco NCS 1014 RPM.

The system health metrics are thresholds set on the device to monitor the usage of CPU and other system resources.

System resource metrics states

You can evaluate the system's health by examining the metric values. If these values cross or approach the set thresholds, it suggests potential problems. By default, metrics for system resources are configured with preset threshold values. You can customize the metrics to monitor by disabling or enabling metrics of interest based on your requirement.

Each metric is tracked and compared with that of the configured threshold, and the state of the resource is classified accordingly.

The system resources metrics can be in one of these states:

- Normal: The resource usage is less than the threshold value.
- Minor: The resource usage is more than the minor threshold, but less than the severe threshold value.
- Severe: The resource usage is more than the severe threshold, but less than the critical threshold value.
- Critical: The resource usage is more than the critical threshold value.

Infrastructure service metrics states

The infrastructure services metrics can be in one of these states:

- Normal: The resource operation is as expected.
- Warning: The resource needs attention. For example, a warning is displayed when the Field-Programmable Device (FPD) needs an upgrade.

Supported system health check metrics

Cisco NCS 1014 supports the following system health check metrics:

- communication-timeout
- cpu
- filesystem
- fpd
- free-mem
- hw-monitoring
- lc-monitoring
- pci-monitoring

- platform
- process-resource
- process-status
- shared-mem
- wd-monitoring

Enable the health check

Configure the health check service on Cisco NCS 1014 by enabling the service and the Network Configuration Protocol (NETCONF) and Google Remote Procedure Call (gRPC) agents so that you can monitor system health.

Enable the health check service on Cisco NCS 1014 so that the system can monitor configured metrics and report the health state of system resources and infrastructure services.

Use this task to enable health check, allowing Cisco NCS 1014 to monitor system health.

Before you begin

Before enabling health check, ensure that:

- An IP address and subnet mask is assigned to the management interface.
- The IP address of the default gateway is configured with a static route.

For more details, see the Configure Management Interface section of the *Cisco NCS 1014 System Setup and Software Installation Guide*.

Procedure

1. Enter the configuration mode.
2. Run the **healthcheck enable** to enable health check.

   Example

   ```
   RP/0/RP0/CPU0:ios(config)#healthcheck enable
   ```

3. Run the **netconf-yang agent ssh** command.

   Example

   ```
   RP/0/RP0/CPU0:ios(config)#netconf-yang agent ssh
   ```

4. Run the **grpc local-connection** command to enable Google Remote Procedure Call (gRPC) using the command.

   Example

   ```
   RP/0/RP0/CPU0:ios(config)#grpc local-connection
   ```

5. Commit the changes using the **commit** command.

The health check service is enabled on Cisco NCS 1014. The NETCONF and gRPC agents are running, allowing you to monitor the health status of system resources and infrastructure services.

Health check refresh cadence

Explains how to adjust the cadence at which the health check status is refreshed on Cisco NCS 1014 so that the system reports metric updates at the interval that suits your monitoring requirements.

A health check refresh cadence is a configurable system parameter that

- determines how often Cisco NCS 1014 updates and reports health check status,
- defines the time interval, in seconds, between consecutive metric refreshes, and
- enables administrators to customize monitoring to match their operational requirements.

By default, the health check refresh cadence is set to 60 seconds, meaning health check status is updated every 60 seconds. You can change this interval by using the `healthcheck cadence [cadence-value]` command, where `[cadence-value]` specifies your preferred time interval in seconds.

The following example shows how to change the health check cadence value to 50 seconds so that health check status is updated every 50 seconds.

```
RP/0/RP0/CPU0:ios(config)#healthcheck cadence 50
```

View the status of all metrics

Verify the status of all supported health check metrics on Cisco NCS 1014, including configured thresholds, health check manager state, and the current health state of each enabled metric.

Check the overall health and configuration status of all metrics in the Cisco NCS 1014 system.

Use this task to view detailed information about supported health check metrics, including which metrics are enabled, their current state, threshold settings, and the health check manager's operational status.

Procedure

1. Run the **show healthcheck status** command to display which metrics are enabled, their threshold settings, and the collector cadence.

   Example

```
RP/0/RP0/CPU0:RP/0/RP0/CPU0:ios#show healthcheck status
Sat Jun 12 02:00:25.204 UTC

Healthcheck status: Enabled
Time started: 12 Jun 02:00:22.392972

Collector Cadence: 30 seconds

METRICS STATS

System Resource metrics
   cpu
       Thresholds: Minor: 20%
                   Severe: 50%
```

```
                Critical: 75%

        Tracked CPU utilization: 15 min avg utilization

   free-memory
       Thresholds: Minor: 10%
                   Severe: 8%
                 Critical: 5%

   filesystem
       Thresholds: Minor: 80%
                   Severe: 95%
                 Critical: 99%

   shared-memory
       Thresholds: Minor: 80%
                   Severe: 95%
                 Critical: 99%

Infra Services metrics
   platform

   fpd

Install Custom Metrics
   process-status

   process-resource

   communication-timeout

   pci-monitoring

   hw-monitoring

   wd-monitoring

   lc-monitoring

Use case
 Use cases are disabled
```

2. Run the `show healthcheck internal states` command to check the internal health state and configuration statuses of the health check manager. command.

   Example

```
RP/0/RP0/CPU0:ios#show healthcheck internal states
Sat Jun 12 02:00:55.425 UTC

 Internal Structure INFO

 Current state: Enabled

 Reason: Success

 Netconf Config State: Enabled

 Grpc Config State: Enabled

 Nosi state: Initialized
```

```
 Appmgr conn state: Connected

 Nosi lib state: Not ready

 Nosi client: Valid client
```

3. Run the `show healthcheck report` command to view the current health state for each enabled metric.

Example

```
RP/0/RP0/CPU0:RP/0/RP0/CPU0:ios#show healthcheck report
Sat Jun 12 02:02:54.417 UTC

Healthcheck report
Last Update Time: 12 Jun 02:02:46.955241
METRICS REPORT

cpu
  State: Normal

free-memory
  State: Normal

filesystem
  State: Normal

shared-memory
  State: Normal

platform
  State: Warning
  Reason: One or more devices are not in operational state

fpd
  State: Warning
  Reason: One or more FPDs are not in CURRENT state

process-status
  State: Normal

process-resource
  State: Normal

communication-timeout
  State: Normal

pci-monitoring
  State: Normal

hw-monitoring
  State: Normal

wd-monitoring
  State: Normal

lc-monitoring
  State: Normal
```

In this output, the state of the Field-Programmable Device (FPD) shows a warning message that indicates an FPD upgrade is required.

The system displays the status of all configured health metrics, including thresholds and current health states. You can quickly identify any warnings or critical states that may need further action.

Metric threshold values

Explains how to customize the health check threshold value for a specific metric on Cisco NCS 1014 so that the system raises alerts at resource usage levels that match your operational requirements.

A metric threshold value is a configurable system parameter that

- defines the usage level at which an alert is raised for a monitored metric,
- allows operators to tailor alerting behavior to their operational needs, and
- supports adjustment for each metric based on specific operational requirements.

healthcheck metric *metric-name* **threshold** *threshold-value*

To change the minor threshold value of the CPU metric to 25%, enter this command:

```
healthcheck metric cpu minor threshold 25%
```

In this example, the threshold for minor alerts on CPU usage is set to 25%, allowing the system to notify you when CPU usage crosses this value.

View health status of an individual metric

View the health status of a specific system resource or infrastructure service metric on Cisco NCS 1014.

Monitor and troubleshoot individual system metrics by checking their health status.

Use this task to check the health status of a chosen metric on Cisco NCS 1014 systems. This helps you focus on and address issues related to specific resources or services.

Procedure

Run the **show healthcheck metric** *metric-name* command.

Example

This example shows how to obtain the health-check status for the *filesystem* metric:

```
RP/0/RP0/CPU0:RP/0/RP0/CPU0:ios#show healthcheck metric filesystem
Sat Jun 12 02:01:32.432 UTC
Filesystem Metric State: Normal
Last Update Time: 12 Jun 02:01:04.446619
Filesystem Service State: Enabled
Number of Active Nodes: 1
Configured Thresholds:
   Minor: 80%
   Severe: 95%
   Critical: 99%
```

```
Node Name: 0/RP0/CPU0
    Partition Count: 5

    Partition Name: tftp:
        Partition Access Attribute: rw
        Partition Type: network
        Partition Size: 0
        Partition Free Bytes: 0
        Partition Free Space in %: 0

    Partition Name: disk0:
        Partition Access Attribute: rw
        Partition Type: flash-disk
        Partition Size: 20024897536
        Partition Free Bytes: 19978481664
        Partition Free Space in %: 99

    Partition Name: /misc/config
        Partition Access Attribute: rw
        Partition Type: flash
        Partition Size: 151314698240
        Partition Free Bytes: 146903269376
        Partition Free Space in %: 97

    Partition Name: harddisk:
        Partition Access Attribute: rw
        Partition Type: harddisk
        Partition Size: 150114078720
        Partition Free Bytes: 144962641920
        Partition Free Space in %: 96

    Partition Name: ftp:
        Partition Access Attribute: rw
        Partition Type: network
        Partition Size: 0
        Partition Free Bytes: 0
        Partition Free Space in %: 0
```

Example

This example shows how to obtain the health-check status for the *platform* metric:

```
RP/0/RP0/CPU0:RP/0/RP0/CPU0:ios#show healthcheck metric platform
Sat Jun 12 02:01:51.922 UTC
Platform Metric State: Warning
Last Update Time: 12 Jun 02:01:38.650003
Platform Service State: Enabled
Number of Racks: 1

Rack Name: 0
    Number of Slots: 5

    Slot Name: RP0
        Number of Instances: 1

    Instance Name: CPU0
        Node Name 0/RP0/CPU0
        Card Type NCS1K14-CNTLR-K9
        Card Redundancy State Active
        Admin State NSHUT,NMON
```

```
        Oper State IOS XR RUN

    Slot Name: PM1
        Number of Instances: 0

        Node Name 0/PM1
        Card Type NCS1K4-AC-PSU-2
        Card Redundancy State None
        Admin State NSHUT,NMON
        Oper State OPERATIONAL

    Slot Name: FT1
        Number of Instances: 0

        Node Name 0/FT1
        Card Type NCS1K14-FAN
        Card Redundancy State None
        Admin State NSHUT,NMON
        Oper State OPERATIONAL

    Slot Name: FT2
        Number of Instances: 0

        Node Name 0/FT2
        Card Type NCS1K14-FAN
        Card Redundancy State None
        Admin State NSHUT,NMON
        Oper State OPERATIONAL

    Slot Name: 2
        Number of Instances: 1

    Instance Name: NXR0
        Node Name 0/2/NXR0
        Card Type NCS1K4-1.2T-K9
        Card Redundancy State None
        Admin State NSHUT,NMON
        Oper State CARD FAILED
```

The health status for the specified metric is displayed in the CLI, allowing you to assess operational states, configured thresholds, and service health.

Disable health check

Configure Cisco NCS 1014 to disable the entire health check service or turn off health checks for individual metrics as required.

Stop monitoring the entire system or specific resources by disabling the health check service or individual health check metrics.

Health checks monitor various system and resource metrics on Cisco NCS 1014 devices. By default, all health check metrics are enabled. Disabling health checks can be necessary when you need to perform configuration changes (as some changes are blocked while health check service is active) or when you want to suspend monitoring for specific reasons.

Procedure

1. Enter the command **no healthcheck enable** to disable health checks for all metrics.

   Example

   ```
   RP/0/RP0/CPU0:#configure
   RP/0/RP0/CPU0:RP/0/RP0/CPU0:ios(config)#no healthcheck enable
   RP/0/RP0/CPU0:RP/0/RP0/CPU0:ios(config)#commit
   ```

   > **Note**
   >
   > When the health check service is enabled, other configuration changes are not permitted. Disable the service before committing configuration changes.

2. Enter the command **healthcheck metric** *metric-name* **disable** to disable health check for an individual metric.

   Example

   ```
   RP/0/RP0/CPU0:RP/0/RP0/CPU0:ios(config)#healthcheck metric free-mem disable
   ```

The health check service or specified metric's monitoring is disabled. You can now commit configuration changes if required, or resume monitoring later by re-enabling the health check service or metric.

7 Configure AAA

Topics:

- AAA administrative models

Describes how to implement task-based authorization on Cisco NCS 1014, including the configuration of TACACS+ and RADIUS servers and server groups, and the deprecation of Type 7 password and Type 5 secret.

AAA administrative models

Describes the administrative model of task-based authorization that controls user access on Cisco NCS 1014, including the configuration of TACACS+ and RADIUS servers and server groups.

Authentication, authorization, and accounting (AAA) is an administrative model that

- controls user access on Cisco NCS 1014 by validating user identity against a local database or a remote server,
- determines which tasks individual users can perform through task-based authorization, and
- records the network services users access and the resources they consume, enabling integration with TACACS+ and RADIUS servers and server groups.

> **Note**
>
> From Release 24.4.1, the AAA local database supports configuring up to 3000 usernames. You can configure more than 3000 users, but the system scale and performance are not assured beyond this limit.

Deprecation of Type 7 password and Type 5 secret

Describes the deprecation of the Type 7 password and Type 5 secret options on Cisco NCS 1014 from Release 24.4.1, and lists the CLI changes, upgrade behavior, syslog messages, and special use cases.

Password configuration options before Release 24.4.1

Until Release 24.4.1, there were two options for configuring a password:

- Password: Uses Type 7 encryption to store the password.
- Secret: Supports Type 5, 8, 9, or 10 hashing algorithms to store the password securely.

Deprecation notice

Starting from Release 24.4.1, the use of Type 7 password and Type 5 secret is deprecated due to security concerns. The deprecation process begins from Release 24.4.1, and the full deprecation is expected in a future release. Cisco recommends using the default option, which is Type 10 secret.

This topic covers the following CLI options and use cases:

- password on page 119
- masked-password on page 119
- password-policy on page 120
- aaa password-policy on page 121
- secret on page 122
- masked-secret on page 122

password

The **password** options available in CLI from Release 24.4.1:

```
RP/0/RP0/CPU0:ios(config-un)#password ?
LINE The type 7 password followed by '7 ' OR SHA512-based password (deprecated,
 use 'secret')
```

Changes:

- All the options that were present until Release 24.4.1 are removed except LINE (to accept cleartext).
- During upgrade: Any configuration using the Type 7 password is automatically converted to Type 10 secret.

  Post-upgrade: You can still use the Type 7 password configuration option after new commits, but the password is stored as Type 10 secret.
- A new syslog message is added to indicate the deprecation process:

```
%SECURITY-PSLIB-4-DEPRECATED_PASSWORD_TYPE : The password configuration is
deprecated.
 Converting it to a Type 10 secret for user <user name>.
```

- **show running configuration** command output before upgrade:

```
username example
password 7 106D000A0618
!
```

  show running configuration command output post-upgrade:

```
username example
secret 10
$6$P53pb/FFxNIT4b/.$yVakako4fp9PZiIYYh1xS0.W6b/yPrSyC8j4gLs6xli57iC1OryPXyN9y8yojRD2nhAWb9pjr/WAIhbXqp8st.
!
```

masked-password

The **masked-password** options available in CLI from Release 24.4.1:

```
RP/0/RP0/CPU0:ios(config-un)#masked-password ?
0 Specifies a cleartext password will follow
clear Config deprecated. Will be removed in 7.7.1. Specify '0' instead.
<cr> The cleartext user password
```

Changes:

- The options 7 and encrypted that were present until Release 24.4.1 are removed.
- During upgrade: Any configuration using the Type 7 password is automatically converted to Type 10 secret.
- Post-upgrade: Masked-password is an alternate method of configuring the password. You can still use the masked-password keyword with a clear string after new commits, but the password is stored as Type 10 secret.
- A new syslog message is added to indicate the deprecation process:

```
%SECURITY-PSLIB-4-DEPRECATED_PASSWORD_TYPE : The password configuration is
deprecated.
 Converting it to a Type 10 secret for user <user name>.
```

- **show running configuration** command output before upgrade:

```
username example
password 7 106D000A0618
!
```

show running configuration command output post-upgrade:

```
username example
secret 10
$6$P53pb/FFxNIT4b/.$yVakako4fp9PZiIYYh1xS0.W6b/yPrSyC8j4gLs6xli57iC1OryPXyN9y8yojRD2nhAWb9pjr/WAIhbXqp8st.
!
```

password-policy

The **password-policy** options available in CLI from Release 24.4.1:

```
RP/0/RP0/CPU0:ios(config-un)#password-policy ?
WORD Specify the password policy name

RP/0/RP0/CPU0:ios(config-un)#password-policy abcd password ?
0 Specifies an UNENCRYPTED password will follow
7 Specifies that an encrypted password will follow
LINE The UNENCRYPTED (cleartext) user password
clear Config deprecated. Will be removed in 7.7.1. Specify '0' instead.
encrypted Config deprecated. Will be removed in 7.7.1. Specify '7' instead.
```

Changes:

- All the options that were present until Release 24.4.1 are removed except LINE (to accept cleartext).
- During upgrade: Any configuration using the Type 7 password is automatically converted to Type 10 secret.

  Post-upgrade: You can still use the password-policy configuration option after new commits, but it is stored as Type 10 secret.
- A new syslog message is added to indicate the deprecation process:

```
%SECURITY-PSLIB-4-DEPRECATED_PASSWORD_TYPE : The password configuration is
deprecated.
 Converting it to a Type 10 secret for user <username>.
```

- **show running configuration** command output before upgrade:

```
username example
password-policy abcd password 7 106D000A0618
!
```

show running configuration command output post-upgrade:

```
username example
secret 10
$6$P53pb/FFxNIT4b/.$yVakako4fp9PZiIYYh1xS0.W6b/yPrSyC8j4gLs6xli57iC1OryPXyN9y8yojRD2nhAWb9pjr/WAIhbXqp8st.
!
!
```

aaa password-policy

The **aaa password-policy** options available in CLI from Release 24.4.1:

```
RP/0/RP0/CPU0:ios(config)#aaa password-policy abcd
RP/0/RP0/CPU0:ios(config-pp)#?
min-char-change Number of characters change required between old and new
passwords (deprecated, will be removed in 25.3.1)
restrict-password-advanced Advanced restrictions on new password (deprecated,
 will be removed in 25.3.1)
restrict-password-reverse Restricts the password to be same as reversed old
password (deprecated, will be removed in 25.3.1)
```

Changes:

- The options min-char-change, restrict-password-advanced, and restrict-password-reverse that were present until Release 24.4.1 are deprecated.
- During upgrade: These deprecated configurations do not change during upgrade.

  Post-upgrade: These deprecated keywords do not take effect when configured post-upgrade.
- New syslog messages are added to indicate the deprecation process:

  ```
  %SECURITY-LOCALD-4-DEPRECATED_PASSWORD_POLICY_OPTION : The password policy
   option 'min-char-change' is deprecated.
  Password/Secret will not be checked against this option now.
  ```

  ```
  %SECURITY-LOCALD-4-DEPRECATED_PASSWORD_POLICY_OPTION : The password policy
   option 'restrict-password-reverse' is deprecated.
   Password/Secret will not be checked against this option now.
  ```

  ```
  %SECURITY-LOCALD-4-DEPRECATED_PASSWORD_POLICY_OPTION : The password policy
   option 'restrict-password-advanced' is deprecated.
  Password/Secret will not be checked against this option now.
  ```

- **show running configuration** command output before upgrade:

  ```
  aaa password-policy abcd
  lower-case 3
  min-char-change 1
  restrict-password-reverse
  restrict-password-advanced
  !
  ```

  show running configuration command output post-upgrade:

  ```
  aaa password-policy abcd
  lower-case 3
  min-char-change 1
  restrict-password-reverse
  restrict-password-advanced
  !
  ```

secret

The **secret** options available in CLI from Release 24.4.1:

```
RP/0/RP0/CPU0:ios(config-un)#secret ?
0 Specifies a cleartext password will follow
10 Specifies that SHA512-based password will follow
8 Specifies that SHA256-based password will follow
9 Specifies that Scrypt-based password will follow
LINE The cleartext user password
```

```
RP/0/RP0/CPU0:ios(config-un)#secret 0 enc-type ?
<8-10> Specifies which algorithm to use. Only 8,9,10 supported [Note: Option
'5' is not available to use from 24.4]
```

Changes:

- The options 5 and encrypted are removed.
- During upgrade: Configurations using Type 5 secret remain unchanged.

  Post-upgrade: Although the keyword 5 has been deprecated, you can still apply the existing configurations using Type 5 secret.
- A new syslog message is added to indicate the deprecation process:

  ```
  %SECURITY-LOCALD-2-DEPRECATED_SECRET_TYPE : Type 5 secret is deprecated.
  Please use the 'secret' keyword with option type 10 for user.
  ```
- **show running configuration** command output before upgrade:

  ```
  username example
  secret 5 $1$kACo$2RtpcwyiRuRB/DhWzabfU1
  !
  !
  ```

  show running configuration command output post-upgrade:

  ```
  username example
  secret 5 $1$kACo$2RtpcwyiRuRB/DhWzabfU1
  !
  !
  ```

masked-secret

The **masked-secret** options available in CLI from Release 24.4.1:

```
RP/0/RP0/CPU0:ios(config-un)#masked-secret ?
0 Specifies a cleartext password will follow
10 Specifies that SHA512-based password will follow
8 Specifies that SHA256-based password will follow
9 Specifies that Scrypt-based password will follow
clear Config deprecated. Will be removed in 7.7.1. Specify '0' instead.
<cr> The cleartext user password
```

Changes:

- The options 5 and encrypted are removed.
- During upgrade: Configurations using masked-secret with Type 5 remain unchanged.

Post-upgrade: Although the keyword 5 has been deprecated, you can still apply the existing configurations using Type 5 masked secret.

- A new syslog message is added to indicate the deprecation process:

```
%SECURITY-LOCALD-2-DEPRECATED_SECRET_TYPE : Type 5 secret is deprecated.
Please use the 'secret' keyword with option type 10 for user.
```

- **show running configuration** command output before upgrade:

```
username example
secret 5 $1$kACo$2RtpcwyiRuRB/DhWzabfU1
!
!
```

show running configuration command output post-upgrade:

```
username example
secret 5 $1$kACo$2RtpcwyiRuRB/DhWzabfU1
!
!
```

Special use cases

Use case 1: Configurations using both Type 7 password and secret with 8, 9, or 10 hashing, for the same user

- During upgrade:
  - For the first 3000 username configurations, the password configuration is rejected, and the secret configuration remains unchanged.
  - For the rest of the username configurations, the original secret configuration is rejected, and the password is converted to Type 10 secret.
- Post-upgrade:
  - For a new username, or a username that already existed before the upgrade, the password configuration is rejected.
  - A new syslog message is added to indicate the deprecation process:

    ```
    %SECURITY-PSLIB-4-SECRET_CONFIG_PRESENT : The password configuration is
    deprecated.
    Once secret is configured, cannot use password config for user <user name>
     at index <x> now.
    ```

    where 'x' is a number representing the index.

Use case 2: Configurations using both Type 7 password and Type 5 secret, for the same user

- During upgrade:
  - For any username configuration, the original Type 5 secret configuration is rejected, and the password is converted to Type 10 secret.
- Post-upgrade:
  - For a new username, or a username that already existed before the upgrade, the password configuration is converted to Type 10 secret.

- A new syslog message is added to indicate the deprecation process:

```
%SECURITY-PSLIB-4-DEPRECATED_PASSWORD_TYPE : The password configuration
is deprecated.
 Converting it to a Type 10 secret for user <username>.
```

TACACS+ applications

Describes how the Terminal Access Controller Access Control System Plus (TACACS+) application centralizes user validation on Cisco NCS 1014 to enhance security, accounting, and administrative control.

A TACACS+ application is a security application that

- centralizes user validation for Cisco NCS 1014,
- uses AAA commands and can be enabled and configured on Cisco NCS 1014, and
- provides detailed accounting information and flexible administrative control over user access.

When a TACACS+ server is configured and the protocol is enabled on the node, Cisco NCS 1014 authenticates user credentials through the TACACS+ server:

- When a user attempts to log in to the node, Cisco NCS 1014 forwards the username and password to the configured TACACS+ servers and receives the authentication status.
- If authentication fails through the TACACS+ server, Cisco NCS 1014 sends the credentials to the node and authenticates them against the local node.
- If authentication fails against the node, Cisco NCS 1014 does not allow the user to log in.

Configure a TACACS+ server

Configures Authentication, Authorization, and Accounting (AAA) accounting on Cisco NCS 1014 to track the network services that users access and to send the activity records to the TACACS+ security server.

Enable AAA accounting on Cisco NCS 1014 so that the switch sends start-record and stop-record accounting notices to the TACACS+ server for each privileged EXEC process.

When AAA accounting is enabled, Cisco NCS 1014 tracks the network services that users access and the amount of network resources that they consume. The switch sends this user activity data to the TACACS+ security server as accounting records. Each record contains attribute-value pairs and is saved on the security server for later analysis, network management, client billing, or auditing.

Follow these steps to configure the TACACS+ server.

Procedure

1. Enter the Cisco IOS XR configuration mode.

   Example

   ```
   RP/0/RP0/CPU0:ios#configure
   ```

2. Enable TACACS+ accounting to send a start-record accounting notice at the beginning of a privileged EXEC process and a stop-record at the end.

Example

```
RP/0/RP0/CPU0:ios(config)#aaa accounting exec default start-stop group
TACACS_ALL
```

3. Create a default command accounting method list for accounting services provided by the TACACS+ security server. Configure the list for privilege-level commands with a stop-only restriction.

Example

```
RP/0/RP0/CPU0:ios(config)#aaa accounting exec default start-stop group
TACACS_ALL
```

Cisco NCS 1014 sends AAA accounting records for each privileged EXEC process to the configured TACACS+ server.

Configure TACACS+ server groups

Configures AAA server groups on Cisco NCS 1014 to group existing TACACS+ server hosts so that a subset of the configured server hosts can be referenced from AAA method lists for authentication, authorization, and accounting.

Group TACACS+ server hosts on Cisco NCS 1014 into named server groups that AAA method lists can reference when configuring authentication, authorization, or accounting.

Configuring AAA server groups on Cisco NCS 1014 lets you create named groups of TACACS+ server hosts. You can select a subset of configured server hosts and use them for a particular service. Each server group refers to specific host IP addresses and works alongside the global server-host list. Once configured, these groups can be referenced in AAA method lists during authentication, authorization, or accounting.

Before you begin

- Ensure the external TACACS+ server is accessible during configuration.
- If you configure the same IP address for global configuration, provide server-private parameters

Procedure

---

1. Enter the Cisco IOS XR configuration mode.

Example

```
RP/0/RP0/CPU0:ios# configure
```

2. Create an AAA server group and enter the server group sub-configuration mode.

Example

```
RP/0/RP0/CPU0:ios(config)# aaa group server tacacs+ tacgroup1
```

3. Configure the IP address of the private TACACS+ server for the group server.

Example

```
RP/0/RP0/CPU0:ios(config-sg-tacacs+)# server-private 10.1.1.1 port 49 key
a_secret
```

> **Note**
>
> - You can configure a maximum of 10 TACACS+ private servers in a server group.
> - If private server parameters are not specified, global configurations are used. If global configurations are not specified, default values are used.

4. Configure the authentication and encryption key used between Cisco NCS 1014 and the TACACS+ daemon running on the TACACS+ server. If no key string is specified, the global value is used.

   Example

   ```
   RP/0/RP0/CPU0:ios(config-sg-tacacs+)# key 7 08984B1A4D0C19157A5F57
   ```

5. Configure the timeout value that sets the length of time the AAA server waits to receive a response from the TACACS+ server.

   Example

   ```
   RP/0/RP0/CPU0:ios(config-sg-tacacs-private)# timeout 4
   ```

6. Repeat steps 3 through 5 for every private server that you want to add to the server group.

The TACACS+ server group is available for AAA method lists on Cisco NCS 1014.

About RADIUS

Describes how Remote Authentication Dial-In User Service (RADIUS) provides protection against unauthorized access in distributed client-server networks that include Cisco NCS 1014.

Remote Authentication Dial-In User Service (RADIUS) is a distributed client-server security system that:

- Provides protection against unauthorized access in distributed client-server networks.
- Runs RADIUS clients on Cisco NCS 1014 that send authentication and accounting requests to a central RADIUS server.
- Stores all user authentication and network service access information on the central RADIUS server.

RADIUS in the Cisco AAA framework

The Cisco AAA security model supports RADIUS along with other security protocols, such as TACACS+, Kerberos, and local username lookup.

Configure RADIUS server groups

Configures RADIUS server groups on Cisco NCS 1014 so that AAA method lists can reference the groups during authentication, authorization, and accounting.

Group external RADIUS servers on Cisco NCS 1014 into named server groups that AAA method lists can reference for authentication, authorization, or accounting.

You can enter one or more server commands to specify the hostname or IP address of an external RADIUS server, along with port numbers. After the server group is configured, AAA method lists can reference the group during authentication, authorization, or accounting. You can configure a maximum of 30 servers and 30 private servers per RADIUS server group.

Before you begin

The external RADIUS server must be accessible at the time of configuration.

Procedure

1. Enter the Cisco IOS XR configuration mode.

   Example

   ```
   RP/0/RP0/CPU0:ios# configure
   ```

2. Group different server hosts into a distinct list and enter the server group configuration mode.

   Example

   ```
   RP/0/RP0/CPU0:ios(config)# aaa group server radius radgroup1
   ```

3. Specify the hostname or IP address of the RADIUS server host.

   Example

   ```
   RP/0/RP0/CPU0:ios(config)# radius-server host 192.168.20.0
   ```

4. Specify the User Datagram Protocol (UDP) destination port for authentication requests. If you set the port to 0, the host is not used for authentication. If you do not specify the port, it defaults to 1645.

   Example

   ```
   RP/0/RP0/CPU0:ios(config)#auth-port 1812
   ```

5. Specify the UDP destination port for accounting requests. If you set the port to 0, the host is not used for accounting. If you do not specify the port, it defaults to 1646.

   Example

   ```
   RP/0/RP0/CPU0:ios(config)# acct-port 1813
   ```

6. Specify the authentication and encryption key used between Cisco NCS 1014 and the RADIUS server. This key overrides the global setting of the **radius-server key** command. If no key string is specified, the global value is used.

   Example

   ```
   RP/0/RP0/CPU0:ios(config-radius-host)#key 7 08984B1A4D0C19157A5F57
   ```

   The key is a text string that must match the encryption key used on the RADIUS server. Always configure the key as the last item in the **radius-server host** command syntax. Leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks themselves are part of the key.

7. Repeat steps 4 through 6 for every external RADIUS server that you want to add to the server group.
8. Specify the default method list for authentication and enable authentication for the console in global configuration mode.

   Example

```
RP/0/RP0/CPU0:ios(config-radius-host)#aaa authentication login default group
 radius local
RP/0/RP0/CPU0:ios(config-radius-host)#commit
```

9. (Optional) Display information about each RADIUS server group that is configured in the system.

   Example

```
RP/0/RP0/CPU0:ios# show radius server-groups
```

The RADIUS server group is available for AAA method lists on Cisco NCS 1014.

8 Access Control Lists

Topics:

- Access control lists
- ACL configuration procedures

This chapter describes Access Control Lists (ACL) configuration for Cisco NCS 1014, including ACL behavior, rule application, IPv4 and IPv6 tasks, and verification guidance.

Access control lists

This section explains access control lists and the key attributes you need to understand before configuring IPv4 or IPv6 packet filtering.

Access control lists is a Cisco NCS 1014 feature or capability that

- supports the related system setup or management workflow,
- defines behavior that affects Cisco NCS 1014 operation, and
- provides context before you configure or verify the feature.

Access control lists details
Table 21: Feature History

Feature Name	Release Information	Feature Description
ACL on Management Port	Cisco IOS XR Release 7.11.1	Access Control List (ACL) feature enables you to permit or deny specific devices to connect to the management port and access NCS 1014 devices. This control enhances network security. Both IPv4 and IPv6 ACLs are supported on the management port. Commands added: • **ipv4-access-list** • **ipv4-access-group** • **show access-lists-ipv4** • **ipv6-access-list** • **ipv6-access-group** • **show access-lists-ipv6**

Access Control Lists (ACLs) perform packet filtering to control the packets that move through the network. These controls allow to limit the network traffic and restrict the access of users and devices to the network. ACLs have many uses, and therefore many commands accept a reference to an access list in their command syntax. An ACL consists of one or more access control entries (ACE) that collectively define the network traffic profile. Access control entries (ACE) are entries in an ACL that describe the access rights related to a particular security identifier or user.

There are 2 types of ACLs:

- Standard ACLs: Verifies only the source IP address of the packets. Traffic is controlled by the comparison of the address or prefix configured in the ACL, with the source address found in the packet.
- Extended ACLs: Verifies more than just the source address of the packets. Attributes such as destination address, specific IP protocols, User Datagram Protocol (UDP) or Transmission Control Protocol (TCP) port numbers, Differentiated Services Code Point (DSCP), and so on are validated. Traffic is controlled by a comparison of the attributes stated in the ACL with those in the incoming or outgoing packets.

Purpose of ACLs

ACLs allows you to:

- Filter incoming or outgoing packets on an interface.
- Restrict the contents of routing updates.
- Limit debug output that is based on an address or protocol.
- Control vty access.

Best practices for ACL workflows

Outlines the best practices for efficiently and securely designing and managing ACL workflows.

To ensure effective and reliable ACL workflow management, follow these best practices:

- Define policies centrally using a policy management system to maintain consistency across all network devices.
- Regularly audit and document ACL changes to track compliance and simplify troubleshooting.
- Automate ACL updates to minimize manual errors and configuration drift.
- Use descriptive naming conventions for ACLs and rules to make management and audits easier.
- Test new or modified ACLs in a staged environment before deploying to production.
- Remove obsolete or unused ACL entries periodically to reduce attack surface and avoid unnecessary complexity.
- Clearly distinguish between permit and deny rules to avoid unintentional service disruptions.
- Provide training and documentation for all administrators involved in ACL management.

How access control lists operate

Describes how access control lists (ACLs) filter network traffic, the components involved, and the workflow for enforcing permit and deny statements.

Access control lists (ACLs) function as network filters that allow administrators to control incoming and outgoing traffic based on defined permit and deny rules. They are essential for implementing security policies and managing access to network resources.

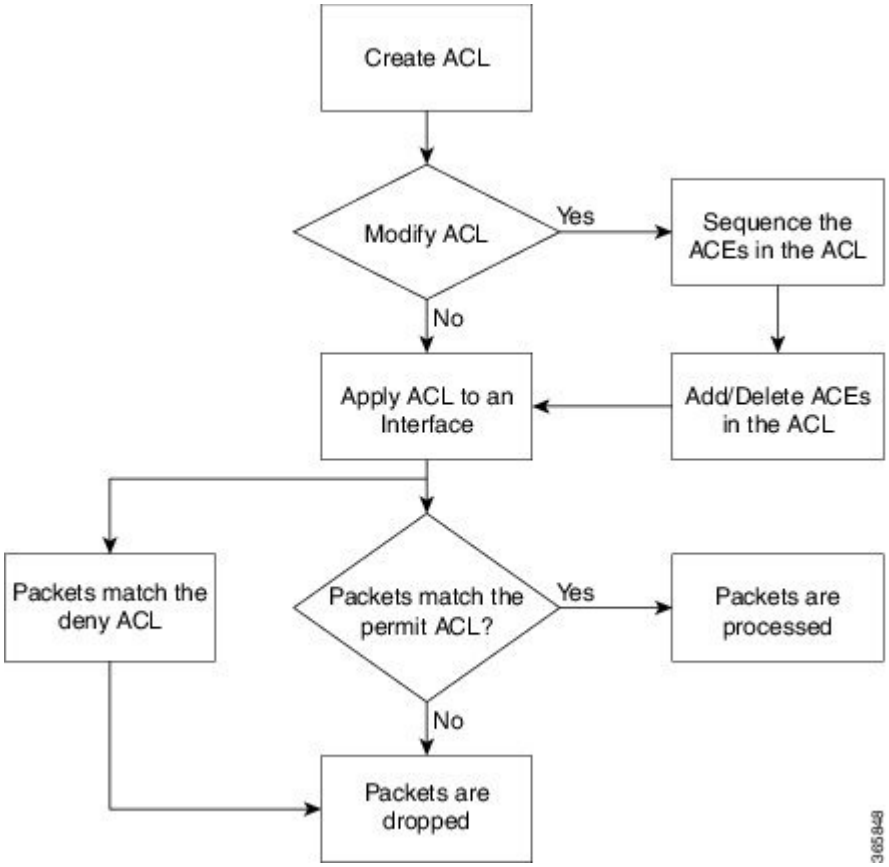
Summary

The key components involved in the process are:

- ACL configuration: Sequential permit and deny statements referencing IP addresses and protocols.
- ACL identifier: A unique name assigned to each ACL for referencing in software commands.
- Router interfaces: Points where ACLs are applied to control traffic entering or leaving the device.

Workflow

Figure 4: ACL Workflow



The process involves the following stages:

1. ACL creation: The administrator defines the ACL with a unique name and a list of permit and deny statements based on source/destination IP addresses and transport-layer protocols (such as TCP, UDP, ICMP, IGMP).
2. ACL application: The ACL is associated with an interface or feature by referencing its name in configuration commands.
3. Packet filtering: As packets arrive at or leave the router interface, the ACL evaluates each packet against its sequential list of rules. The first matching rule determines whether the packet is permitted or denied.
4. Traffic control: Only permitted packets are processed or forwarded; denied packets are blocked.
5. Modification: The administrator may update ACL rules as needed, even while the ACL is attached to an interface. ACLs can be referenced by multiple commands.
6. Review: Remarks and organization of rules, placing more specific conditions before general ones, make ACL management more effective and transparent.

Result

This process enables precise control over network access, improves security, and ensures that only authorized traffic passes through the network device interfaces.

Access control list application behavior

This section provides details about access control list application behavior, including ACL references, inbound and outbound packet checks, and undefined ACL handling on terminal lines and network interfaces.

Review how Cisco IOS XR software applies access control lists to terminal lines and network interfaces after you create an ACL.

Access control list application behavior depends on these conditions.

- ACL creation and reference requirements.

- Inbound and outbound packet processing.
- Undefined ACL handling.

Table 22: ACL application behavior

If	Then
You create an ACL.	You must reference the ACL to make it work. You can apply an ACL to outbound or inbound interfaces. This behavior applies to terminal lines and network interfaces.
An inbound ACL receives a packet.	Cisco IOS XR software checks the source address of the packet against the ACL. If the ACL permits the address, the software continues to process the packet. If the ACL rejects the address, the software discards the packet and returns an ICMP host unreachable message. The ICMP message is configurable.
An outbound ACL receives a packet that is routed to a controlled interface.	Cisco IOS XR software checks the source address of the packet against the ACL. If the ACL permits the address, the software sends the packet. If the ACL rejects the address, the software discards the packet and returns an ICMP host unreachable message.
You apply an ACL that has not yet been defined to an interface.	Cisco IOS XR software acts as if the ACL has not been applied to the interface and accepts all packets. Note this behavior if you use undefined ACLs as a means of security in your network.

ACL configuration procedures

Provides links to ACL configuration procedures for Cisco NCS 1010, including IPv4, IPv6, extended ACL, and ACL modification tasks.

Use this reference to choose the correct ACL configuration task for the required traffic direction, IP version, or access-list operation.

The ACL configuration procedures include the following tasks.

- For inbound IPv4 filtering on the management Ethernet interface, see Configure an ingress IPv4 ACL on management Ethernet interface.
- For outbound IPv4 filtering on the management Ethernet interface, see Configure an egress IPv4 ACL on the management Ethernet interface.
- For inbound IPv6 filtering on the management Ethernet interface, see Configure an ingress IPv6 ACL on the management Ethernet interface.
- For outbound IPv6 filtering on the management Ethernet interface, see Configure an egress IPv6 ACL on the management Ethernet interface.
- For ACL rules that match destination address, protocol, port, or DSCP attributes, see Configure extended access lists.
- For updates to an existing ACL, see Modify ACLs.

Configure an ingress IPv4 ACL on management Ethernet interface

Configure an ingress IPv4 ACL on the management Ethernet interface so that inbound IPv4 traffic follows defined rules.

Configure an ingress IPv4 ACL on management Ethernet interface so that Cisco NCS 1014 uses the required settings for the selected system setup, management, or routing workflow.

Before you begin

Follow these steps to configure an ingress IPv4 ACL on management Ethernet interface:

Procedure

1. Assign the management Ethernet interface with an IPv4 address.

   Example

```
/* Configure mgmtEth interface with an IPv4 address */
RP/0/RP0/CPU0:ios#configure
Thu Oct 19 17:30:23.719 UTC
RP/0/RP0/CPU0:ios(config)#
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/0
RP/0/RP0/CPU0:ios(config-if)#interface mgmtEth 0/RP0/CPU0/2
RP/0/RP0/CPU0:ios(config-if)#ipv4 address 198.51.100.247 255.255.255.0
RP/0/RP0/CPU0:ios(config-if)#no shutdown
RP/0/RP0/CPU0:ios(config-if)#commit
Thu Oct 19 17:31:25.127 UTC
RP/0/RP0/CPU0:ios(config-if)#exit
```

2. Enter the **show ipv4 interface brief** command to verify if the interface is up.

   Example

```
/* Verify if the interface is up */
RP/0/RP0/CPU0:ios(config)#do show ipv4 interface brief
Thu Oct 19 17:32:10.998 UTC

Interface                      IP-Address      Status          Protocol
Vrf-Name
Loopback0                      192.0.2.89      Up              Up       default

GigabitEthernet0/0/0/0         198.51.100.1    Up              Up       default

GigabitEthernet0/0/0/2         198.51.100.10   Up              Up       default

MgmtEth0/RP0/CPU0/0            198.51.100.247  Up              Up       default

PTP0/RP0/CPU0/0                unassigned      Shutdown        Down     default

MgmtEth0/RP0/CPU0/1            192.0.2.121     Up              Up       default

PTP0/RP0/CPU0/1                unassigned      Shutdown        Down     default

MgmtEth0/RP0/CPU0/2            192.0.2.1       Down            Down     default
```

3. Enter the **ipv4 access-list V4-ACL-INGRESS** command to configure an IPv4 ingress ACL.

Example

```
/* Configure an IPv4 ingress ACL */
RP/0/RP0/CPU0:ios(config)# ipv4 access-list V4-ACL-INGRESS
RP/0/RP0/CPU0:ios(config-ipv4-acl)# 10 permit tcp 192.0.2.2 255.255.255.0 any
RP/0/RP0/CPU0:ios(config-ipv4-acl)# 20 deny udp any any
RP/0/RP0/CPU0:ios(config-ipv4-acl)# 30 permit ipv4 192.0.2.64 255.255.255.0
any
RP/0/RP0/CPU0:ios(config-ipv4-acl)# commit
Thu Oct 19 18:31:25.127 UTC
```

4. Enter the **show access-lists ipv4** command to verify the ingress ACL creation.

Example

```
/* Verify the ingress ACL creation */
RP/0/RP0/CPU0:ios(config)# do show access-lists ipv4
Thu Oct 19 18:32:25.127 UTC
...
ipv4 access-list V4-ACL-INGRESS
 10 permit tcp 192.0.2.2 255.255.255.0 any
 20 deny udp any any
 30 permit ipv4 192.0.2.64 255.255.255.0 any
```

5. Enter the **interface mgmtEth 0/RP0/CPU0/0** command to apply the ingress ACL to the mgmtEth interface.

Example

```
/* Apply the ingress ACL to the mgmtEth interface */
RP/0/RP0/CPU0:ios(config)# interface mgmtEth 0/RP0/CPU0/0
RP/0/RP0/CPU0:ios(config-if)# ipv4 access-group V4-ACL-INGRESS ingress
RP/0/RP0/CPU0:ios(config-if)#commit
Thu Oct 19 18:34:25.127 UTC
RP/0/RP0/CPU0:ios(config-if)#exit
```

6. Enter the **show ipv4 interface mgmtEth 0/RP0/CPU0/0** command to verify if the ingress ACL has been successfully applied to the interface.

Example

```
/* Verify if the ingress ACL has been successfully applied to the interface
*/

RP/0/RP0/CPU0:ios#show ipv4 interface mgmtEth 0/RP0/CPU0/0
Fri Oct 20 05:07:06.383 UTC
MgmtEth0/RP0/CPU0/0 is Up, ipv4 protocol is Up
  Vrf is default (vrfid 0x60000000)
  Internet address is 198.51.100.247/16
  MTU is 1514 (1500 is available to IP)
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  common access list is not set, access list is V4-ACL-INGRESS
  Proxy ARP is disabled
  ICMP redirects are never sent
```

```
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  Table Id is 0xe0000000
```

You have successfully configured an IPv4 ingress ACL on the mgmtEth interface.

Configure an egress IPv4 ACL on the management Ethernet interface

Configure an egress IPv4 ACL on the management Ethernet interface so that outbound IPv4 traffic follows defined rules.

Configure an egress IPv4 ACL on the management Ethernet interface so that Cisco NCS 1014 uses the required settings for the selected system setup, management, or routing workflow.

Before you begin

Follow these steps to configure an egress IPv4 ACL on the management Ethernet interface:

Procedure

1. Assign the management Ethernet interface with an IPv4 address.

   Example

```
/* Configure mgmtEth interface with an IPv4 address */
RP/0/RP0/CPU0:ios#configure

Thu Oct 19 17:30:23.719 UTC
RP/0/RP0/CPU0:ios(config)#
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/0
RP/0/RP0/CPU0:ios(config-if)#ipv4 address 198.51.100.247 255.255.255.0
RP/0/RP0/CPU0:ios(config-if)#no shutdown
RP/0/RP0/CPU0:ios(config-if)#commit
Thu Oct 19 17:31:25.127 UTC
RP/0/RP0/CPU0:ios(config-if)#exit
```

2. Enter the **show ipv4 interface brief** command to verify if the interface is up.

   Example

```
/* Verify if the interface is up */
RP/0/RP0/CPU0:ios(config)#do show ipv4 interface brief
Thu Oct 19 17:32:10.998 UTC

Interface                      IP-Address      Status          Protocol
Vrf-Name
Loopback0                   192.0.2.89      Up              Up       default

GigabitEthernet0/0/0/0       198.51.100.1    Up              Up       default

GigabitEthernet0/0/0/2       198.51.100.10   Up              Up       default

MgmtEth0/RP0/CPU0/0          198.51.100.247  Up              Up       default

PTP0/RP0/CPU0/0              unassigned      Shutdown        Down     default
```

```
MgmtEth0/RP0/CPU0/1           192.0.2.121     Up             Up       default

PTP0/RP0/CPU0/1              unassigned      Shutdown       Down     default

MgmtEth0/RP0/CPU0/2           192.0.2.1       Down           Down     default
```

3. Enter the **ipv4 access-list** *name* command to configure an IPv4 egress ACL.

   Example

```
/* Configure an IPv4 egress ACL */
RP/0/RP0/CPU0:ios(config)# ipv4 access-list V4-ACL-EGRESS
RP/0/RP0/CPU0:ios(config-ipv4-acl)# 10 permit ipv4 203.0.113.1 255.255.255.0
 192.0.2.1 0.255.255.255
RP/0/RP0/CPU0:ios(config-ipv4-acl)# 20 deny udp any any
RP/0/RP0/CPU0:ios(config-ipv4-acl)# commit
Thu Oct 19 18:31:25.127 UTC
```

4. Enter the **show access-lists ipv4** command to verify the egress ACL creation.

   Example

```
/* Verify the egress ACL creation */
RP/0/RP0/CPU0:ios(config)# do show access-lists ipv4
Thu Oct 19 18:32:25.127 UTC
ipv4 access-list V4-ACL-EGRESS
 10 permit ipv4 203.0.113.1 255.255.255.0 192.0.2.1 255.255.255.0
 20 deny ipv4 any any
...
```

5. Enter the **interface mgmtEth 0/RP0/CPU0/0** command to apply the egress ACL to the management Ethernet interface.

   Example

```
/* Apply the egress ACL to the mgmtEth interface */
RP/0/RP0/CPU0:ios(config)# interface mgmtEth 0/RP0/CPU0/0
RP/0/RP0/CPU0:ios(config-if)# ipv4 access-group V4-ACL-EGRESS egress
RP/0/RP0/CPU0:ios(config-if)# commit
Thu Jul 11 09:19:49.569 UTC
RP/0/RP0/CPU0:ios(config-if)# exit
```

6. Enter the **show ipv4 interface mgmtEth 0/RP0/CPU0/0** command to verify if the egress ACL has been successfully applied to the interface.

   Example

```
/* Verify if the egress ACL has been successfully applied to the interface
*/

RP/0/RP0/CPU0:ios#show ipv4 interface mgmtEth 0/RP0/CPU0/0
Fri Oct 20 05:07:06.383 UTC
```

```
MgmtEth0/RP0/CPU0/0 is Up, ipv4 protocol is Up
  Vrf is default (vrfid 0x60000000)
  Internet address is 198.51.100.247/16
  MTU is 1514 (1500 is available to IP)
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is V4-ACL-EGRESS
  Inbound  common access list is not set, access list is not set
  Proxy ARP is disabled
  ICMP redirects are never sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  Table Id is 0xe0000000
```

You have successfully configured an IPv4 egress ACL on the mgmtEth interface.

Configure an ingress IPv6 ACL on the management Ethernet interface

Configure an ingress IPv6 ACL on the management Ethernet interface so that inbound IPv6 traffic follows defined rules.

Configure an ingress IPv6 ACL on the management Ethernet interface so that Cisco NCS 1014 uses the required settings for the selected system setup, management, or routing workflow.

Before you begin

Follow these steps to configure an ingress IPv6 ACL on the management Ethernet interface:

Procedure

1. Assign the management Ethernet interface with an IPv6 address.

   Example

   ```
   /* Configure a mgmtEth interface with an IPv6 address */
   RP/0/RP0/CPU0:ios#
   RP/0/RP0/CPU0:ios#configure
   Fri Oct 20 05:25:58.785 UTC
   RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/1
   RP/0/RP0/CPU0:ios(config-if)#ipv6 address 2001::1/64
   RP/0/RP0/CPU0:ios(config-if)#no shutdown
   RP/0/RP0/CPU0:ios(config-if)#commit
   Fri Oct 20 05:26:13.669 UTC
   RP/0/RP0/CPU0:ios(config-if)#exit
   ```

2. Enter the **show ipv6 interface mgmtEth 0/RP0/CPU0/1 brief** command to verify if the interface is up.

   Example

   ```
   /* Verify if the interface is up */
   RP/0/RP0/CPU0:ios(config)#do show ipv6 interface mgmtEth 0/RP0/CPU0/1 brief
   Fri Oct 20 05:26:52.612 UTC
   MgmtEth0/RP0/CPU0/1    [Up/Up]
       fe80::3afd:f8ff:fe66:872
       2001::1
   ```

3. Enter the **ipv6 access-list V6-INGRESS-ACL** command to configure an IPv6 ingress ACL.

   Example

```
/* Configure an IPv6 ingress ACL */
RP/0/RP0/CPU0:ios(config)#ipv6 access-list V6-INGRESS-ACL
RP/0/RP0/CPU0:ios(config-ipv6-acl)#10 permit ipv6 any any
RP/0/RP0/CPU0:ios(config-ipv6-acl)#20 deny udp any any
RP/0/RP0/CPU0:ios(config-ipv6-acl)#commit
Fri Oct 20 05:28:46.664 UTC
RP/0/RP0/CPU0:ios(config-ipv6-acl)#exit
```

4. Enter the **show access-lists ipv6** command to verify the ingress ACL creation.

   Example

```
/* Verify the ingress ACL creation */
RP/0/RP0/CPU0:ios(config)#do show access-lists ipv6
Fri Oct 20 05:29:01.125 UTC
ipv6 access-list V6-INGRESS-ACL
 10 permit ipv6 any any
 20 deny udp any any
```

5. Enter the **interface mgmtEth 0/RP0/CPU0/1** command to apply the ingress ACL to the HundredGigE interface.

   Example

```
/* Apply the ingress ACL to the HundredGigE interface */
RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/1
RP/0/RP0/CPU0:ios(config-if)#ipv6 access-group V6-INGRESS-ACL ingress
RP/0/RP0/CPU0:ios(config-if)#commit
Fri Oct 20 05:37:32.738 UTC
RP/0/RP0/CPU0:ios(config-if)#exit
```

6. Enter the **show ipv6 interface mgmtEth 0/RP0/CPU0/1** command to verify if the ingress ACL has been successfully applied to the interface.

   Example

```
/* Verify if the ingress ACL has been successfully applied to the interface
*/
RP/0/RP0/CPU0:ios(config)#do show ipv6 interface mgmtEth 0/RP0/CPU0/1
Fri Oct 20 05:38:00.753 UTC
MgmtEth0/RP0/CPU0/1 is Up, ipv6 protocol is Up, Vrfid is default (0x60000000)

  IPv6 is enabled, link-local address is fe80::3afd:f8ff:fe66:872
  Global unicast address(es):
    2001::1, subnet is 2001::/64
  Joined group address(es): ff02::1:ff00:1 ff02::1:ff66:872 ff02::2
      ff02::1
  MTU is 1514 (1500 is available to IPv6)
  ICMP redirects are disabled
  ICMP unreachables are enabled
  ND DAD is enabled, number of DAD attempts 1
```

```
ND reachable time is 0 milliseconds
ND cache entry limit is 1000000000
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 160 to 240 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.
Outgoing access list is not set
Inbound  common access list is not set, access list is V6-INGRESS-ACL
Table Id is 0xe0800000
Complete protocol adjacency: 0
Complete glean adjacency: 0
Incomplete protocol adjacency: 0
Incomplete glean adjacency: 0
Dropped protocol request: 0
Dropped glean request: 0
RA DNS Server Address Count: 0
RA DNS Search list Count: 0
```

You have successfully configured an IPv6 ingress ACL on the mgmtEth interface.

Configure an egress IPv6 ACL on the management Ethernet interface

Use this procedure to configure an egress IPv6 ACL on the management Ethernet interface so that outbound IPv6 traffic follows defined rules.

Configure an egress IPv6 ACL on the management Ethernet interface so that Cisco NCS 1010 uses the required settings for the selected system setup, management, or routing workflow.

Use the following configuration steps to configure an egress IPv6 ACL on the mgmtEth interface.

Before you begin

Follow these steps to configure an egress IPv6 ACL on the management Ethernet interface:

Procedure

1. Run the **interface mgmtEth 0/RP0/CPU0/1** command to configure a mgmtEth interface with an IPv6 address.

   Example

   ```
   /* Configure a mgmtEth interface with an IPv6 address */
   RP/0/RP0/CPU0:ios(config)#interface mgmtEth 0/RP0/CPU0/1
   RP/0/RP0/CPU0:ios(config-if)#ipv6 address 2001::1/64
   RP/0/RP0/CPU0:ios(config-if)#no shutdown
   RP/0/RP0/CPU0:ios(config-if)#commit
   Thu Jan 25 11:41:25.778 IST
   Router(config-if)# exit
   ```

2. Run the **show ipv6 interface brief** command to verify if the interface is up.

   Example

   ```
   /* Verify if the interface is up */
   Router(config)# do show ipv6 interface brief
   Thu Jul 11 09:47:50.812 UTC
   ```

```
HundredGigE 0/0/0/0 [Up/Up]
    fe80::bd:b9ff:fea9:5606
    1001::1
HundredGigE 0/0/0/1 [Up/Up]
    fe80::23:e9ff:fea8:a44e
    2001::1
```

3. Run the **ipv6 access-list V6-EGRESS-ACL** command to configure an IPv6 egress ACL.

   Example

```
/* Configure an IPv6 egress ACL */
Router(config)# ipv6 access-list V6-EGRESS-ACL
Router(config-ipv6-acl)# 10 permit ipv6 any any
Router(config-ipv6-acl)# 20 deny udp any any
Router(config-ipv6-acl)# commit
Thu Jul 11 09:50:40.566 UTC
Router(config-ipv6-acl)# exit
```

4. Run the **show ipv6 interface mgmtEth 0/RP0/CPU0/1 brief** command to verify the egress ACL creation.

   Example

```
/* Verify the egress ACL creation */
RP/0/RP0/CPU0:ios(config)#do show ipv6 interface mgmtEth 0/RP0/CPU0/1 brief
Fri Oct 20 05:26:52.612 UTC
MgmtEth0/RP0/CPU0/1    [Up/Up]
    fe80::3afd:f8ff:fe66:872
    2001::1
…
```

5. Run the **interface mgmtEth 0/RP0/CPU0/1** command to apply the egress ACL to the mgmtEth interface.

   Example

```
/* Apply the egress ACL to the mgmtEth interface */
Router(config)# interface mgmtEth 0/RP0/CPU0/1
Router(config-if)# ipv6 access-group V6-EGRESS-ACL egress
Router(config-if)# commit
Thu Jul 11 09:52:57.751 UTC
Router(config-if)# exit
```

6. Run the **show ipv6 interface mgmtEth 0/RP0/CPU0/1** command to verify if the egress ACL has been successfully applied to the interface.

   Example

```
/* Verify if the egress ACL has been successfully applied to the interface
*/
RP/0/RP0/CPU0:ios(config)#do show ipv6 interface mgmtEth 0/RP0/CPU0/1
Fri Oct 20 05:38:00.753 UTC
MgmtEth0/RP0/CPU0/1 is Up, ipv6 protocol is Up, Vrfid is default (0x60000000)
```

```
  IPv6 is enabled, link-local address is fe80::3afd:f8ff:fe66:872
  Global unicast address(es):
    2001::1, subnet is 2001::/64
  Joined group address(es): ff02::1:ff00:1 ff02::1:ff66:872 ff02::2
      ff02::1
  MTU is 1514 (1500 is available to IPv6)
  ICMP redirects are disabled
  ICMP unreachables are enabled
  ND DAD is enabled, number of DAD attempts 1
  ND reachable time is 0 milliseconds
  ND cache entry limit is 1000000000
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 160 to 240 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
  Outgoing access list is V6-EGRESS-ACL
  Inbound  common access list is not set, access list is not set
  Table Id is 0xe0800000
  Complete protocol adjacency: 0
  Complete glean adjacency: 0
  Incomplete protocol adjacency: 0
  Incomplete glean adjacency: 0
  Dropped protocol request: 0
  Dropped glean request: 0
  RA DNS Server Address Count: 0
  RA DNS Search list Count: 0
...
```

You have successfully configured an IPv6 egress ACL on the mgmtEth interface.

Configure extended access lists

Use this procedure to configure extended access lists on Cisco NCS 1010 so that packet filtering can match source, destination, and protocol fields.

Configure extended access lists so that Cisco NCS 1010 uses the required settings for the selected system setup, management, or routing workflow.

To configure Extended Access Lists, you must create an access list and specify the condition to allow or deny the network traffic.

Use Extended Access Lists to verify more than just the source address of the packets. Attributes such as destination address, specific IP protocols, UDP or TCP port numbers, DSCP, and so on are validated. Traffic is controlled by a comparison of the attributes stated in the ACL with those in the incoming or outgoing packets.

Running Configuration

Verification

Before you begin

Follow these steps to configure extended access lists:

Procedure

1. Run the **configure** command to enter the global configuration mode and create the access list.

Example

```
/* Enter the global configuration mode and create the access list*/
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#ipv4 access-list acl_1
RP/0/RP0/CPU0:ios(config-ipv4-acl)#10 remark Do not allow user1 to telnet out
/*Specify the condition to allow or deny the network traffic.*/
RP/0/RP0/CPU0:ios(config-ipv4-acl)#10 permit 172.16.0.0 0.0.255.255
RP/0/RP0/CPU0:ios(config-ipv4-acl)#20 deny 192.168.34.0 0.0.0.255
RP/0/RP0/CPU0:ios(config-ipv4-acl)#commit
```

2. Run the **show running-config** command to verify the configuration.

Example

```
RP/0/RP0/CPU0:ios#show running-config
Fri Oct 20 06:21:11.024 UTC
!! Building configuration...
!! IOS XR Configuration 24.1.1.23I
!! Last configuration change at Fri Oct 20 06:19:08 2023 by cisco

!
ipv4 access-list acl_1
 10 permit ipv4 172.16.0.0 0.0.255.255 any
 20 deny ipv4 192.168.34.0 0.0.0.255 any
!
```

3. Run the **show access-lists ipv4 acl_1** command to verify the configuration.

Example

```
RP/0/RP0/CPU0:ios#show access-lists ipv4 acl_1
Fri Oct 20 06:22:17.223 UTC
ipv4 access-list acl_1
 10 permit ipv4 172.16.0.0 0.0.255.255 any
 20 deny ipv4 192.168.34.0 0.0.0.255 any
```

The system completes the configure extended access lists task.

Modify ACLs

Use this procedure to modify ACLs on Cisco NCS 1010 so that existing packet-filtering rules match current traffic-control requirements accurately.

Configure modify ACLs so that Cisco NCS 1010 uses the required settings for the selected system setup, management, or routing workflow.

This section describes a sample configuration to modify ACLs.

/When an entry is added without a sequence number, it is automatically given a sequence number that puts it at the end of the access list. Because the default increment is 10, the entry will have a sequence number 10 higher than the last entry in the existing access list/

Before you begin

Follow these steps to modify ACLs:

Procedure

1. Run the **configure** command to complete this action.

   Example

```
*/ Create an Access List*/
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#ipv4 access-list acl_1

*/Add entries (ACEs) to the ACL*/
RP/0/RP0/CPU0:ios(config-ipv4-acl)#10 permit ip host 10.3.3.3 host 172.16.5.34
RP/0/RP0/CPU0:ios(config-ipv4-acl)#20 permit icmp any any
RP/0/RP0/CPU0:ios(config-ipv4-acl)#30 permit tcp any host 10.3.3.3
RP/0/RP0/CPU0:ios(config-ipv4-acl)#end
```

2. Run the ***/Verify the entries of the ACL*/:** command to complete this action.

   Example

```
*/Verify the entries of the ACL*/:
Router#show access-lists ipv4 acl_1
ipv4 access-list acl_1
10 permit ip host 10.3.3.3 host 172.16.5.34
20 permit icmp any any
30 permit tcp any host 10.3.3.3
```

3. Run the **ipv4 access-list acl_1** command to complete this action.

   Example

```
*/Add new entries, one with a sequence number "15" and another without a
sequence number to the ACL. Delete an entry with the sequence number "30":*/
RP/0/RP0/CPU0:ios(config)#ipv4 access-list acl_1
RP/0/RP0/CPU0:ios(config-ipv4-acl)# 15 permit 10.5.5.5 0.0.0.255
RP/0/RP0/CPU0:ios(config-ipv4-acl)# no 30
RP/0/RP0/CPU0:ios(config-ipv4-acl)# permit 10.4.4.4 0.0.0.255
RP/0/RP0/CPU0:ios(config-ipv4-acl)# commit
```

4. Run the **show access-lists ipv4 acl_1** command to verify the configuration.

   Example

```
*/Verify the entries of the ACL:*/
RP/0/RP0/CPU0:ios#show access-lists ipv4 acl_1
ipv4 access-list acl_1
 10 permit ipv4 host 10.3.3.3 host 172.16.5.34

15 permit 10.5.5.5 0.0.0.255---*/newly added ACL (with the sequence number)*/
20 permit icmp any any
30 permit ipv4 10.4.4.0 0.0.0.255 any ---*/newly added ACL (without the
sequence number)*/
```

```
*/The entry with the sequence number 30, that is, "30 permit tcp any host
10.3.3.3" is deleted from the ACL*/
```

You have successfully modified ACLs in operation.

9 Smart Licensing

Topics:

- Smart Licensing
- Create a token
- Supported transport modes for Smart Licensing
- License entitlements and smart licensing features for NCS1K4-QXP-L-K9 and NCS1K14-OXP2-L-K9 line card
- Smart licensing features for NCS1014-EDFA2 line card

Provides instructions for configuring and managing Smart Licensing on the Cisco NCS 1004. It explains how to register devices, monitor license consumption, and utilize different transport modes to ensure compliance across various line card configurations and network deployments.

Smart Licensing

Describes Smart Licensing, a cloud-based Cisco licensing approach that simplifies how you purchase, deploy, track, and renew Cisco software, and provides visibility into license ownership and consumption through a single interface.

Smart Licensing is a cloud-based licensing solution that simplifies the licensing experience across the enterprise by making it easier to purchase, deploy, track, and renew Cisco software through a single, simple user interface.

- Provides visibility into license ownership and consumption.
- Allows you to track the status of your license and software usage trends.
- Helps you simplify three core functions: purchasing, management, and reporting.

Core functions of Smart Licensing

Smart Licensing helps you simplify the following core functions.

- **Purchasing**: The software that you have installed in your network can be registered, without Product Activation Keys (PAKs).
- **Management**: You can automatically track activations against your license entitlements. Also, there is no need to install the license file on every node. You can create license pools (logical grouping of licenses) to reflect your organization structure. Smart Licensing offers you Cisco Smart Software Manager, a centralized portal that enables you to manage all your Cisco software licenses from one centralized website.
- **Reporting**: Through the portal, Smart Licensing offers an integrated view of the licenses you have purchased and what has been deployed in your network. You can use this data to make better purchasing decisions, based on your consumption.

Smart Licensing features

Smart Licensing provides the following features.

- Your device initiates a call home and requests the licenses it needs.
- Pooled licences - Licences are company account-specific, and can be used with any compatible device in your company. You can activate or deactivate different types of licenses on the device without actually installing a license file on the device.
- Licenses are stored securely on Cisco servers.
- Licenses can be moved between product instances without license transfer. This greatly simplifies the reassignment of a software license as part of the Return Material Authorization (RMA) process.
- It provides a complete view of all the Smart Software Licenses used in the network using a consolidated usage report of software licenses and devices in one easy-to-use portal.

Cisco Smart Account

NCS 1014 integrates with Cisco Smart Accounts to simplify license management.

Smart Accounts provide a centralized, organized, and simple-to-use solution for managing Cisco software licenses across an entire organization. They act as a container that holds all Cisco software assets, allowing customers to view, store, manage, and move these assets as needed. Smart Accounts enable full visibility into license entitlements and usage, helping optimize software management. Smart Accounts support license pooling, portability, and provide compliance reporting.

When creating a Smart Account, you must have the authority to represent the requesting organization. After you submit the request, it goes through a brief approval process. Access http://software.cisco.com to learn about, set up, or manage Smart Accounts.

Cisco Smart Software Manager

Cisco Smart Software Manager enables you to manage all your Cisco Smart software licenses from one centralized website. With Cisco Smart Software Manager, you organize and view your licenses in groups called virtual accounts (collections of licenses and product instances). Use the Cisco Smart Software Manager to do the following tasks:

- Create, manage, or view virtual accounts.
- Create and manage Product Instance Registration Tokens.
- Transfer licenses between virtual accounts or view licenses.
- Transfer, remove, or view product instances.
- Run reports against your virtual accounts.
- Modify your email notification settings.
- View overall account information.

Virtual Accounts

Virtual Accounts are customizable subaccounts within a Smart Account used to organize and optimize Cisco licenses. They can be structured to reflect business units, product types, or geographic locations, allowing better planning and utilization of assets. They are created and maintained by the Smart Account administrator. Smart Licencing allows you to create multiple license pools or virtual accounts within the Smart Software Manager portal. Using the Virtual Accounts option that you can aggregate licenses into discrete bundles that are associated with a cost center so that one section of an organization cannot use the licenses of another section of the organization. For example, if you segregate your company into different geographic regions, you can create a virtual account for each region to hold the licenses and product instances for that region.

All new licenses and product instances are placed in the default virtual account in the Smart Software Manager, unless you specify a different one during the order process. After you access the default account, you may choose to transfer them to any other account, provided you have the required access permissions.

Use the Smart Software Manager portal to create license pools or transfer licenses.

Product Instance Registration Tokens

A product requires a registration token until you have registered the product. On successful registration, the device receives an identity certificate. This certificate is saved and automatically used for all future communications with Cisco. Registration tokens are stored in the Product Instance Registration Token Table that is associated with your enterprise account. Registration tokens can be valid 1–365 days.

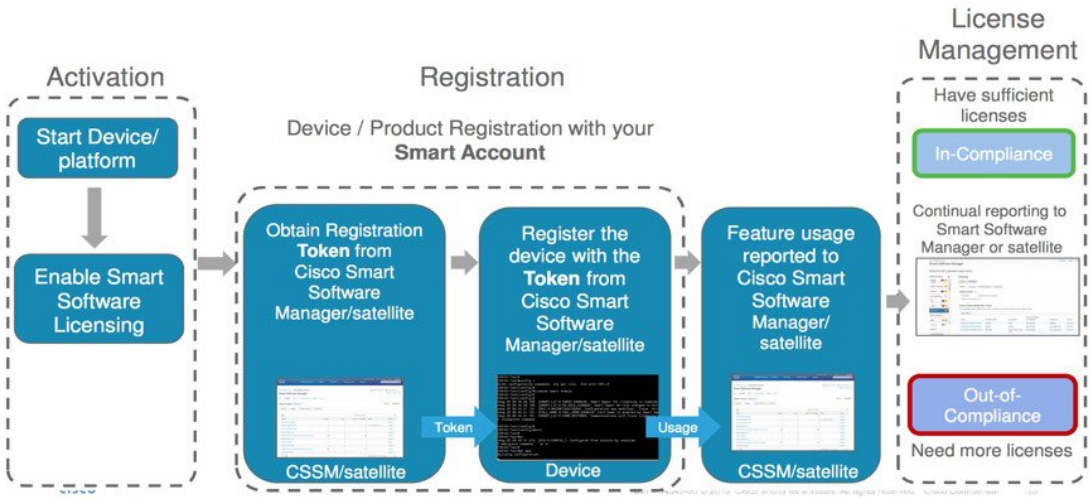
Product Instances

A product instance is an individual device with a unique device identifier (UDI) that is registered using a product instance registration token (or registration token). You can register any number of instances of a product with a single registration token. Each product instance can have one or more licenses residing in the same virtual account. Product instances must periodically connect to the Cisco Smart Software Manager servers during a specific renewal period. If you remove the product instance, its licenses are released and made available within the virtual account.

Smart Licensing workflow

The following figure depicts a working model of smart licensing that involves a three-step procedure.

Figure 5: Smart Licensing Work Flow



1. **Setting up Smart Licensing**: You can place the order for Smart Licensing, to manage licenses on the Cisco.com portal. You agree to the terms and conditions governing the use and access of Smart Licensing in the Smart Software Manager portal.
2. **Enabling and Use Smart Licensing**: Smart Licensing is enabled by default. You can use either of the following options to communicate:
   - **Smart Call Home**: The Smart Call Home feature is automatically configured when Smart Licensing is enabled. Smart Call Home is used by Smart Licensing as a medium for communication with the Cisco license service. Call Home feature allows Cisco products to periodically call-home and perform an audit and reconciliation of your software usage information. This information helps Cisco efficiently track your install base, keep them up and running, and effectively pursue service and support contract renewals. For more information on Smart Call Home feature, see http://www.cisco.com/c/dam/en/us/td/docs/switches/lan/smart_call_home/SCH_Deployment_Guide.pdf.
   - **Smart Software Manager Satellite**: is a component of Cisco Smart Licensing and works with Cisco Smart Software Manager (SSM). It helps customers intelligently manage product licenses, providing near real-time visibility and reporting of the Cisco licenses they purchase and consume.

     For customers who do not want to manage their installed base using a direct Internet connection, the Smart Software Manager satellite is installed on the customer premises and provides a subset of Cisco SSM functionality. After you download the satellite application, deploy it, and register it to Cisco SSM, you can perform the following functions locally:

     - Activate or register a license
     - Get visibility to your company's licenses
     - Transfer licenses between company entities

     Periodically, the satellite must synchronize with Cisco SSM to reflect the latest license entitlements.

     For more information about the Smart Software Manager satellite, see http://www.cisco.com/c/en/us/buy/smart-accounts/software-manager-satellite.html.
3. **Manage and Report Licenses**: You can manage and view reports about your overall software usage in the Smart Software Manager portal. Compliance reporting describes the types of Smart Licensing reports.

Smart licensing benefits for Cisco NCS 1014

Lists the key benefits that Smart Licensing provides for Cisco NCS 1014, including license pooling, compliance notifications, and asset management visibility.

Smart Licensing for Cisco NCS 1014 offers several important advantages. These benefits streamline license management, enhance operational flexibility, and improve visibility of software assets across Cisco NCS 1014 deployments.

- License pooling: Licenses can be pooled and reused across the entire organization, enabling flexibility across organizational boundaries.
- No lock-out on over-consumption: Configurations can be performed even if license usage exceeds the purchased limit. When this occurs, out-of-compliance notifications are issued, allowing continued operations while prompting the user to buy additional licenses as needed.
- Asset management visibility: Smart Licensing provides detailed information for software asset management, which helps organizations plan and track their license usage.

Limitations of Smart Licesing

Lists known issues affecting smart licensing status updates after line card reloads or swaps.

These are the issues that affect smart licensing status updates after line card reloads or swaps.

- After a line card reload or a complete line card swap, the trunk license consumption status does not update immediately.
- The system requires the next scheduled synchronization interval to accurately reflect the current license usage status.
- During this interval, the displayed license usage may temporarily be inaccurate.

Create a token

Describes how to create a new product instance registration token by using Cisco Smart Software Manager so that you can register Cisco NCS 1014 with smart licensing.

To create a new token using Cisco Smart Software Manager, perform the following tasks:

Procedure

1. Log in to the Cisco Smart Software Manager.

   https://software.cisco.com/software/csws/ws/platform/home#SmartLicensing-Inventory

2. Click the **Inventory** tab, and select your virtual account from the **Virtual Account** drop-down list.
3. Click the **General** tab, and click **New Token**.

   The **Create ID Token** window is displayed.

4. Enter the token description. Specify the number of days the token must be active.
5. Check the **Allow export-controlled functionality on the products registered with this token** check box.
6. Click **Create ID Token**.
7. Copy the token and register NCS1014 with the same token ID.

   An example of the token ID: YzY2ZjYyNjktY2NlOS00NTc4LWIxNTAtMjZkNmNiNzMxMTY1LTE2NjAzNjQ3%0ANzY4NjI8ZVJSckxKN2pFV2tleHVoMUkxbGxTazFDVm9kc1B5MGlHQmlFWUJi%0Ac3VNRT0%3D%0A

Supported transport modes for Smart Licensing

Lists the supported transport modes for Smart Licensing, describing each mode's main characteristics and selection options.

Smart Licensing software management enables you to choose from several transport modes, depending on your network connectivity and operational requirements. The default transport mode is Cisco Smart Licensing Utility (CSLU), but you can select Call-Home, Smart Transport, or Offline modes based on your needs.

The supported transport modes and their key features are:

- **Call-Home mode**: Directly communicates with Cisco through a Call-Home feature to register and manage licenses. Suitable for networks with internet access.
- **Smart Transport mode**: Utilizes Smart Licensing Transport to relay license information securely to Cisco servers. Recommended for environments that require secure, automated reporting.
- **CSLU mode**: Uses Cisco Smart Licensing Utility as an intermediary for managing licenses locally, offering flexibility and local control. This mode is the default.
- **Offline mode**: Operates without an active internet connection by generating and transferring files manually for license management. Designed for isolated or secure environments where connectivity is restricted.

You can change the transport mode to best match your network and management preferences.

Configure Call Home for Smart Licensing

Configure Call Home settings to enable your device to connect with the Cisco Smart Software Manager (CSSM) for automated license management.

Enable Call Home to establish secure communication with CSSM, which is required for smart licensing and license reporting on your device.

Call Home connects your device with CSSM and automates license registration, reporting, and updates. Proper configuration ensures compliance and full access to Cisco licensing features.

Procedure

1. Enable Call Home and configure Call Home mode settings:

   Example

```
RP/0/RP0/CPU0:ios#call-home
RP/0/RP0/CPU0:ios(config-call-home)#service active
RP/0/RP0/CPU0:ios(config-call-home)#contact smart-licensing
RP/0/RP0/CPU0:ios(config-call-home)#profile CiscoTAC-1
RP/0/RP0/CPU0:ios(config-call-home-profile)#active
RP/0/RP0/CPU0:ios(config-call-home-profile)#destination address http
https://tools.cisco.com/its/service/oddce/services/DDCEService
RP/0/RP0/CPU0:ios(config-call-home-profile)#reporting smart-call-home-data
RP/0/RP0/CPU0:ios(config-call-home-profile)#reporting smart-licensing-data
RP/0/RP0/CPU0:ios(config-call-home-profile)#destination transport-method email
 disable
RP/0/RP0/CPU0:ios(config-call-home-profile)#destination transport-method http
RP/0/RP0/CPU0:ios(config-call-home-profile)#commit
RP/0/RP0/CPU0:ios(config-call-home-profile)#end
```

2. Configure the domain name and DNS name server.

Example

```
RP/0/RP0/CPU0:ios#config
RP/0/RP0/CPU0:ios(config)#domain name cisco.com
RP/0/RP0/CPU0:ios(config)#domain name-server 64.102.6.247
RP/0/RP0/CPU0:ios(config)#commit
RP/0/RP0/CPU0:ios(config)#end
```

3. Enable Certificate Revocation List (CRL) configuration, if you have a Trustpool CA.

Example

```
RP/0/RP0/CPU0:ios#config
RP/0/RP0/CPU0:ios(config)#crypto ca trustpoint Trustpool crl optional
RP/0/RP0/CPU0:ios(config)#commit
RP/0/RP0/CPU0:ios(config)#end
```

4. Set Call Home as the smart license transport mode.

Example

```
RP/0/RP0/CPU0:ios#config
RP/0/RP0/CPU0:ios(config)#License smart transport callhome
RP/0/RP0/CPU0:ios(config)#commit
RP/0/RP0/CPU0:ios(config)#end
```

Verify whether the Callhome is Configured.

```
RP/0/RP0/CPU0:ios#show license all
Transport:  Type: Callhome
```

5. Establish trust with smart licensing using an ID token.

Example

```
license smart trust idtoken Zesdf3243u48329fdfhsfhsfkjs1233j4h1j1j4j41n
```

Call Home is enabled and configured on your device. The device will connect to CSSM for automated license management, compliance, and reporting.

Configure Smart Transport

Configure Smart Transport to enable secure license communication and registration with Cisco Smart Software Manager (CSSM) without using Call Home.

Set up Smart Transport to connect your device to the CSSM for license management and establish a trusted network connection.

Smart Transport serves as an alternative to Call Home for connecting to CSSM. It enables secure network-based license registration and trust establishment.

Procedure

1. Configure the Smart proxy and hostname for communication with CSSM.

   Example

   ```
   RP/0/RP0/CPU0:ios#config
   RP/0/RP0/CPU0:ios(config)#license smart proxy port 80
   RP/0/RP0/CPU0:ios(config)#license smart proxy hostname proxy.esl.cisco.com
   RP/0/RP0/CPU0:ios(config)#commit
   RP/0/RP0/CPU0:ios(config)#end
   ```

2. Enable Certificate Revocation List (CRL) configuration to manage certificate trust.

   Example

   ```
   RP/0/RP0/CPU0:ios#config
   RP/0/RP0/CPU0:ios(config)#crypto ca trustpoint Trustpool crl optional
   RP/0/RP0/CPU0:ios(config)#commit
   RP/0/RP0/CPU0:ios(config)#end
   ```

3. Enable Smart Transport for license registration.

   Example

   ```
   RP/0/RP0/CPU0:ios#config
   RP/0/RP0/CPU0:ios(config)#License smart transport smart
   RP/0/RP0/CPU0:ios(config)#commit
   RP/0/RP0/CPU0:ios(config)#end
   ```

   Verify Smart Transport is configured correctly.

   ```
   RP/0/RP0/CPU0:ios#show license all
   Transport:
     Type: Smart
     URL: https://smartreceiver.cisco.com/licservice/license
     Proxy:
       Address: proxy.esl.cisco.com
       Port: 80
       Username: <empty>
       Password: <empty>
     VRF:
       Not Supported
   ```

4. Establish trust with CSSM using an id-token.

   Example

   ```
   license smart trust idtoken Zesdf3243u48329fdfhsfhsfkjs1233j4h1j1j4j41n
   ```

Smart Transport is configured for secure communication and trust establishment with CSSM. The device can now use CSSM for license registration and management tasks.

Configure CSLU transport mode

Configure CSLU transport mode to enable device software licensing using Cisco Smart License Utility.

Establish CSLU as the transport mode for license communication between your device and an on-premise CSLU server.

CSLU (Cisco Smart License Utility) is the default transport mode for software license management. Configuring CSLU enables secure license communications and compliance reporting for your device.

Procedure

1. Use the **license smart url cslu http://<cslu-local>:8182/cslu/v1/pi** command to configure the CSLU URL.

   Example
   In this sample configuration, the **10.127.59.44** is the CSLU URL that the on-premise server provided. This URL changes for each on-premise servers.

   ```
   RP/0/RP0/CPU0:ios#config
   RP/0/RP0/CPU0:ios(config)#license smart url cslu
   http://10.127.59.44:8182/cslu/v1/pi
   RP/0/RP0/CPU0:ios(config)#commit
   RP/0/RP0/CPU0:ios(config)#end
   ```

2. Use this sample configuration to enable CRL Configuration.

   Example

   ```
   RP/0/RP0/CPU0:ios#config
   RP/0/RP0/CPU0:ios(config)#crypto ca trustpoint Trustpool crl optional
   RP/0/RP0/CPU0:ios(config)#commit
   RP/0/RP0/CPU0:ios(config)#end
   ```

3. Use the **license smart transport cslu** command to enable CSLU.

   Example

   ```
   RP/0/RP0/CPU0:ios#config
   RP/0/RP0/CPU0:ios(config)#license smart transport cslu
   RP/0/RP0/CPU0:ios(config)#commit
   RP/0/RP0/CPU0:ios(config)#end
   ```

4. Use the **show license all** command to verify the CSLU configuration.

   Example
   The sample output shows the configuration transport type with the on-prem server address.

   ```
   RP/0/RP0/CPU0:ios#show license all
   Transport:
     Type: cslu
     Cslu address: http://10.127.59.44:8182/cslu/v1/pi
     Proxy:
       Not Configured
     VRF:
       Not Supported
   ```

5. (Optional) Use the **license smart trust idtoken** *<token-id>* command to establish trust using id-token.

   Example
   The command uses **Zesdf3243u48329fdfhsfhsfkjs1233j4h1j1j4j41n** as a sample token id.

```
RP/0/RP0/CPU0:ios#license smart trust idtoken
Zesdf3243u48329fdfhsfhsfkjs1233j4h1j1j4j41n
```

6. Use the **license smart sync all** command to send the license usage reports to server and receive the compliance status for the licenses consumed by the node.

   Example

```
RP/0/RP0/CPU0:ios#license smart sync all
Tue Aug 26 13:03:28.287 IST
```

> **Note**
>
> If you are using on-prem version 9-202407 or later, this command retrieves the on-prem account details and the compliance information.

The device uses CSLU transport for license communications, and all required trust relationships are established.

What to do next

- Review device status in Cisco Smart Software Manager.
- Ensure device reports successful license communication

Configure offline license mode

Configure offline license mode by disabling transport, saving the license usage report, and importing the acknowledgment report.

Enable offline license operation for devices that cannot directly connect to Cisco Smart Licensing servers.

Offline license mode is necessary when your device cannot communicate with Cisco Smart Licensing servers over the network. This process allows you to manage licenses through manual file operations.

Procedure

1. Disable transport to switch the device to offline license mode

   Example

```
RP/0/RP0/CPU0:ios#config
RP/0/RP0/CPU0:ios(config)#License smart transport off
RP/0/RP0/CPU0:ios(config)#commit
RP/0/RP0/CPU0:ios(config)#end
```

2. Save the license usage report.

   Example

   ```
   RP/0/RP0/CPU0:ios#license smart save usage unreported /misc1/disk1/usage.txt
   ```

3. Import the acknowledgment report.

   Example

   ```
   RP/0/RP0/CPU0:ios#license smart import /misc/disk1/ACK_usage.txt
   ```

The device operates in offline license mode. License usage is recorded and acknowledgments are managed manually without network transport to Cisco Smart Licensing.

License entitlements and smart licensing features for NCS1K4-QXP-L-K9 and NCS1K14-OXP2-L-K9 line card

Provides detailed information about smart licensing support, feature history, and license entitlements for the NCS1K4-QXP-L-K9 and NCS1K14-OXP2-L-K9 line card.

Table 23: Feature History

Feature Name	Release Information	Feature Description
Smart Licensing for OXP2-L-K9 Line Card	Cisco IOS XR Release 26.3.1	The NCS1K14-OXP2-L-K9 supports the smart licensing feature. It enables you to automate the time-consuming manual licensing tasks and allows you to easily track the status of your license and software usage trends.
Smart Licensing for QXP Line Card	Cisco IOS XR Release 24.3.1	Now the NCS1K4-QXP-L-K9 supports the smart licensing feature. It enables you to automate the time-consuming manual licensing tasks and allows you to easily track the status of your license and software usage trends.

Table 24: License Entitlements for NCS1K4-QXP-L-K9 and NCS1K14-OXP2-L-K9

Display Name in CSSM Server	Description	Consumption criteria
S_N1K4_LIC_TRK	NCS 1K4 Smart License QDDTXP Trunk	QXP-L-K9 card supports up to eight licenses; OXP2-L-K9 card supports up to 16 licenses. **Note** The number of licenses equals the number of trunk ports as shown in the output of the **hw-module muxponder slice** command on the licensed card.
NCS1014_ESS_TXP_RTU	Essential Tier Transponder RTU	One license per line card
NCS1014_ESS_TXP_SIA	Essentials Subscription Transponder SIA	One license per line card

Smart licensing features for NCS1014-EDFA2 line card

Lists the supported Smart Licensing features, license types, and license consumption patterns available for the NCS1014-EDFA2 line card.

The NCS1014-EDFA2 line card supports Cisco Smart Licensing, making license management and usage monitoring simple and efficient. Smart Licensing Using Policy (SLP) streamlines the licensing process for Cisco IOS XR products by eliminating device registration during installation and removing evaluation license states or periods. Support for Smart Licensing Using Policy is now available for the NCS1014-EDFA2 line card.

Table 25: Feature History

Feature Name	Release Information	Feature Description
Smart Licensing using Policy on EDFA2 Line Card	Cisco IOS XR Release 25.1.1	Cisco Smart Licensing Using Policy (SLP) streamlines the licensing process for Cisco IOS XR products. You no longer need to register your device during installation, and there is no evaluation license state or period. Support for Smart Licensing using Policy is now extended to NCS1014-EDFA2 line card

The NCS1K14-EDFA2 line card supports Smart Licensing, allowing you to manage licenses and monitor usage trends with ease.

This table lists the licenses available for the EDFA2 line card.

Table 26: Licenses and usage consumption pattern

License Name	Description	Consumption Pattern
ESS-EDFA-RTU	Essential Tier EDFA RTU	One license per EDFA card
ESS-EDFA-SIA	Essentials Subscription EDFA SIA	One license per EDFA card for 3 years

10 Automated file management

Topics:

- Automated file management system

Introduces the automated file management system for NCS 1014, including supported system-generated file types, log file archiving and retention, and cleanup of showtech-related residual files to conserve SSD space.

The automated file management feature manages system-generated files on each node to conserve local storage.

Table 27: Feature History

Feature Name	Release Information	Feature Description
Automated File Management System	Cisco IOS XR Release 24.4.1	The new Automated File Management System is designed for efficient file handling on each node. This system automatically archives older files and removes them from local nodes to free up valuable SSD space. It manages the following types of files: • System-generated log files • Showtech-related residual files

Automated file management system

Explains how the automated file management system frees solid-state drive space on Cisco NCS 1014 by archiving older system-generated files and removing them from local nodes.

The automated file management system is a file-handling system that

- archives older files on each node,
- removes older files from local nodes to free valuable solid-state drive (SSD) space, and
- applies to system-generated files, including log files, showtech-related residual files, and optical time-domain reflectometer (OTDR) files.

File types for automated file management

Describes the user-generated and system-generated file types stored on the solid-state drive, and identifies which files automated file management handles.

The solid-state drive (SSD) stores two types of generated files:

- User: Creates and owns files for requirement purposes and deletes the files when they are no longer needed.
- System: Organizes files automatically based on the file content and the application that created the content, such as `.log` and showtech-related residual files.

  Automated file management is applicable for the system-generated files.

How automated file management works

Describes how automated file management handles log files, showtech-related residual files, and optical time-domain reflectometer files to conserve storage space on Cisco NCS 1014.

Automated file management handles system-generated files on Cisco NCS 1014, including log files, showtech-related residual files, and optical time-domain reflectometer (OTDR) files.

Summary

The process applies to these file categories:

- Log files: The NCS 1014 system uses log rotation configurations to manage log files as required.
- Showtech-related residual log files: The system removes residual files after log collection.

Workflow

These stages describe how automated file management works for various files.

1. For log files, the system checks for `.log` files that exceed 10 MB.

   > **Note**
   >
   > This threshold is applicable for `/tmp` folder files only. For files in other folders, the system uses a different threshold and follows the same process.

2. After locating the file, the system archives that file with the `.gz` extension and creates a new `.log` file with the same name.

3. The system then maintains one active log file and two archived files.
4. When new files are archived, the system deletes the oldest archived file to make sure that only the two most recently archived files and the current log file are retained.
5. For showtech-related residual log files, the system generates the residual files during the collection of logs when **Showtech logs** is in operation.
6. After collecting the logs, the system automatically removes these residual files to conserve space.

Result

The system retains the current log file, the two most recently archived log files, and the allowed OTDR file count while removing excess or residual files.

11 Implementing audit monitoring

Topics:

- Audit logging
- Configure audit logging
- AuditD process keepalive

Describes audit logging and monitoring support on Cisco NCS 1014 and identifies the release, commands, and feature behavior that administrators use to enable security event monitoring.

This chapter explains the audit monitoring and logging capabilities available on NCS 1014 and how to configure audit monitoring.

Feature name	Release information	Description
Audit logging and monitoring	Cisco IOS XR 25.3.1	You can enable audit logging and monitoring on the NCS 1014. You can also configure predefined rule groups that allow NCS 1014 to monitor activities, log events, and, when necessary, forward audit logs to a remote syslog server for centralized analysis and incident response. This feature helps enhance security and compliance on your network. CLI: These new commands are introduced: • **linux security audit monitor** *group-keyword* • **show linux security audit monitor status** • **linux security audit logging syslog** • **logging** *remote-server-ipv4/remote-server-ip* • **show linux security audit logging syslog**

Audit logging

Explains audit logging as a security and compliance feature that monitors selected system activity, creates audit records, and can forward logs to a remote syslog server for centralized review.

Audit logging is a security and compliance feature that

- operates according to defined audit rules, automatically creating audit logs whenever specified actions or changes occur on the router,
- integrates with the Linux Audit Daemon to monitor and log relevant security events across the router, and
- allows forwarding of audit logs to a remote syslog server.

The Linux audit daemon is a user-space component of the Linux auditing system that

- tracks and logs system calls, file accesses, user actions, and other events as specified by audit rules, and
- provides administrators with insights to detect suspicious behavior and maintain system integrity.

Audit rules and audit logs

An audit rule is a configuration that

- specifies which files, directories, or system events should be monitored,
- determines the conditions for monitoring, and
- forms the foundation of an audit logging system.

An audit log is a chronological record that

- is automatically generated when a monitored event, as defined by an audit rule, occurs, and
- typically includes details such as the event type, timestamp, user or process involved, and affected resources.

Audit rules and audit logs for security monitoring

Administrators define audit rules to track changes to sensitive files, monitor system calls, and observe other critical activities. By customizing audit rules, organizations can align monitoring with their unique security and compliance requirements.

Audit rules establish what to watch, while audit logs capture and document every relevant occurrence, ensuring a complete and actionable history of system activity.

For example, an audit rule that monitors changes to the `/etc/passwd` file creates an audit log entry each time this file is modified.

Audit logging is not to be confused with system logging. While audit logging records security-relevant events, such as user actions and changes to sensitive files, system logging (syslog) captures general system events like service status updates, routine errors, or informational messages.

How audit logging works

Describes how audit logging applies configured audit rules across router nodes, records matching events locally, and forwards audit logs to a remote syslog server when forwarding is enabled.

Audit logging relies on configured rule groups and optional syslog forwarding to monitor security-relevant activity on each node.

Summary

These are the key components involved in this feature:

- Network administrator: The user who initiates configurations from the CLI.
- Linux audit daemon: The process that monitors system activity according to the installed rules and writes audit event logs.
- Local rsyslog daemon: The process that forwards logs to a remote syslog server.
- Remote syslog server: The external server that maintains the logs generated by the router.

The Linux audit daemon is the core service that performs event monitoring and logging, based on the audit rules configured by the administrator. It operates at the operating system level on each node, such as line cards and processors.

Workflow

These stages describe how audit monitoring and logging works.

1. The network administrator enables audit monitoring from the CLI.
2. The router software receives the configuration, applies the relevant audit rules, and distributes these rules to all appropriate nodes.
3. On each node, the Linux audit daemon monitors system events as defined by the audit rules and writes the logs to the local log file at `/var/log/audit/audit.log`.
4. If the network administrator has enabled log forwarding, the audit logs are sent to the local rsyslog daemon, which then forwards the logs to a remote syslog server.

Result

Audit monitoring records security-relevant activity locally and can send the records to a remote syslog server for centralized analysis.

Guidelines for audit logging

Provides guidelines for choosing audit rule groups, reviewing audit status, protecting audit data, and using remote syslog forwarding without creating unnecessary resource or security risks.

Granularity of audit rules

- Enable or disable audit rules only at the group level, not individually within a group.
- Regularly review the status of audit rules and audit log forwarding to ensure that monitoring remains effective.

Resource usage on NCS 1014

Use caution when enabling all rule groups, especially rule groups that monitor frequent events, because this configuration may increase CPU, memory, or disk usage. Enable only the groups required for compliance or security needs.

Security of audit logs and syslog servers

- Allow only users with appropriate administrative privileges to configure or view Linux security audit settings.
- Protect access to audit logs and syslog servers to prevent unauthorized access or tampering.

Log forwarding to remote syslog servers

- Confirm that the remote syslog server is reachable and properly configured before enabling log forwarding.
- Use only trusted network segments for remote syslog servers because NCS 1014 forwards audit logs to remote syslog servers in unencrypted plain text.

These guidelines apply when you enable audit monitoring or audit log forwarding on Cisco NCS 1014.

Notes about audit log storage

Provides notes about local audit log storage, including the default file location, rotation count, and maximum audit log file size when forwarding is not enabled.

Local audit log storage

- NCS 1014 stores audit logs locally at `/var/log/audit/audit.log`, unless you enable log forwarding.
- By default, the system rotates up to five audit log files, each up to 8 MB in size.

These notes apply to local audit log storage when remote forwarding is not enabled.

Configure audit logging

Configures audit logging on Cisco NCS 1014 by enabling audit rule groups, verifying their status, and optionally forwarding audit logs to a remote syslog server.

Configure and monitor audit logs for specific system events by enabling the relevant audit rule groups.

Enable audit rule groups to monitor specific files, directories, or system events. Optional syslog forwarding sends audit logs to a configured remote syslog server for centralized analysis.

Procedure

1. Enter the **linux security audit monitor** *<group-keyword>* command to enable a group of audit rules.

   Example

   ```
   RP/0/RP0/CPU0:ios# configure
   RP/0/RP0/CPU0:ios(config)# linux security audit monitor xr-software
   RP/0/RP0/CPU0:ios(config)# linux security audit monitor user-group-config-files
   RP/0/RP0/CPU0:ios(config)# commit
   ```

2. Enter the **show linux security audit monitor status** command to verify the general status of all active audit rule groups.

   Example

   ```
   RP/0/RP0/CPU0:ios# show linux security audit monitor status
   Wed Aug 20 16:16:23.518 IST
   key name: xr-software                  status: enabled
   rules:
   -a always,exit -F arch=b64 -F dir=/pkg/bin -F perm=wa -k xr_bin_changes
   -a always,exit -F arch=b64 -F dir=/pkg/sbin -F perm=wa -k xr_sbin_changes
   -a always,exit -F arch=b64 -F dir=/pkg/lib -F perm=wa -k xr_lib_changes
   ------------------------------------------------------------
   key name: user-group-config-files      status: enabled
   rules:
   -a always,exit -F arch=b64 -F path=/etc/passwd -F perm=wa -k passwd_changes
   -a always,exit -F arch=b64 -F path=/etc/shadow -F perm=wa -k shadow_changes
   -a always,exit -F arch=b64 -F path=/etc/group -F perm=wa -k group_changes
   -a always,exit -F arch=b64 -F path=/etc/sudoers -F perm=wa -k sudoers_changes
   ------------------------------------------------------------
   ```

3. Optional: Enter the **linux security audit logging syslog** command to enable forwarding of audit logs.

Example

```
RP/0/RP0/CPU0:ios# configure
RP/0/RP0/CPU0:ios(config)# linux security audit logging syslog
RP/0/RP0/CPU0:ios(config)# commit
```

4. Optional: Enter the **logging** *remote-server-ip* **vrf** *vrf-name* command to configure the remote syslog server.

Example

```
RP/0/RP0/CPU0:ios# configure
RP/0/RP0/CPU0:ios(config)# logging 10.0.1.2 vrf default severity info port
default facility local6
RP/0/RP0/CPU0:ios(config)# commit
```

5. Optional: Enter the **show linux security audit logging syslog** command to verify whether audit log forwarding is enabled and to view the configured remote syslog server.

Example

```
RP/0/RP0/CPU0:ios# show linux security audit logging syslog
Wed Aug 20 16:16:44.553 IST
status: enabled
syslog-server(s):
ipaddr: 10.0.1.2 vrf: vrf-default port: 514
ipaddr: 10.0.1.9 vrf: vrf-default port: 514
```

Audit logging monitors the enabled rule groups and forwards audit logs when syslog forwarding and remote server configuration are enabled.

AuditD process keepalive

Explains how AuditD process keepalive events verify the liveness of the AuditD service and audit pipeline on NCS 1014, improving compliance, enabling early detection of silent failures, and supporting large-scale monitoring across distributed fleets.

AuditD process keepalive is a NCS 1014 capability that

- verifies the health of the Linux AuditD service by periodically generating keepalive events
- confirms that the audit logging path remains operational, and
- provides a positive audit signal for compliance and early detection of silent failures.

Table 28: Feature History Table

Feature name	Release information	Feature description
AuditD process keepalive	Cisco IOS XR 26.3.1	AuditD keepalive periodically generates an identifiable Linux audit event at a configurable interval between 10 and 43,200 minutes. Monitoring systems can use these events to verify AuditD health and detect interruptions in audit-event generation or delivery. Additionally, the feature provides automated syslog notifications for service state changes across all nodes. This proactive monitoring eliminates silent failures, ensuring continuous compliance and enhanced operational visibility for large-scale deployments. CLI keywords introduced: The keywords **keepalive** and **interval** are added to the **linux security audit** command.

AuditD process keepalive

Starting with Release 26.3.1, NCS 1014 supports an AuditD keepalive capability. When you enable keepalive, the system periodically generates an identifiable Linux audit event at a user-configurable interval using the **linux security audit keepalive interval** *interval* command. This keepalive event verifies that AuditD process is running and the audit log forwarding remains operational, improving detection of silent failures and strengthening auditability across distributed system fleets.

AuditD process keepalive produces periodic, identifiable Linux audit events that act as heartbeat signals for monitoring systems, helping detect failures in audit-event generation or delivery.

AuditD process keepalive works independently of remote syslog forwarding. If remote syslog forwarding is enabled, NCS 1014 forwards keepalive events to the configured remote syslog server. If remote syslog forwarding is not enabled, NCS 1014 stores the keepalive events in the local audit log.

Each keepalive event is written to the AuditD log as an identifiable record. You can use this record to confirm that:

- AuditD is running
- the audit event path is working
- audit events continue to reach the remote syslog server, if forwarding is enabled

Benefits of AuditD process keepalive

These are key benefits of the AuditD process keepalive:

- Confirms AuditD liveness automatically without manual device checks.
- Helps detect silent AuditD or log-pipeline failures earlier.
- Supports large-scale monitoring across distributed system fleets.
- Improves compliance verification by providing a positive audit signal.
- Allows operators to configure hearbeat log frequency.

Guidelines for AuditD process keepalive

Outlines guidelines for configuring and monitoring the AuditD process keepalive feature on NCS 1014.

Follow these guidelines when deploying AuditD process keepalive:

- Configure the keepalive interval in minutes. The supported range is 10 to 43200 minutes.
- Filter on the `audit_keepalive` key in your monitoring tools so that you can identify keepalive events separately from operational audit events.
- Use the `location` field in the audit log, when present, to identify the source node for a keepalive event. This field helps you distinguish events from different route processor, line card, sysadmin, or host locations.
- Enable AuditD process keepalive explicitly. The feature is not active until you enable it.

Restrictions for AuditD process keepalive

Outlines key restrictions for deploying AuditD process keepalive on NCS 1014, including feature scope, server configuration, and device compatibility.

Always remember these restrictions when deploying AuditD process keepalive:

- AuditD process keepalive verifies AuditD liveness. It does not replace general device health monitoring.
- Keepalive event delivery to a remote server depends on remote syslog forwarding being configured and operational.
- If remote forwarding is not configured, the feature still generates keepalive events, but NCS 1014 stores them locally.
- Verify platform and release availability before deployment.

Configure the AuditD process keepalive

Use this procedure to configures AuditD process keepalive on NCS 1014 and verify that keepalive audit events are generated as expected.

Ensure NCS 1014 AuditD generates keepalive audit events at a configured interval and optionally forwards them to a remote syslog server for monitoring.

Use this task to maintain the integrity of audit logging by confirming periodic keepalive events. You can forward these events to a remote syslog server or store them locally for review.

Before you begin

- Decide how often you want NCS 1014 to generate keepalive events.
- If you want to forward keepalive events to a remote syslog server, configure the remote syslog destination on the system.

Follow these steps to configure AuditD process keepalive:

Procedure

1. Enable AuditD process keepalive and set the interval.

   Example

   ```
   RP/0/RP0/CPU0:ios# linux security audit keepalive interval 1010
   ```

   To disable AuditD process keepalive, use the **no linux security audit keepalive interval 1010** command.

2. Enable audit log forwarding.

   Example

```
RP/0/RP0/CPU0:ios# linux security audit logging syslog
```

3. Configure the remote syslog destination.

   Example

```
RP/0/RP0/CPU0:ios#  logging 10.0.1.2 vrf default port 514
RP/0/RP0/CPU0:ios(config)# commit
```

4. If you enabled remote forwarding, verify the forwarding status.

   Example

```
RP/0/RP0/CPU0:ios#  show linux security audit logging syslog
Mon Aug 10 10:18:01.703 IST
status: disabled
syslog-server(s):
ipaddr: 10.0.1.2 vrf: vrf-default port: 514
```

5. Confirm that keepalive events are generated:
   - Check on the remote syslog server if remote forwarding is enabled and the following syslogs are generated.

```
#Audit configuration syslog
<182>1 2026-03-29T12:41:19.653359+05:30 NCS1020_B0_DT_01
SECURITY-AUDIT-6-EVENT - - - type=CONFIG_CHANGE
msg=audit(1774768279.648:2200): auid=4294967295 ses=4294967295
subj=system_u:system_r:auditctl_t:s0 op=add_rule key="audit_keepalive"
list=4 res=1#035AUID="unset”

#AuditD keepalive interval syslog
<182>1 2026-03-29T12:41:29.105188+05:30 NCS1020_B0_DT_01
SECURITY-AUDIT-6-EVENT - - - type=SYSCALL msg=audit(1774768289.101:2235):
 arch=c000003e syscall=87 success=yes exit=0 a0=7fe4a994a60c a1=7fe4a994a47b
 a2=1 a3=6 items=2 ppid=8348 pid=13338 auid=4294967295 uid=0 gid=1000
euid=0 suid=0 fsuid=0 egid=1000 sgid=1000 fsgid=1000 tty=(none)
ses=4294967295 comm="xr_auditd"
exe="/opt/cisco/install-iosxr/base/bin/xr_auditd"
subj=system_u:system_r:iosxr_t:s0 key="audit_keepalive"#035ARCH=x86_64
SYSCALL=unlink AUID="unset" UID="root" GID="iosxr" EUID="root" SUID="root"
 FSUID="root" EGID="iosxr" SGID="iosxr" FSGID="iosxr”
```

> **Note**
>
> By default, the syslogs are generated in `/var/log/syslog/`. Otherwise, the syslogs are generated in the path configured on the remote syslog server.

- Check in the local audit log at `/var/log/audit/audit.log` if remote forwarding is not enabled.

NCS 1014 generates periodic keepalive audit events at the configured interval.

What to do next

Configure your monitoring system to alert when expected `audit_keepalive` events stop arriving for a device or location.

12 Process Memory Management

Topics:

- Core dump folders

Explains how to manage core dump storage in Cisco IOS XR by configuring disk usage limits and automated cleanup processes. It enables administrators to maintain system reliability and prevent disk space exhaustion by controlling the retention of diagnostic files during system crashes.

This chapter describes memory management processes.

Process memory management refers to the efficient allocation and isolation of memory for individual processes within the modular operating system. Each process runs in its own protected memory space to ensure stability and prevent interference.

Dynamic memory allocation leverages segments like code, data, stack, heap, and shared memory for process execution.

Table 29: Feature History Table

Feature Name	Release	Description
Core dump folder limit	Cisco IOS-XR Release 26.1.1	You can now set disk storage limits to 20% to prevent excessive core dumps, and a new folder limit retains up to 20 coreinfo files. The system automatically cleans up every 15 minutes, ensuring efficient use of disk space and preserving resources for critical operations. This process provides better disk management and enhanced system reliability with configurable storage limits and automated cleanup. **CLI**: • The command **exception disk-usage-limit** is introduced. **YANG data model:** • Cisco-IOS-XR-sysadmin-dumper.yang

Core dump folders

Describes core dump folder storage, attributes, system limits, and troubleshooting features in Cisco IOS XR.

A core dump folder is a directory in the Cisco IOS XR file system that

- stores diagnostic files, known as **core dumps**, which capture the state of the system's memory during a crash or failure,
- is typically located at `/misc/disk1/coredumps`, and
- provides critical information for troubleshooting and debugging system issues.

To manage storage and maintain system reliability, Cisco IOS XR introduces several mechanisms related to core dump folder limits:

- Configurable storage limits: Administrators can set disk storage limits to 20% of total disk space for core dumps, helping prevent the folder from consuming excessive space during repeated crash scenarios.
- Maximum retained files: The storage capacity of /misc/disk1 is 140GB, and 20% of the disk is configured for core dumps, which comes to roughly 28GB. This indicates that the disk usage limit (20%) is being enforced. When the limit is exceeded, older files, typically those ending with `.core.txt` are automatically deleted to free space, every 15 minutes.
- Syslog alerts: The system generates syslog messages to notify administrators when storage limits are reached or when automatic cleanup actions occur. This optimizes disk usage and helps prevent data loss during continuous crash scenarios.

Before Release 26.1.1, the core dump folder lacked a storage limit for core dump files. This could result in the allocated hard disk space being filled during continuous crash scenarios, potentially preventing Cisco IOS XR from saving the latest core files and leading to data loss.

From Release 26.1.1 onward, these improvements allow you to effectively control core dump folder storage and preserve disk space for critical operations, by configuring

- **Throttling mechanism bypass**: Use the **exception filepath** to redirect core files to another directory, utilizing new storage space.
- **Disk usage limit**: Use the **exception disk-usage-limit <usage limit percent>** command to set the storage limit between 20% and 50% of the total hard disk space.

Example

If the number of coreinfo files exceeds 50, Cisco IOS XR deletes older files during its periodic cleanup. These syslog messages indicate this action:

```
RP/0/RP0/CPU0:Jan 20 20:15:02.014 UTC: dumper[69243]: %OS-SYSLOG-6-LOG_INFO :
 Coreinfo count is currently 52, deleting
/misc/disk1/coredumps/sleep_11899.by.3.20250120-191744.xr-vm_node0_RP0_CPU0.34684.core.txt
RP/0/RP0/CPU0:Jan 20 20:15:02.056 UTC: dumper[69255]: %OS-SYSLOG-6-LOG_INFO :
 Coreinfo count is currently 51, deleting
/misc/disk1/coredumps/sleep_13995.by.3.20250120-193249.xr-vm_node0_RP0_CPU0.34684.core.txt
```

Requirement: Maintain core dump folder disk space and usage thresholds

Outlines the requirements for managing core dump folder disk space and usage to ensure system reliability and preserve critical diagnostic data.

To ensure reliable system operation and preserve critical diagnostic data, follow these requirements for managing core dump folder storage:

- Maintain at least 10% free disk space before saving host core dumps. If disk space drops below this threshold, the system stops saving core files and deletes any temporary files.
- Prevent the core dump folder from exceeding the configured disk usage limit. f the folder is over this limit and a recent core file exists within 10 minutes of a new crash, the system stops saving the new core dump and deletes any incoming files.
- Regularly monitor disk space and core dump usage to avoid unexpected data loss and ensure proper incident investigation.

Failure to observe these requirements may result in the loss of core dumps necessary for troubleshooting and analysis.

How core dump folder limit works

Describes how Cisco IOS XR manages core dump storage, monitors disk usage, and takes actions to prevent disk space exhaustion.

Summary

Cisco IOS XR Software controls the core dump folder storage by monitoring disk usage and enforcing configured limits to ensure the system does not run out of disk space.

The key components that are involved in core dump folder limit are:

- **Cisco IOS XR Software**: Monitors disk usage and performs actions.
- **Core dump folder**: Stores the core files created during system crashes.

Workflow

After you enable the disk usage limit with the **exception disk-usage-limit <usage limit percent>** command, IOS XR software controls the core dump folder storage under various disk storage scenarios:

1. Cisco IOS XR software continuously monitors the disk usage of the core dump folder against the configured limit.
2. If the core dump folder storage size exceeds the configured disk usage limit, Cisco IOS XR software performs a set of actions.

When...	And...	Then Cisco IOS XR software...	And displays these syslog message...
the core dump folder storage size exceeds the configured disk usage limit value and	if the latest core file in the hard disk is more than 10 minutes older than the current crash time	deletes the older core files starting from oldest to newest until the core dump folder storage value is below the the default or configured disk usage limit value.	RP/0/RP0/CPU0:Jan 21 10:19:36.119 UTC: dumper[67458]: %OS-SYSLOG-3-LOG_ERR : sleep_14157 signature 4f24a379e430a4a8806975b2548 RP/0/RSP0/CPU0:Jan 21 10:19:36.251 UTC: dumper[399]: %OS-COREHELPER-6-DELETE_STALE_CORE : **Deleting /misc/disk1/coredumps/sleep_1042.by.3.20250121-100007.xr-vm_node0_RSP0_CPU0.34684.core.lz4 on active RP for reason: Freeing up old core in /misc/disk1/coredumps, attempting to copy new core file**
the core dump folder storage size exceeds the configured disk usage limit value and	if the latest core file in the hard disk is within 10 minutes than the current crash time	will stop saving the core files and delete any incoming core files.	RP/0/RP0/CPU0:/misc/disk1/coredumps]$RP/0/RSP0/CPU0:Jan 24 10:23:53.921 UTC: dumper[175]: %OS-COREHELPER-2-ABORT_COPY : Copy of sleep_19617.by.3.20250124-102353.xr-vm_node0_RSP0_CPU0.34684.core.lz4 to 0/RSP0/CPU0:/misc/disk1/**coredumps stoped for reason: Dumper disk usage exceeds 20% threshold of /misc/disk1. Potential /misc/disk1/coredumps usage with new core: 100.0%, seconds since last core: 36.0s. Deleting core from /misc/scratch/core.**

When...	And...	Then Cisco IOS XR software...	And displays these syslog message...
the hard disk is filled more than 90 percent of the total hard disk space	-	will stop saving the core files and delete any incoming core files as it can't delete any old core files.	`RP/0/RP0/CPU0:Mar 25 04:25:06.252 UTC: dumper[329]: %OS -COREHELPER-2-ABORT_COPY : Copy of sleep_3731_by_3_20080825-042505.x-vm_node0_RP0_CPU0_3484.core.lz4 to 0/RSP0/CPU0:/misc/disk1/coredumps` **`stoped for reason: Not enough available space in /misc/disk1 (less than 10%). Current /misc/disk1 bytes in use: 23590813696B, size of core file: 1471276B. Deleting core from /misc/scratch/core.`**

Result

Cisco IOS XR ensures that core dump storage remains within configured limits, prevents disk space exhaustion, and logs actions so operators can respond if necessary.

Configure core dump folder limit

Configure the storage usage limit for the core dump folder to prevent excessive accumulation of core dumps and ensure optimal disk space management.

Use this procedure to limit the storage usage of the core dump folder. This prevents excessive accumulation of core dumps and ensures optimal disk space management and system reliability.

Procedure

1. Set the core dump folder storage limit. This prevents excessive core dumps on the hard disk.

   Example

   ```
   RP/0/RP0/CPU0# configuration
   RP/0/RP0/CPU0(config)# exception disk-usage-limit 29
   RP/0/RP0/CPU0(config)# commit
   ```

   You can configure the core dump folder storage limit to range to 20% of the total hard disk space.

2. Run **show exception** command to view the set core dump folder storage limit.

Example

This example displays the core dump folder storage limit set to 20 percent of the total hard disk space.

```
RP/0/RP0/CPU0# show exception
 Exception path for choice 1 is not configured or removed
 Exception path for choice 2 is not configured or removed
 Exception path for choice 3 is not configured or removed
 Default fallback/copy path = /misc/disk1/coredumps
 Core dump usage on disk limited to 20%
```

The system enforces the specified storage limit for the core dump folder, ensuring disk space is optimally managed and preventing excess core dump accumulation.