

Bring-up 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. NCS 1014 completes the boot process using the pre-installed operating system (OS) image. If no image is available, NCS 1014 can be booted using the iPXE boot or an external bootable USB drive.

After booting, create the root username and password, and then use it 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.

- Boot Using Zero Touch Provisioning, on page 1
- Configure Management Interface, on page 14
- Boot NCS 1014, on page 16
- Boot NCS 1014 Using USB Drive, on page 16
- iPXE, on page 18
- Bring-Up Line Card, on page 22
- Configure NTP Server, on page 25

Boot Using Zero Touch Provisioning

ZTP allows you to provision the network device with day 0 configurations and supports both management ports and data ports.

ZTP provides multiple options such as:

- 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 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:

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

ZTP is initiated in one of the following ways:

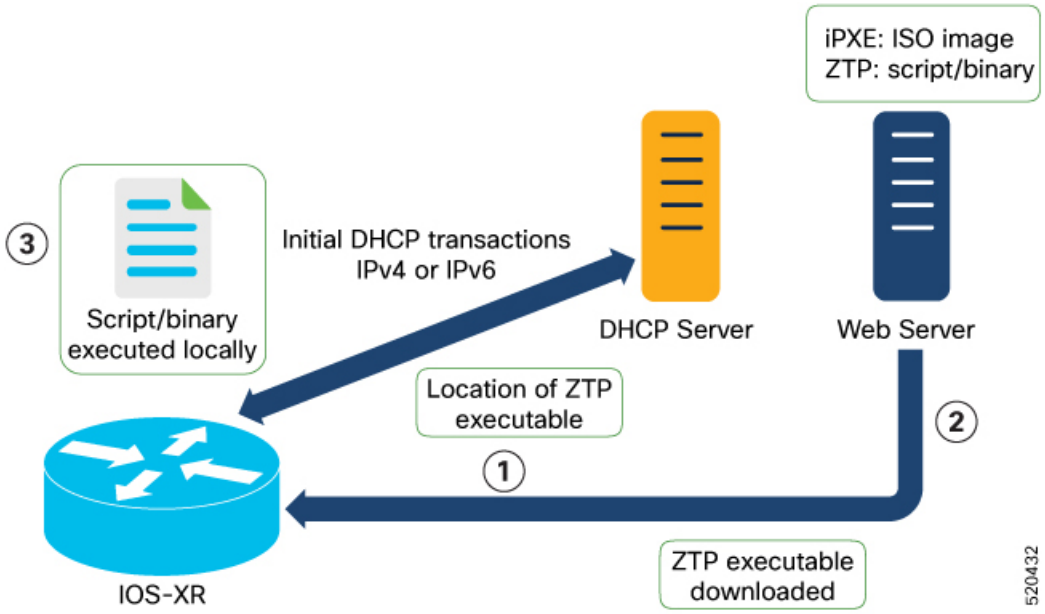
- **Automated Fresh Boot**: When you boot the device, the ZTP process initiates automatically if the device does not have a prior configuration. During the process, the router receives the details of the configuration file from the DHCP server. Use this method for devices that has no pre-loaded configuration. For more information, see Fresh Boot Using DHCP, on page 2.

  You must define the configuration file or the bootscript that is downloaded from the DHCP server.
- **Configuration File**: The first line of the file must contain **!! IOS XR configuration"**, to process the file as a configuration. If you are trying to bring up ten new nodes, you have to define ten configuration files. See Build your Configuration File, on page 4.
- **ZTP Bootscript**: Define the script to be executed on every boot. See Configure ZTP BootScript, on page 8 .
- **Manual Invocation using CLI:** Use this method when you want to forcefully initiate ZTP on a fully configured device using CLI. See Invoke ZTP Manually through CLI, on page 10.
- **Invocation using Reload Command:** Use this method when you want to forcefully initiate ZTP on a fully configured device using the **reload** command. See Invoke ZTP Through Reload, on page 11.

Fresh Boot Using DHCP

The ZTP process initiates when you boot the network device with an IOS XR image. The ZTP process starts only on a device without prior configuration.

This figure depicts the high-level workflow of the ZTP process:



1. ZTP sends DHCP request to fetch the ZTP configuration file or user script. To help the Bootstrap server uniquely identify the device, ZTP sends below DHCP option.
   - DHCP(v4/v6) client-id=Serial Number
   - DHCPv4 option 124: Vendor, Platform, Serial-Number
   - DHCPv6 option 16: Vendor, Platform, Serial-Number

   The following is the default sequential flow of the ZTP process defined in the configuration file. You can modify this sequence in the configuration file, if required.
   - ZTP sends IPv4 DHCP request first on all the management ports. If the request fails, then ZTP sends IPv6 DHCP request on all the management ports.
   - ZTP sends IPv4 DHCP request first on all the data ports. If the request fails, then ZTP sends IPv6 DHCP request on all the data ports.

2. DHCP server identifies the device and responds with DHCP response.

   DHCP server should be configured to respond with DHCP response and supply script/config location with one of the following DHCP options:
   - DHCPv4 using BOOTP filename.
   - DHCPv4 using Option 67 (bootfile-name).
   - DHCPv6 using Option 59 (OPT_BOOTFILE_URL).

3. The network device downloads the file from the web server using the URL location provided in the DHCP response.
4. The device receives a configuration file or script file from the HTTP server.

Note

- 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.

5. The device applies the configuration file or executes the script or binary in the default bash shell.
6. The Network device is now up and running.

Build your Configuration File

Based on the business need, you can use a configuration or script file to initiate the ZTP process.

The configuration file content starts with *!! IOS XR*.

The following is the sample configuration file. You can automate all the configurations. For more information on creating ZTP configuration file, refer ZTP Configuration Files Creation.

```
!! 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
```

Configure ZTP BootScript

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. The first line of the file must contain **#! /bin/bash** or **#! /bin/sh** for ZTP to process the file as script.

You can either use the ZTP bash script or the ZTP configuration file.

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

```
Router#configure
Router(config)#ztp bootscript /disk0:/myscript
Router(config)#commit
```

To ensure that we have connectivity in the third-party namespace for applications to use, the above configuration waits for the first data plane interface to be configured and wait an extra minute for the management interface to be configured with an IP address. If the delay is not desired, use:

```
Router#configure
Router(config)#ztp bootscript preip /disk0:/myscript
Router(config)#commit
```



Note When the above command is first configured, you will be prompted if you wish to invoke it now. The prompt helps with testing.

This is the example 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://10.33.0.51/P1B_DT_08/ncs1010-x64.iso";
} else {
filename "http://10.33.0.51/P1B_DT_08/startup.cfg";
}
fixed-address 10.33.0.19;
}
```

The following is the sample content of the 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"`
```

The following is the sample content of the 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
```

Invoke ZTP Manually through CLI

Manual ZTP can be invoked through CLI commands. This manual way helps you to provision the router in stages. Ideal for testing out ZTP configuration without a reboot. If you want to invoke a ZTP on an interface (data ports or management ports), you need not bring up and configure the interface first. You can execute the **ztp initiate** command, even if the interface is down, so that the ZTP script brings it up and invoke *dhclient*. ZTP can run on all interfaces irrespective of whether the interfaces are up or not.

Use the **ztp initiate**, **ztp terminate**, and **ztp clean** commands to force ZTP to run on more interfaces.

- **ztp initiate**—Invokes a new ZTP DHCP session. Logs can be found in the */disk0:/ztp/ztp.log* location.
- **ztp terminate**—Terminates current ZTP sessions.
- **ztp clean**—Removes only the ZTP state files.

The log file *ztp.log* is saved in the */var/log/ztp.log* folder, and a copy of log file is available in the */disk0:/ztp/ztp.log* location using a soft link. However, executing **ztp clean** clears files saved on disk and not from the **/var/log/ztp.log** folder where current ZTP logs are saved. To get a log from current ZTP run, you must manually remove the ZTP log file from **/var/log/ztp.log**.

SUMMARY STEPS

1. (optional) **ztp clean**
2. **ztp initiate**
3. (Optional) **ztp terminate**

DETAILED STEPS

Procedure

Step 1 (optional) **ztp clean**

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#
```

Removes all the ZTP logs and saved settings.

Step 2 **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#
```

Reboots the Cisco NCS 1014 system.

Use the **show logging** command or see the */var/log/ztp.log* to check progress.

Step 3 (Optional) **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#
```

Terminates the ZTP process.

Invoke ZTP Through Reload

The ZTP process can be automatically invoked using the **reload** command.

SUMMARY STEPS

1. **configure**
2. **commit replace**
3. **ztp clean**
4. **reload**

DETAILED STEPS

Procedure

Step 1 **configure**

Example:

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

Enters the configuration mode.

Step 2 **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)#
```

Removes the entire running configuration.

Step 3 **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#
```

Removes all the ZTP logs and saved settings.

Step 4 **reload**

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]
```

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

```
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
```

Reboots the Cisco NCS 1014 system.

ZTP Logging

ZTP logs its operation on the flash file system in the */disk0:/ztp/* directory. ZTP logs all the transactions with the DHCP server and all the state transitions.

The following example displays the execution of a simple configuration script downloaded from a data interface using the command **ztp initiate interface MgmtEth 0/RP0/CPU0/0 verbose**. This script unshuts all the interfaces of the system and configures a load interval of 30 seconds on all of them.

```
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
```

Generate Tech Support Information for ZTP

When you have a problem that you cannot resolve in the ztp process, contact the Cisco Systems technical support representative. To analyze a problem, your technical support representative needs certain information about the situation and the symptoms that you are experiencing. To speed up the resolution, collect the necessary data before you contact your representative.

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

Example:

```
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 the above example, the tech support information is saved as .tgz file in the specified location. This information can be shared with the Cisco Technical Support representatives for troubleshooting the ztp process.

Configure Management Interface

The management interface can be used for system management and remote communication. To use the management interface for system management, you must configure an IP address and subnet mask. To use the management interface for remote communication, you must configure a static route.

Before you begin

- Consult your network administrator to procure IP addresses and a subnet mask for the management interface.

- Ensure that the management interface is connected to the management network.

Procedure

Step 1 **configure**

Example:

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

Enters IOS XR configuration mode.

Step 2 **interface mgmtEth** *rack/slot/instance/port*

Example:

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

Enters interface configuration mode for the management interface.

Step 3 **ipv4 address** *ipv4-address subnet-mask*

Example:

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

Assigns an IP address and a subnet mask to the management interface.

Step 4 **no shutdown**

Example:

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

Places the management interface in an "up" state.

Step 5 **exit**

Example:

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

Exits the management interface configuration mode.

Step 6 **router static address-family ipv4 unicast** *0.0.0.0/0 default-gateway*

Example:

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

Specifies the IP address of the default gateway to configure a static route. This IP address must be used for communication with devices on other networks.

Step 7 Use the **commit** or **end** command.

commit-Saves the configuration changes and remains within the configuration session.

end-Prompts user to take one of these actions:

- **Yes**-Saves configuration changes and exits the configuration session.
- **No**-Exits the configuration session without committing the configuration changes.

- **Cancel**-Remains in the configuration session without committing the configuration changes.

Boot NCS 1014

Use the console port to connect to NCS 1014. By default, the console port connects to the XR mode. If necessary, you can establish subsequent connections through the management port, after it is configured.

Procedure

Step 1 Connect a terminal to the console port of the RP.

Step 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.

Step 3 Power on the NCS 1014.

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

Step 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 NCS 1014 more time to complete the initial boot procedure; then press **Enter**.

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

Boot NCS 1014 Using USB Drive

The bootable USB drive is used to reimage NCS 1014 for system upgrade or to boot the NCS 1014 in case of boot failure. A bootable USB drive is created by copying a compressed boot file into a USB drive. The USB drive becomes bootable after the contents of the compressed file are extracted.

You can complete this task using the Windows, Linux, or MAC operating systems available on your local machine. The exact operation to be performed for each generic step that is outlined here depends on the operating system in use.

Use this task to boot the NCS 1014 using the USB drive.

Before you begin

- You need a USB drive with a storage capacity of at least 4 GB.
- The USB drive should have 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

Step 1 Connect the USB drive to your local machine and format it with the FAT32 file system.

Step 2 Copy the compressed boot file to the USB drive.

Step 3 Verify that the copy operation is successful. To verify, compare the file size at source and destination. Also, verify the MD5 checksum value.

Step 4 Extract the content of the compressed boot file by unzipping it in the USB drive. This 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.

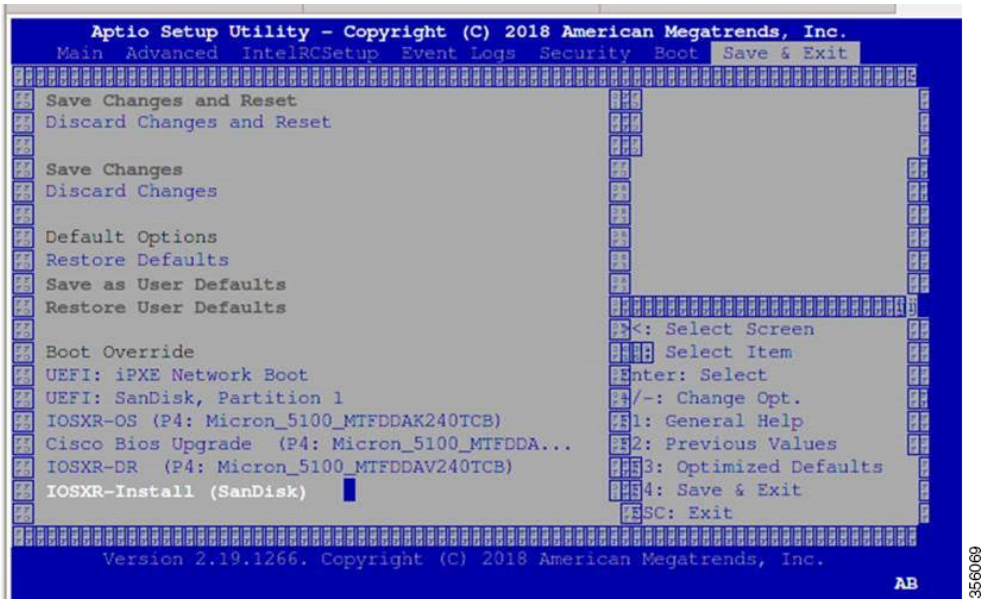
Step 5 Insert the USB drive in one of the USB ports of NCS 1014.

Step 6 Reboot NCS 1014 using power cycle or console.

Note
Use the **reload bootmedia usb noprompt** command to boot the NCS1010 from the USB. If you are using the **reload bootmedia usb noprompt** command, then you can skip the remaining steps.

Step 7 Press **Esc** to enter BIOS.

Step 8 Select the **Save & Exit** tab of BIOS.



Step 9 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...
```

Step 10 Remove the USB drive after the `Rebooting the system after installation` message is displayed. The NCS 1014 reboots automatically.

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

iPXE

iPXE is a Preboot Execution Environment that is included in the network card of the management interfaces and works at the system firmware (UEFI) level of the chassis. iPXE is used to reimage the system, and boot the chassis in case of boot failure or in the absence of a valid bootable partition. iPXE downloads the ISO image, proceeds with the installation of the image, and finally bootstraps inside the new installation.

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.

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

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

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

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

Setup DHCP Server

A DHCP server must be configured for IPv4, IPv6, or both communication protocols.

Note For DHCPv6, a routing advertisement (RA) message must be sent to all nodes in the network that indicates which method is to be used to obtain the IPv6 address. Configure 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;
        };
};
```

To setup a DHCP server:

1. Create the `dhcpd.conf` file (for IPv4, IPv6 or both communication protocols), `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, location of the ISO install file, location of the provisioning configuration file, serial number, MAC address of the chassis.
2. Test the server when the DHCP server is running:

   For example, for ipv4:

   a. Use MAC address of the chassis:

      Ensure that the above configuration is successful.

   b. Use serial number of the chassis:

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

      The serial number of the chassis is derived from the BIOS and is used as an identifier.

      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;
          }
      ```

iPXE boot process

Follow these steps to run the iPXE boot process using the CLI and BIOS interface.

Before you begin

Before you perform the iPXE boot, ensure that:

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

Procedure

Step 1 (Preferred method) Run the **reload bootmedia network location all** command or **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!
```

Step 2 (Alternative method) Initiate the boot process using the **BIOS** interface.

a) Reboot 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.
```

Bring-Up Line Card

Use the following procedure to bring-up the NCS1014 line cards:

Procedure

Step 1 Insert the line card into the slot.

Step 2 Wait until the line card LED turns Green.

Step 3 Check 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
```

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

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

- When a slot is not in use, we recommend inserting a filler to allow proper airflow across the line cards to maintain an optimal system temperature.
- When a port is not in use, we recommend inserting a clip to maintain an optimal card temperature.
- Ensure to secure the line card in the chassis by tightening the top and bottom screws.

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

Configure NTP Server

Understand NTP

Table 1: 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 uses the User Datagram Protocol (UDP) as its transport protocol. All NTP communication uses Coordinated Universal Time (UTC). An NTP network usually receives its time from an authoritative time source, such as a radio clock or an atomic clock attached to a time server. NTP distributes this time across the network.

NTP uses the concept of a “stratum” to describe how many NTP hops away a machine is from an authoritative time source. A “stratum 1” time server typically has an authoritative time source (such as a radio or atomic clock, or a GPS time source) directly attached, a “stratum 2” time server receives its time through NTP from a “stratum 1” time server, and so on.

The communications between machines running NTP (known as associations) are usually statically configured; each machine is given the IP address of all machines with which it should form associations. Accurate timekeeping is made possible by exchanging NTP messages between each pair of machines with an association.

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 Clock with NTP Server

IOS XR has an independent system clock that must be synchronized with an NTP server to prevent deviation from true time.

To sync the clock with an NTP server, perform these steps:

Before you begin

Procedure

Step 1 **configure**

Example:

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

Enters the configuration mode.

Step 2 **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
```

Synchronizes the console clock with the specified NTP server.

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

Step 3 **commit**

Example:

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

Commits the configuration.

Step 4 **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
```

Displays the status of NTP associations.

Step 5 **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.
```

Verifies that the clock is synchronized with the NTP server.

Verify the Status of the External Reference Clock

This task explains how to verify the status of NTP components.

Note The commands can be entered in any order.

Procedure

Step 1 **show ntp associations [detail] [location** *node-id***]**

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
```

Step 2 **show ntp status [location** *node-id***]**

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

For NTP troubleshooting information, see here.