Bring-up Cisco NCS 1001

After installing the hardware, boot the Cisco NCS 1001 system. You can connect to the XR console port and power on the system. NCS 1001 completes the boot process using the pre-installed operating system (OS) image. If no image is available, NCS 1001 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. From the XR console, access the System Admin console to configure system administration settings.

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.

Cisco NCS 1001 Overview

Cisco NCS 1001 (NCS1001-K9) is 1 RU chassis that addresses the growing bandwidth needs of data center DWDM applications. It provides a DWDM line system that is optimized for data center environments and is optimized for point-to-point applications at maximum capacity. NCS 1001 supports up to three optical modules. The modules can be amplifiers, protection switching modules, or OTDR modules.

NCS 1001 has the following components:

- Removable control card
- Four removable fans
- Two removable 600W AC power supply modules (PSU)
- Three slots for optical modules. The Optical Amplifier Module (NCS1K-EDFA), Protection Switching Module (NCS1K-PSM), and Optical Time Domain Reflectometer (NCS1K-OTDR) module can be inserted in these slots.

  The optical modules can be inserted in slots 1 to 3. The optical modules can be inserted and removed from the slots while the system is operational. In amplified configuration, the Optical Amplifier module can be inserted in any slot. In protected configuration, the protect Optical Amplifier module is inserted in slot 1, Protection Switching Module in slot 2, and working Optical Amplifier module in slot 3. The OTDR line card can be inserted in any slot.

Boot Sequence

The boot sequence in NCS 1001 that you need to follow is:

1. Boot using SSD (hard disk)
2. Boot using USB drive
3. Boot using iPXE

If there is no bootable image in all three boot options, reboot the system.

Boot NCS 1001

Use the console port to connect to NCS 1001. By default, the console port connects to the XR mode. If required, subsequent connections can be established 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, 8 data bits, 1 stop bit and no parity.

Step 3 Power on the NCS 1001.

To turn on the power shelves, press the power switch up. As NCS 1001 boots up, the boot process details are displayed 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 the NCS 1001 more time to complete the initial boot procedure; then press **Enter**.

Important If the boot process fails, it may be because the pre-installed image on the NCS 1001 is corrupt. In this case, the NCS 1001 can be booted using an external bootable USB drive.

Boot NCS 1001 Using USB Drive

The bootable USB drive is used to re-image the NCS 1001 for the purpose of system upgrade or to boot the NCS 1001 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.

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

Before you begin

- You need a USB drive with a storage capacity of at least 4 GB.
- NCS 1001 software image can be downloaded from this location.
- Copy the compressed boot file from the software download page at cisco.com to your local machine. The file name for the compressed boot file is in the format *ncs1001-usb-boot-<release_number>.zip*. For example, *ncs1001-usb-boot-7.1.1.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.

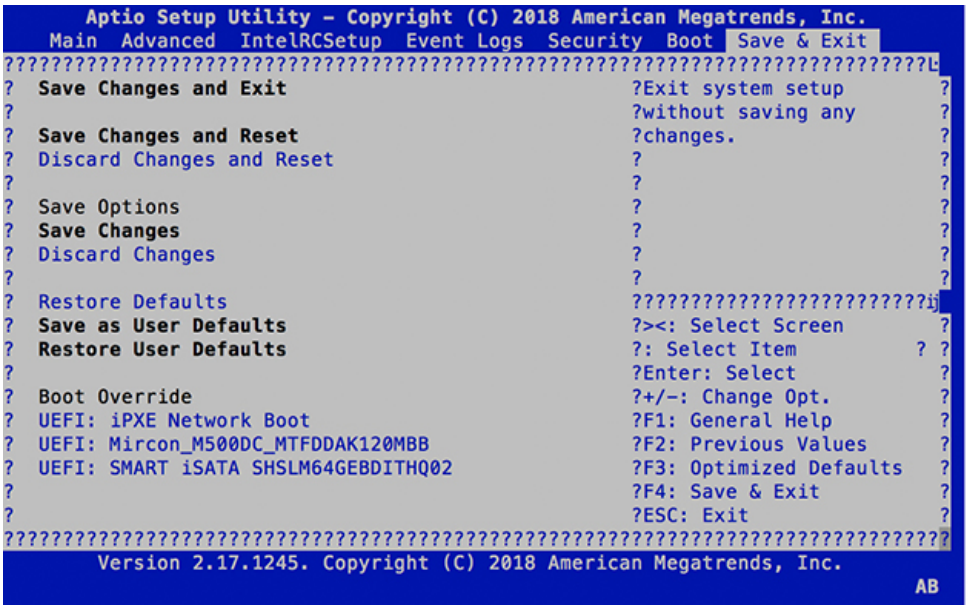
Note The content of the zipped file ("EFI" and "boot" directories) must be extracted directly in the root folder of the USB drive. If the unzipping application places the extracted files in a new folder, move the "EFI" and "boot" directories to the root folder of the USB drive.

Step 5 Insert the USB drive in one of the four USB ports of NCS 1001 after unplugging any other USB drive.

Step 6 Reboot NCS 1001 using power cycle or console.

Step 7 Press Esc to enter BIOS.

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



Step 9 Choose **UEFI based USB device**.

The system detects USB and boots the image from USB.

```
Admin Console:
GNU GRUB version 2.00
Press F2 to goto grub Menu..
Booting from USB..
Loading Kernel..

Validating End Entity Certificate...

Validating SubCA Certificate...

Validating Root Certificate...
Loading initrd..

Validating End Entity Certificate...

Validating SubCA Certificate...

Validating Root Certificate...
CiscoSec: Image signature verification completed.
XR Console:
CiscoSec: Image signature verified.
[    9.957281] i8042: No controller found
Starting udev
udevd[972]: failed to execute '/etc/udev/scripts/network.sh' '/etc/udev/scripts/network.sh':
 No such file or directory
Populating dev cache
Running postinst /etc/rpm-postinsts/100-dnsmasq...
update-rc.d: /etc/init.d/run-postinsts exists during rc.d purge (continuing)
 Removing any system startup links for run-postinsts ...
  /etc/rcS.d/S99run-postinsts
Configuring network interfaces... done.
```

Step 10 Remove the USB drive. The NCS 1001 reboots automatically.

```
Setting maximal mount count to -1
Setting interval between checks to 0 seconds
Fri Dec 11 20:35:56 UTC 2015: Install EFI on /dev/mb_disk4
Fri Dec 11 20:35:57 UTC 2015: Install finished on mb_disk
Rebooting system after installation ...
[  116.973666] reboot: Restarting system

Version 2.17.1245. Copyright (C) 2015 American Megatrends, Inc.
BIOS Date: 11/29/2015 12:02:45 Ver: 0ACBZ1110
Press <DEL> or <ESC> to enter setup.
CiscoSec: Image signature verified.


GNU GRUB version 2.00
Press F2 to goto grub Menu..
Booting from Disk..
Loading Kernel..

Validating End Entity Certificate...

Validating SubCA Certificate...

Validating Root Certificate...
Loading initrd..

Validating End Entity Certificate...

Validating SubCA Certificate...

Validating Root Certificate...
CiscoSec: Image signature verification completed.
Initrd, addr=0xff69a000, size=0x955cb0
[    1.745686] i8042: No controller found
```

Boot Using iPXE

iPXE is a pre-boot 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 re-image 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 boot loader 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. iPXE must be defined in the DHCP server configuration file.

Note For IPv6 configuration, see **Step 2** in Setup DHCP Server, on page 6.

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 once the DHCP server is running:

   For example, for ipv4:

   a. Use MAC address of the chassis:

   ```
   host ncs1001
   {
   hardware ethernet ab:cd:ef:01:23:45;
   fixed-address <ip address>;
   filename "http://<httpserver-address>/<path-to-image>/ncs1001-mini-x.iso";
   }
   ```

   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 40:55:39:56:0c:e8;
          if exists user-class and option user-class = "iPXE" {
          filename "http://10.89.205.127/box1/ncs1001-mini-x-7.1.1.iso";
       } else {
           filename "http://10.89.205.127/box1/StartupConfig.cfg";
       }
       fixed-address 10.89.205.202;
    }
```

For example, for ipv6:

1. Use serial number of the chassis:

   The serial number of the chassis is derived from the XR command 'show inventory' and is used as an identifier. The S/N must be converted in an ipv6 DUID format.

   ```
   host ncs1001 {
     host-identifier  option dhcp-client-identifier "<IPV6-DIUD>";
     filename
   "<http,tftp>://<IPv6-dhcp-server-address>/<hardware-platform>-mini-x.iso";
   }
   ```

   Example

   ```
   host mys-p1b-212 {

      # PID: NCS1001-K9        , VID: V01, SN: CAT2018B02M

      host-identifier option dhcp6.client-id
   00:02:00:00:00:09:43:41:54:32:30:31:38:42:30:32:4d:00;

      if exists dhcp6.user-class and substring(option dhcp6.user-class, 2, 4) =
   "iPXE" {
        option dhcp6.bootfile-url
   "http://[2001:420:4491:2005::227:11]/rosco/mys-p1b-212/ncs1001-mini-mod.iso";
      }
      else {
        option dhcp6.bootfile-url
   "http://[2001:420:4491:2005::227:11]/rosco/mys-p1b-212/ztp.ipv6.sh";
      }
   }
   ```

   Here a simple shell script that converts from chassis S/N to DUID

   Example

   ```
   #! /bin/sh

   # Comply with IPv6 EN-DUID - see DHCPv6 RFC for detail
   ## 00:02 = Indicate Enterprise number based DUID
   ## 00:00:00:09 = CIsco Enterprise number
   ## 00 = Terminator

   SERIALNUMBER=$1
   en_duid=0002
   cisco_en=00000009

   sn=`xxd -l11 -pu <<< ${SERIALNUMBER}`
   null=00
   DHCPv6_DUID=`sed 's/\(..\)/\1:/g;s/:$//' <<< ${en_duid}${cisco_en}${sn}${null}`

   echo $DHCPv6_DUID
   ```

Boot Using iPXE

Before you use the iPXE boot, ensure that:

- DHCP server is set and is running.
- You have logged in to the System Admin console using the **admin** command.

Run the following command to invoke the iPXE boot process to reimage the chassis:

```
hw-module location all bootmedia network reload
```

Example:

```
sysadmin-vm:0_RP0# hw-module location all bootmedia network reload
Wed Dec 23 15:29:57.376 UTC
Reload hardware module ? [no,yes]
```

The following example shows the output of the command:

```
iPXE 1.0.0+ (3e573) -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS HTTP TFTP VLAN EFI ISO9660 NBI Menu
Trying net0...
net0: c4:72:95:a6:14:e1 using dh8900cc on PCI01:00.1 (open)
[Link:up, TX:0 TXE:0 RX:0 RXE:0]
Configuring (net0 c4:72:95:a6:14:e1)................... Ok << Talking to DHCP/PXE server to
 obtain network information
net0: 10.37.1.101/255.255.0.0 gw 10.37.1.0
net0: fe80::c672:95ff:fea6:14e1/64
net0: 2001:1800:5000:1:c672:95ff:fea6:14e1/64 gw fe80::20c:29ff:fefb:b9fe
net1: fe80::c672:95ff:fea6:14e3/64 (inaccessible)
Next server: 10.37.1.235
Filename: http://10.37.1.235/ncs1001/ncs1001-mini-x.iso
http://10.37.1.235/ncs1001/ncs1001-mini-x.iso ncs1000/ncs1001-mini-x.iso... 58% << Downloading
 file as indicated by DHCP/PXE server to boot install image
```

Boot NCS 1001 Using Zero Touch Provisioning (ZTP)

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 Cisco 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 providers infrastructures effortlessly. Following are the added benefits of using ZTP:

- ZTP helps you to remotely provision a router anywhere in the network. Thus eliminates the need to send an expert to deploy network devices and reduces IT cost.

- Automated provisioning using ZTP can remove delay and increase accuracy and thus is cost-effective and provides better customer experience.

  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 by hand, you can reset a system to well-known working status.

Prerequisites:

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

Note For IPv6 configuration, see **Step 2** in Setup DHCP Server, on page 6.

ZTP is initiated in one of the following ways:

- **Automated Fresh Boot:**

  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 preloaded configuration. See Fresh Boot Using DHCP for ZTP, on page 9.

  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 11.

- **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 12.

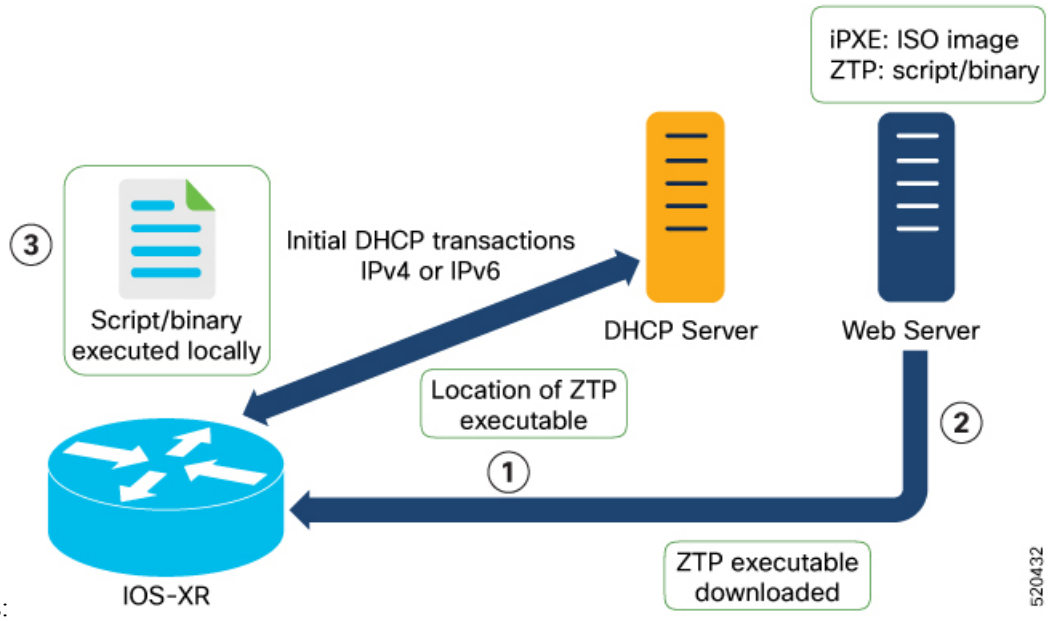
Fresh Boot Using DHCP for ZTP

When you boot the device, the ZTP process initiates automatically if the device does not have a prior configuration.

Fresh Boot Using DHCP

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.

This image depicts the high-level work flow of the ZTP process:



The ZTP process initiates when you boot the network-device with an IOS-XR image. The process starts only on the device that doesn't have a prior configuration.

Here is the high-level work flow of the ZTP process for the Fresh boot:

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:
   - ZTP sends IPv4 DHCP request first on all the management port. In case there is a failure, then ZTP sends IPv6 DHCP request on all the management port.
   - ZTP sends IPv4 DHCP request first on all the data port. In case there is a failure, then ZTP sends IPv6 DHCP request on all the data port.

     The default sequential flow is defined in configuration file and you can modify the sequence using the configuration file.

2. DHCP server identifies the device and responds with DHCP response using one of the following options:

   DHCP server should be configured to respond with the DHCP options.
   - DHCPv4 using BOOTP filename to supply script/config location.
   - DHCPv4 using Option 67 (bootfile-name) to supply script/config location.
   - DHCPv6 using Option 59 (OPT_BOOTFILE_URL) to supply script/config location
   - For example, for IPv4:

```
host mys-p1b-209-Ztp {
option dhcp-client-identifier 00:02:00:00:00:09:43:41:54:32:31:30:37:42:30:4d:44:00;

  filename "http://192.0.2.10/ncs1000/mys-p1b-209-Ztp/-mini-x.iso";

}
```

For example, for IPv6:

```
host mys-p1b-209-Ztp {
                # PID: NCS1001-K9        , VID: V01, SN: CAT2107B0MD
                # needed for ztp
                host-identifier option dhcp6.client-id
00:02:00:00:00:09:43:41:54:32:31:30:37:42:30:4d:44:00;


                if exists dhcp6.user-class and substring(option dhcp6.user-class,
2, 4) = "iPXE" {
                        option dhcp6.bootfile-url
"http://[2001:DB8::4491:2005::227:11]/ncs1001/mys-p1b-209/ncs1001-mini-mod.iso";
                }
                else {
                        option dhcp6.bootfile-url
"http://[2001:DB8::4491:2005::227:11]/ncs1001/mys-p1b-209/ztp.ipv6.sh";
                }
        }
```

3. The network device downloads the file from the web server using the URI location that is 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 for IPv4. You can automate all the configurations.

```
!! IOS XR Configuration version = 6.3.2
!
telnet vrf default ipv4 server max-servers 20
!
vty-pool default 0 20 line-template default
!
interface MgmtEth0/RP0/CPU0/0
 ipv4 address dhcp
 no shutdown
```

```
!
router static
 address-family ipv4 unicast
  0.0.0.0/0 10.77.132.1
!
end
```

The following is the sample configuration file for IPv6.

```
!! IOS XR Configuration version = 7.10.1
!
telnet vrf default ipv6 server max-servers 20
!
vty-pool default 0 20 line-template default
!
interface MgmtEth0/RP0/CPU0/0
 ipv6 address dhcp6
 no shutdown
!
router static
 address-family ipv6 unicast
  2001:db8:1000::/36 2001:db8:2000:2::1
!
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 port), you don't have to bring up and configure the interface first. You can execute the ztp initiate command, even if the interface is down, ZTP script brings it up and invoke dhclient. So ZTP could run over all interfaces no matter it is up or down.

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

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

The log file ztp.log is saved in /var/log/ztp.log folder, and a copy of log file is available at /disk0:/ztp/ztp.log location using a soft link. However, executing ztp clean clears files saved on disk and not on /var/logztp.log folder where current ZTP logs are saved. In order to have a log from current ZTP run, you must manually clear the ZTP log file from /var/log/ztp.log folder.

Procedure

Step 1 (optional) **ztp clean**

Example:

```
RP/0/RP0/CPU0:ios#ztp clean
Fri Apr 29 06:49:29.760 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.
```

Removes all the ZTP logs and saved settings.

Step 2 **ztp initiate**

Example:

```
RP/0/RP0/CPU0:ios#ztp initiate
Fri Jun 17 11:44:08.791 UTC
Initiating 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#
```

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

Reboots the Cisco NCS 1001 system.

Step 3 (Optional) **ztp terminate**

Example:

```
RP/0/RP0/CPU0:ios#ztp terminate
Fri Apr 29 06:38:59.238 UTC
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
No ZTP process running
```

Terminates the ZTP process.

Note For IPv6 configuration, see **Step 2** in Setup DHCP Server, on page 6.

Invoke ZTP Through Reload

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

Procedure

Step 1 **configure**

Example:

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

Enters the configuration mode.

Step 2 **commit replace**

Example:

```
Fri Apr 29 06:48:46.236 UTC
RP/0/RP0/CPU0:ios(config)#commit replace
Fri Apr 29 06:48:53.199 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)#
RP/0/RP0/CPU0:ios(config)#end
```

Removes the entire running configuration.

Step 3 **ztp clean**

Example:

```
RP/0/RP0/CPU0:ios#ztp clean
Fri Apr 29 06:49:29.760 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.
```

Removes all the ZTP logs and saved settings.

Step 4 **reload**

Example:

```
RP/0/RP0/CPU0:ios#admin reload location 0/RP0 all
Fri Apr 29 06:50:29.760 UTC
Reload node ? [no,yes] yes
```

```
RP/0/RP0/CPU0:ios#admin hw-module location 0/RP0 reload
Fri Apr 29 06:52:29.760 UTC
Reload hardware module ? [no,yes] yes
```

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

```
RP/0/RP0/CPU0:Apr 29 06:55:33.242 UTC: pyztp2[377]: %INFRA-ZTP-4-CONFIG_INITIATED : ZTP has
 initiated config load and commit operations
RP/0/RP0/CPU0:Apr 29 06:55:39.263 UTC: ifmgr[381]: %PKT_INFRA-LINK-3-UPDOWN : Interface
GigabitEthernet0/0/0/0, changed state to Down
RP/0/RP0/CPU0:Apr 29 06:55:39.287 UTC: osa_driver[183]: %PKT_INFRA-FM-4-FAULT_MINOR :
ALARM_MINOR :PROV-INPROGRESS :DECLARE :GigabitEthernet0/0/0/0:
RP/0/RP0/CPU0:Apr 29 06:55:39.287 UTC: osa_driver[183]: %PKT_INFRA-FM-4-FAULT_MINOR :
ALARM_MINOR :PROV-INPROGRESS :DECLARE :Osc0/0/0/0:
RP/0/RP0/CPU0:Apr 29 06:55:39.287 UTC: ifmgr[381]: %PKT_INFRA-LINK-3-UPDOWN : Interface
GigabitEthernet0/0/0/0, changed state to Up
RP/0/RP0/CPU0:Apr 29 06:55:39.716 UTC: osa_driver[183]: %PKT_INFRA-FM-4-FAULT_MINOR :
ALARM_MINOR :PROV-INPROGRESS :CLEAR :Osc0/0/0/0:
RP/0/RP0/CPU0:Apr 29 06:55:39.728 UTC: osa_driver[183]: %PKT_INFRA-FM-4-FAULT_MINOR :
ALARM_MINOR :PROV-INPROGRESS :CLEAR :GigabitEthernet0/0/0/0:
RP/0/RP0/CPU0:Apr 29 06:55:47.904 UTC: osa_driver[183]: %PKT_INFRA-FM-4-FAULT_MINOR :
ALARM_MINOR :PROV-INPROGRESS :DECLARE :Ots0/0/0/1:

User Access Verification

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

Reboots the Cisco NCS 1001 system.

ZTP Logging

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

The following example displays the execution of a simple configuration script that is downloaded from a management interface:

```
2023-05-12 14:43:18,406   5845 [Env          ] INF: MgmtDhcp4Fetcher fetcher created.
2023-05-12 14:43:18,482   5845 [Engine       ] DEB: MgmtDhcp4Fetcher, current state:idle.
Processing work: [privileged] start an engine. done = False
2023-05-12 14:43:18,484   5845 [Engine       ] INF: MgmtDhcp4Fetcher, current state:active:
 state changed to active
2023-05-12 14:43:18,508   5845 [Engine       ] DEB: ZAdmin, current state:active. Processing
 work: Monitor fetcher work for ZAdmin. done = False
2023-05-12 14:43:18,585   5845 [Engine       ] DEB: MgmtDhcp4Fetcher, current state:active.
 Processing work: epoch work. done = False
2023-05-12 14:43:18,586   5845 [Engine       ] DEB: MgmtDhcp4Fetcher, current state:active.
 Processing work: [privileged] getting engine status. done = False
2023-05-12 14:43:18,587   5845 [Engine       ] DEB: MgmtDhcp4Fetcher, current state:active.
 Processing work: Bringing up interfaces before next retry. done = False
2023-05-12 14:43:18,589   5845 [Port         ] DEB: <Port count=1>: bringing interface(s)
up "MgmtEth0/RP0/CPU0/0"
2023-05-12 14:43:18,624   5845 [Port         ] DEB: Saving 1 interfaces to
/disk0:/ztp/xr_config/interface_list
2023-05-12 14:43:18,626   5845 [Engine       ] DEB: MgmtDhcp4Fetcher, current state:active.
 Processing work: Filtering up interfaces for MgmtDhcp4Fetcher. done = False
2023-05-12 14:43:18,627   5845 [Management   ] DEB: Determining operstate of interface:
brXRmgmt1
2023-05-12 14:43:18,666   5845 [Port         ] DEB: Filtered up interfaces: [Name:
MgmtEth0/RP0/CPU0/0 (Type: Management) (Up: False)]
2023-05-12 14:43:18,670   5845 [Engine       ] DEB: MgmtDhcp4Fetcher, current state:active.
 Processing work: Setup interfaces for MgmtDhcp4Fetcher. done = False
2023-05-12 14:43:18,671   5845 [Env          ] INF: Env::getVlanIDs: vlan.mode:2
2023-05-12 14:43:18,672   5845 [Engine       ] DEB: MgmtDhcp4Fetcher, current state:active.
 Processing work: Start Dhclient for MgmtDhcp4Fetcher. done = False
2023-05-12 14:43:18,687   5845 [Port         ] DEB: <Dhclient count=1>: started dhclient
using "ip netns exec xrnns /sbin/dhclient -4 -cf /etc/dhcp/dhclient.conf.ztp -lf
/var/lib/dhcp/dhclient.leases.ztp -sf /etc/dhcp/dhclient-script.ztp2 brXRmgmt1"
2023-05-12 14:43:19,090   5845 [Engine       ] DEB: ZAdmin, current state:active. Processing
 work: Checking for username configuration. done = False
2023-05-12 14:43:19,630   5845 [Engine       ] DEB: ZAdmin, current state:active. Processing
 work: Monitor fetcher work for ZAdmin. done = False
2023-05-12 14:43:19,692   5845 [Engine       ] DEB: MgmtDhcp4Fetcher, current state:active.
 Processing work: [privileged] getting engine status. done = False
2023-05-12 14:43:20,696   5845 [Engine       ] DEB: ZAdmin, current state:active. Processing
 work: Monitor fetcher work for ZAdmin. done = False
2023-05-12 14:43:20,797   5845 [Engine       ] DEB: MgmtDhcp4Fetcher, current state:active.
 Processing work: [privileged] getting engine status. done = False
2023-05-12 14:43:21,099   5845 [Engine       ] DEB: ZAdmin, current state:active. Processing
 work: Sending standby sync message. done = False
2023-05-12 14:43:21,212   5845 [Engine       ] DEB: ZAdmin, current state:active. Processing
 work: [privileged] getting engine status. done = False
2023-05-12 14:43:21,700   5845 [Engine       ] DEB: MgmtDhcp4Fetcher, current state:active.
 Processing work: Monitor dhclient for MgmtDhcp4Fetcher on interface ['brXRmgmt1']. done =
 False
2023-05-12 14:43:21,704   5845 [MgmtDhcp4Fetcher] DEB: Received DHCP4 response
2023-05-12 14:43:21,705   5845 [MgmtDhcp4Fetcher] INF: (dhclient env) reason=BOUND
```

```
2023-05-12 14:43:21,706   5845 [MgmtDhcp4Fetcher] INF: (dhclient env) interface=brXRmgmt1
2023-05-12 14:43:21,706   5845 [MgmtDhcp4Fetcher] INF: (dhclient env)
new_ip_address=10.58.227.177
2023-05-12 14:43:21,707   5845 [MgmtDhcp4Fetcher] INF: (dhclient env)
new_network_number=10.58.227.0
2023-05-12 14:43:21,708   5845 [MgmtDhcp4Fetcher] INF: (dhclient env)
new_subnet_mask=255.255.255.0
2023-05-12 14:43:21,708   5845 [MgmtDhcp4Fetcher] INF: (dhclient env)
new_broadcast_address=10.58.227.255
2023-05-12 14:43:21,709   5845 [MgmtDhcp4Fetcher] INF: (dhclient env) new_routers=10.58.227.1
2023-05-12 14:43:21,710   5845 [MgmtDhcp4Fetcher] INF: (dhclient env)
new_dhcp_server_identifier=10.58.227.11
2023-05-12 14:43:21,711   5845 [MgmtDhcp4Fetcher] INF: (dhclient env)
new_domain_name=cisco.com
2023-05-12 14:43:21,711   5845 [MgmtDhcp4Fetcher] INF: (dhclient env)
new_domain_name_servers=144.254.71.184 173.38.200.100
2023-05-12 14:43:21,712   5845 [MgmtDhcp4Fetcher] INF: (dhclient env)
new_filename=http://10.58.227.11/rosco/mys-p1b-212/ztp.sh
2023-05-12 14:43:21,713   5845 [MgmtDhcp4Fetcher] INF: (dhclient env) new_ip6_prefixlen=64
2023-05-12 14:43:21,714   5845 [Engine      ] INF: MgmtDhcp4Fetcher, current state:active,
 exit code:success
2023-05-12 14:43:21,815   5845 [Engine      ] DEB: ZAdmin, current state:active. Processing
 work: Monitor fetcher work for ZAdmin. done = False
2023-05-12 14:43:21,828   5845 [Port        ] DEB: Dhclient processes:
root      7033  0.0  0.0  31096  7312 ?        Ss   14:46   0:00 /sbin/dhclient -4 -cf
/etc/dhcp/dhclient.conf.ztp -lf /var/lib/dhcp/dhclient.leases.ztp -sf
/etc/dhcp/dhclient-script.ztp2 brXRmgmt1
root      7052  0.0  0.0  20316  1592 ?        S    14:46   0:00 /bin/sh -c ps aux | grep
dhclient
root      7054  0.0  0.0  16248   948 ?        Sl   14:46   0:00 grep dhclient
2023-05-12 14:43:21,830   5845 [Port        ] DEB: <Dhclient count=1>: dhclient 4 is
stopped: keepIpAddress=True
2023-05-12 14:43:21,831   5845 [Engine      ] INF: MgmtDhcp4Fetcher, current state:final,
 exit code:success: state changed to final
2023-05-12 14:43:21,832   5845 [Engine      ] DEB: MgmtDhcp4Fetcher, current state:final,
 exit code:success. Processing work: [privileged] getting engine status. done = False
2023-05-12 14:43:21,834   5845 [MgmtDhcp4Fetcher] DEB: dhcp: shutdown : Entry
2023-05-12 14:43:21,933   5845 [Engine      ] DEB: MgmtDhcp4Fetcher, current state:final,
 exit code:success. Processing work: [privileged] prepare engine shutdown. done = False
2023-05-12 14:43:22,035   5845 [Engine      ] DEB: MgmtDhcp4Fetcher, current state:final,
 exit code:success. Processing work: [privileged] shutting down MgmtDhcp4Fetcher engine.
done = False
2023-05-12 14:43:22,036   5845 [Engine      ] INF: MgmtDhcp4Fetcher, current state:final,
 exit code:shutdown
2023-05-12 14:43:22,037   5845 [Engine      ] INF: MgmtDhcp4Fetcher, exit code:shutdown:
state changed to None
2023-05-12 14:43:22,038   5845 [Engine      ] DEB: MgmtDhcp4Fetcher, exit code:shutdown:
breaking engine loop after shutdown
2023-05-12 14:43:22,039   5845 [Engine      ] DEB: ZAdmin, current state:active. Processing
 work: Setup fetching. done = False
2023-05-12 14:43:22,039   5845 [Engine      ] DEB: MgmtDhcp4Fetcher, exit code:shutdown:
end of event loop
2023-05-12 14:43:22,052   5845 [Env         ] DEB: No authentication required for Mgmt
Interface
2023-05-12 14:43:22,053   5845 [Env         ] DEB: No authentication required when initiated
 using CLI
2023-05-12 14:43:22,055   5845 [Xr          ] DEB: Writing to file
/tmp/ztp2-fzwmlfid/sysdb_cfg_cmd.tmp
2023-05-12 14:43:22,181   5845 [Xr          ] DEB: No inconsistency found in config
2023-05-12 14:43:23,460   5845 [Xr          ] DEB: Applying TPA default route
2023-05-12 14:43:23,523   5845 [Xr          ] DEB: No IPv4 Address assigned to linux
management interface
2023-05-12 14:43:23,524   5845 [Xr          ] DEB: Applying IPv4 configuration
2023-05-12 14:43:23,525   5845 [Xr          ] DEB: Validating IP Address: 10.58.227.177
```

```
2023-05-12 14:43:23,527   5845 [Xr          ] DEB: Applying IPv4 gateway route configuration
2023-05-12 14:43:23,527   5845 [Xr            ] DEB: Validating DHCP server identifier IP
Address: 10.58.227.11
2023-05-12 14:43:23,528   5845 [Xr          ] DEB: Validating Gateway IP Address: 10.58.227.1
2023-05-12 14:43:23,530   5845 [Xr            ] DEB: Configuring domain name with
domain-name-server 144.254.71.184 173.38.200.100
2023-05-12 14:43:23,532   5845 [Configuration] DEB: Config file type is IOS XR config.
Replace False
2023-05-12 14:43:23,534   5845 [Configuration] DEB: Applying following config:
tpa
 vrf default
  address-family ipv4
   default-route mgmt
  address-family ipv6
   default-route mgmt

interface MgmtEth0/RP0/CPU0/0
no ipv4 address
ipv4 address 10.58.227.177 255.255.255.0
no shutdown

tpa
 vrf default
  address-family ipv4
   default-route mgmt
router static
 address-family ipv4 unicast
  0.0.0.0/0 10.58.227.1

domain name cisco.com

domain name-server 144.254.71.184

domain name-server 173.38.200.100
```

Generate Tech Support Information for ZTP

When you have a problem in the ZTP process that you cannot resolve, the resource of last resort is your 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 problem isolation and resolution process, collect the necessary data before you contact your representative.

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

Example:

```
RP/0/RP0/CPU0:R1#show tech-support ztp
Thu Jun 15 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 15 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:R1#
```

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.

IPv6 Support for ZTP

Table 1: Feature History

Feature Name	Release Information	Description
IPv6 Support for ZTP	Cisco IOS XR Release 7.10.1	From this release, the DHCP server supports Zero Touch Provisioning (ZTP) to bring up the NCS 1001 nodes with IPv6 addressing. The DHCP configuration file must be updated with the dhcp6.client id and IPv6 address-based bootfile URL. IPv6 addressing ensures efficient and secure management of the devices.

From Release 7.10.1, you can bring up the NCS 1001 nodes that are enabled with IPv6, using the DHCP configuration file updated with dhcp6.client id and IPv6 address based bootfile URL. See IPv6 configuration file.

Boot NCS 1001 Using Golden ISO

Golden ISO is a feature provided to user for building customized ISO using mini ISO, required SMUs and IOS-XR configuration.

Before the introduction of Golden ISO feature, the user must perform the following three steps, to install a new image.

Step 1 : Boot the system with mini ISO. This can be done using iPXE or USB boot.

Step 2 : Install, add, and activate all the relevant SMUs/optional packages on to NCS 1001. NCS 1001 reloads on reload of any SMUs.

Step 3 : Apply IOS-XR configuration.

Benefits of Golden ISO

- Saves installation effort and time.
- System gets ready in a single command and single boot.

Golden ISO is built using ‘gisobuild.py’script, which is available at /pkg/bin/gisobuild.py location.

Prerequisites

- The tools 'mount','rm','cp','umount','zcat','chroot','mkisofs' must be available and the user must have privilege to execute these tools.
- Python version must be 2.7
- The gisobuild.py script can be run either on a Linux machine or a NCS 1001 system. Destination must be an EXT file system. FAT32, NTFS file systems are not supported.

- If gisobuild.py is running on a NCS 1001 system, the environment variable PYTHONPATH must be set as export PYTHONPATH = /pkg/bin.
- System must have at least 3 GB to 4 GB of free disk space.
- Mini iso is mandatory.

  argument -r RPMREPO is mandatory.
- Kernel version of the system must be greater than 3.16 or the version of kernel of cisco iso.
- User must have proper permission for security rpm(k9sec-rpm) in rpm repository, otherwise security rpm would be ignored for Golden ISO creation.

Limitations

- install operation over IPv6 is not supported.

Golden ISO file is created in the following format:

platform-name-golden-x.iso-*version.label* (does not contain security package(*k9sec*.rpm) rpm)

Example: ncs1001-golden-x-7.1.1.14I-V1.iso

platform-name-goldenk9-x.iso-*version.label* (contains security package(*k9sec*.rpm) rpm)

Example: ncs1001-goldenk9-x-7.1.1.14I-V1.iso

Boot NCS 1001 using GISO

The following steps are used to boot NCS 1001 using giso image.

Step 1 : Create the giso image using the gisobuild.py script available at /pkg/bin.

Step 2 : Create the usb zip package using the following command.

/create_usb_zip ncs1001 ncs1001xxxxx-giso.iso

Step 3 : Extract and copy the content of zip package and copy in the USB to be used for boot.

Step 4 : Insert the USB in the usb port under test.

Step 5 : Reboot the system and install the new image.

Step 6 : After the system reboots, check the configuration using show running-config command, release using show version command, and the installed packages using show install active command.

Verify Boot Operation

Procedure

Step 1 After the boot operation, reload the NCS 1001.

Step 2 **show version**

Example:

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

```
Tue Jan 14 12:31:05.745 CET
Cisco IOS XR Software, Version 7.1.1
Copyright (c) 2013-2019 by Cisco Systems, Inc.
Build Information:
Built By : nkhai
Built On : Tue Jan 7 16:31:55 PST 2020
Built Host : iox-lnx-071
Workspace : /auto/iox-lnx-071-san1/prod/7.1.1/ncs1001/ws
Version : 7.1.1
Location : /opt/cisco/XR/packages/
Label : 7.1.1
cisco NCS-1001 () processor
```

Compare the displayed version with the boot image version. The versions need to be the same.

Accessing Consoles

Table 2: Feature History

Feature Name	Release	Description
Console Swap for NCS 1001	Cisco IOS XR Release 7.8.1	Console swap feature provides a quicker and simpler way to toggle between the following console sessions using a keyboard shortcut: • from XR to Admin console • from Admin to Host console • from Host to XR console

NCS 1001 has three types of consoles to interact with it. After entering the root username and password, you are logged in to the XR console. From the XR console, you can switch to the System Admin console and then to the host console using CLI commands.

The following table lists the console session available on NCS 1001.

Table 3: Console Types

Console Type	Functions
XR Console	Run the regular CLI commands to configure and manage the node.
System Admin Console	Perform all system administration and hardware management setups.
Host Console	Run the Linux commands such as pwd, scp, cp, traceroute, ls -la and so on.

From Release 7.8.1, you can enter **Ctrl+O** to swap between the console session, host session, and XR session. This keyboard shortcut enables you to toggle between consoles without using several CLI commands. However, you can still use the CLI commands to toggle between the console sessions.

The following table lists the CLI to move between consoles.

Table 4: Source and Target Console Swap for NCS 1001

Changing Console Session		Command Prompt	Keyboard Shortcut	Notes
From	To			
XR console	System Admin console	**XR console** RP/0/RP0/CPU0:ios#**admin** **System Admin console** sysadmin-vm:0_RP0#	CTRL+O **Example** **RP/0/RP0/CPU0#** Disconnecting from 'default-sdr--1' console. Continue(Y/N)? (Enter -> Y and then <enter> to confirm) Connecting to 'default-sdr--1' console Connecting to 'sysadmin' console ▒▒▒▒▒▒▒▒▒▒ System Admin Username: <enter username> Password: <enter password> **sysadmin-vm:0_RP0#**	When entering the System Admin console for the first time, you must provide the username and password to continue with the System Admin console. To return to the XR console from the System Admin console using CLI, enter **exit**.
System Admin console	Host console	**System Admin console** sysadmin-vm:0_RP0#**run** sysadmin-vm:0_RP0#**ssh 10.0.2.16** **Host console** [host:~]$	CTRL+O **Example** **sysadmin-vm:0_RP0#** Disconnecting from 'sysadmin' console. Continue(Y/N)? (Enter -> Y and then <enter> to confirm) Connecting to 'host' console host login:<username> Password:<password> **[host:/]$**	When entering the host console for the first time, you must provide the username and password to continue with the host console. To access the host console, use the internal host IP address 10.0.2.16 using SSH.

Changing Console Session		Command Prompt	Keyboard Shortcut	Notes
From	To			

Host console	XR console	**Host console**
		`[host:~]$exit`
		`logout`
		`Connection to 10.0.2.16 closed.`
		`[sysadmin-vm:0_RP0:~]$`
		`[sysadmin-vm:0_RP0:~]$exit`
		`exit`
		`sysadmin-vm:0_RP0#`
		`sysadmin-vm:0_RP0#exit`
		`Thu Oct 13 10:08:17.162 UTC+00:00`
		`RP/0/RP0/CPU0#`
		XR console
		RP/0/RP0/CPU0:ios#
			CTRL+O
			Example
			`[host:/]$`
			`Disconnecting from host console. Continue(Y/N)? (Enter -> Y and then <enter> to confirm)`
			`Connecting to 'default-sdr--1' console`
			`RP/0/RP0/CPU0#`
				When toggling from host console to XR console using CLI, you must enter **exit** to switch to System Admin console. Then in Admin console, enter **exit** twice to switch to XR console.

Configure Management Interface

To use the management interface for system management and remote communication, you must configure an IP address and subnet mask for the management ethernet interface. To communicate with devices on other networks (such as remote management stations or TFTP servers), you need to configure a default (static) route for the NCS 1001.

The range of supported MTU of management plane is 64 to 1514 bytes.

Before you begin

- Consult your network administrator or system planner to procure IP addresses and a subnet mask for the management port.
- Ensure that the management port is connected to the management network.

Procedure

Step 1 **configure**

Example:

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

Enters 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:

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

For IPv4, use the following command.

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

For IPv6, use the following command.

ipv6 address *ipv6-address subnet-mask*

Example:

```
RP/0/RP0/CPU0:ios(config-if)# ipv6 address 2001:420:4491:2000::229:118/64
```

Step 4 **no shutdown**

Example:

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

Places the 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** *0.0.0.0/0default-gateway*

Example:

Specifies the IP address of the default-gateway to configure a static route; this is to be used for communications with devices on other networks.

For IPv4, use the following command

Example:

```
RP/0/RP0/CPU0:ios(config)# router static address-family ipv4 unicast 0.0.0.0/0 12.25.0.1
```

For IPv6, use the following command.

Example:

```
RP/0/RP0/CPU0:ios(config)# router static address-family ipv6 unicast ::/0
2001:420:4491:2000::228:2
```

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.

What to do next

Configure Telnet, on page 24 and Configure SSH, on page 24.

Configure Telnet

With a terminal emulation program, establish a telnet session to the management interface port using its IP address.

Procedure

Step 1 **configure**

Example:

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

Enters the Configuration mode.

Step 2 **telnet {ipv4 | ipv6} server max-servers** *limit*

Example:

```
RP/0/RP0/CPU0:ios(config)# telnet ipv4 server max-servers 10
```

Specifies the number of allowable Telnet servers. Up to 100 Telnet servers are allowed. By default, no Telnet servers are allowed. You must configure this command to enable the use of Telnet servers.

Step 3 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.

What to do next

Configure SSH, on page 24

Configure SSH

With a terminal emulation program, establish a SSH connection to the management interface port using its IP address.

Before you begin

- Install the ncs1001-k9sec package on the NCS 1001. For details about package installation, see Install Packages.
- Generate the crypto key for SSH using the **crypto key generate dsa** command.

Procedure

Step 1 **configure**

Example:

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

Enters the Configuration mode.

Step 2 **ssh server v2**

Example:

```
RP/0/RP0/CPU0:ios(config)# ssh server v2
```

Enables the SSH server to accept only SSHv2 client connections.

Step 3 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.

Step 4 **show ssh session details**

Example:

```
RP/0/RP0/CPU0:ios# show ssh session details
```

Displays a detailed report of the SSHv2 connections to and from NCS 1001.

What to do next

Perform Clock Synchronization with NTP Server, on page 25

Perform Clock Synchronization with NTP Server

There are independent system clocks for the XR and the System Admin. To ensure that these clocks do not deviate from true time, they need to be synchronized with the clock of a NTP server. In this task you will configure a NTP server for the XR. After the XR clock is synchronized, the System Admin clock automatically synchronizes with the XR clock.

Before you begin

Configure and connect to the management port.

Procedure

Step 1 **configure**

Example:

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

Enters XR Configuration mode.

Step 2 **ntp server** *server_address*

Example:

```
RP/0/RP0/CPU0:ios# ntp server 64.90.182.55
```

The XR clock is configured to be synchronized with the specified sever.