Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide, Release 10.6(x)

Cisco Nexus 9000 Series Switches
Updated August 23, 2026

© 2025-2026 Cisco Systems, Inc. All rights reserved.

Trademarks

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS REFERENCED IN THIS DOCUMENTATION ARE SUBJECT TO CHANGE WITHOUT NOTICE. EXCEPT AS MAY OTHERWISE BE AGREED BY CISCO IN WRITING, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENTATION ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

The Cisco End User License Agreement and any supplemental license terms govern your use of any Cisco software, including this product documentation, and are located at: https://www.cisco.com/c/en/us/about/legal/cloud-and-software/software-terms.html. Cisco product warranty information is available at https://www.cisco.com/c/en/us/products/warranty-listing.html. US Federal Communications Commission Notices are found here https://www.cisco.com/c/en/us/products/us-fcc-notice.html.

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

Any products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability for delay in the delivery or failure to deliver any products or feature roadmap items that may be set forth in this document.

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

The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

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

Topics included

1 What's Changed in This Book.....5

What's changed in this book.....6

2 Upgrade or Downgrade the N9000 Series NX-OS Software.....7

Software image.....8
Upgrade paths.....10
NX-OS software upgrade.....11
Prerequisites for NX-OS software upgrade.....13
Nexus health and configuration checks.....14
NX-OS software upgrade guidelines.....14
MACsec non-disruptive ISSU.....18
ISSU platform support.....20
ISSU for N9300-FX platform switches.....21
ISSU for N9300-GX platform switches.....25
ISSU for N9300-HX platform switches.....26
ISSU for N9300-SG2 platform switches.....26
ISSU for N9400 platform switches.....27
Upgrade the NX-OS software.....27
In-Service Software Upgrade.....32
NX-OS upgrade history.....34
Examples of upgrade types.....35
Prerequisites for NX-OS software downgrade.....62
NX-OS software downgrade guidelines.....63
Downgrade to an earlier software release.....65

3 Transceiver Firmware.....69

Transceiver firmware software.....70
Transceiver firmware software upgrades.....70
Firmware upgrade using NX-OS commands.....70
Parallel firmware download queues and port mapping.....71
Firmware upgrade phases.....71
Firmware upgrade procedures.....72
Guidelines and limitations for firmware upgrade using NX-OS commands.....73
Upgrade firmware using NX-OS commands.....73
Examples for firmware upgrade.....82
Parallel firmware download.....94
Firmware download guidelines and limitations.....94
Copy Transceiver Firmware to Switch and Line Card.....95

Download, Run, and Commit Transceiver Firmware....96

4 Optionality in NX-OS Software....99

Optionality in NX-OS software....100
Modular packages....101
NX-OS image boot modes....102
Red Hat Package Managers....102
Format of the RPM....103
Optional RPMs and their associated features....104
Guidelines for NX-OS feature RPM installation....105
Guidelines for third-party RPM installation....107
Install command options for feature and third-party RPMs....107
Use install commands for digital signature support....110
Query all installed RPMs....110
Install RPMs using the one-step procedure....111
Install RPMs using the two-step procedure....112
Upgrade the RPMs....113
Downgrade RPMs....114
Uninstall the RPMs....115
Remove the RPMs....116
Dandified YUM commands....116
Package operations with DNF commands....116
Mapping of NX-OS commands to the DNF commands....131
Configure an FTP server and set up a local FTP YUM repository....132
Create an FTP server on Red Hat Enterprise Linux 7 (RHEL7) Virtual Machine....133
Create a local FTP YUM repository....134
Configure a switch to reach an FTP server....135
Create user roles for install operation....137

5 Convert from NX-OS to ACI Boot Mode and from ACI back to NX-OS Boot Mode 139

Convert a N9000 Series switch from NX-OS to ACI boot mode....140
Convert a replacement standby supervisor to ACI boot mode....142
Convert a N9000 Series switch back to NX-OS....143
Load NX-OS image into bootflash using SCP on the ACI shell....147

6 Migrate Switches in a vPC Topology....149

Upgrade process for a vPC topology on the primary switch....150
Upgrade process for a vPC topology on the secondary switch....151
vPC forklift upgrade....151
vPC upgrade and downgrade procedure for N9000 -R series switches....151

1 What's Changed in This Book

Topics:

- What's changed in this book

This chapter includes the new and changed features for the N9000 Series NX-OS Software Upgrade and Downgrade Guide, Release 10.6(x).

What's changed in this book

This section lists the new features for 10.6(x) releases and provides links to the sections that were updated for those features.

This table summarizes the new and changed features in this guide for Release 10.6(x) and provides links to the sections where they are documented.

Table 1: New and changed features for NX-OS Release 10.6(x)

Feature	Description	Changed in Release	Where Documented
Software image enhancement to support DPU security mode on N9300 Smart Switches	Enhanced the s1-dpu software image for N9300 Smart Switches to support DPU security mode	10.6(3s)F	Software image on page 8
Min-disruptive ISSU support on N9364E-SG2 switches	Added support for minimally disruptive ISSU on N9364E-SG2 switches	10.6(3)F	NX-OS software upgrade on page 11 NX-OS software upgrade guidelines on page 14 Upgrade the NX-OS software on page 27 Examples of upgrade types on page 35
N9164E-NS4-O software image	Added a new software image for N9164E-NS4-O switch	10.6(2n)	Software image on page 8 NX-OS software upgrade guidelines on page 14
MACsec ND ISSU	Added support for ND ISSU on MACsec-enabled interfaces on N9300-GX2, H2R, and H1 switches	10.6(2)F	NX-OS software upgrade guidelines on page 14 MACsec non-disruptive ISSU on page 18
Transceiver firmware upgrade using NX-OS commands	Added support for transceiver firmware upgrade using the NX-OS command framework on N9364E-SG2-Q and N9364E-SG2-O switches	10.6(2)F	Transceiver firmware software on page 70
Software image for Smart switches	Introduced a new software image for N9324C-SE1U and N9348Y2C6D-SE1U Smart switches	10.6(2)F	Software image on page 8
NA	No new features added for this release	10.6(1)F	NA

2 Upgrade or Downgrade the N9000 Series NX-OS Software

Topics:

- Software image
- Upgrade paths
- NX-OS software upgrade
- Prerequisites for NX-OS software upgrade
- NX-OS software upgrade guidelines
- ISSU platform support
- Upgrade the NX-OS software
- In-Service Software Upgrade
- NX-OS upgrade history
- Examples of upgrade types
- Prerequisites for NX-OS software downgrade
- NX-OS software downgrade guidelines
- Downgrade to an earlier software release

Explains how to manage software upgrades and downgrades for N9000 Series switches, including the use of standard and enhanced In-Service Software Upgrades (ISSU) to minimize network downtime.

Software image

This section provides details about the NX-OS software image, which consolidates all necessary system components and outlines the support for disruptive upgrades, SMU package files, and EPLD handling across various Nexus 9000 Series switches.

The NX-OS software image is an image that

- consolidates the necessary system components into one image,
- includes a bundled EPLD image beginning with Release 10.5(3)F, and
- uses a 64-bit format.

Each Nexus switch is shipped with the NX-OS software preinstalled. The NX-OS software consists of one NX-OS software image, and this image is required to load the NX-OS operating system.

> **Note**
>
> Beginning with Release 10.5(3)F, NX-OS no longer provides a separate EPLD image. The EPLD image is bundled with all NX-OS images, increasing image sizes. For more information, refer to Cisco Nexus 9000 Series FPGA/EPLD Upgrade Release Notes.

NX-OS Release 10.6(x) supports only 64-bit images, which are listed in this table.

Table 2: Software image

Image prefix	Example	Remarks
`nxos64-cs`	`nxos64-cs.10.5.3.F.bin`	Supported on all Nexus 9000 series switches except Nexus 9500 -R and -R2 switches and line cards, Nexus 9800 switches, and N9364E-SG2 switches
`nxos64-msll`	`nxos64-msll.10.5.3.F.bin`	Supported only on Nexus 9500 Series -R and -R2 modular switches
`nxos64-s1`	`nxos64-s1.10.5.3.bin`	Mandatory on Nexus 9800 and N9364E-SG2 switches. This image is supported from NX-OS Release 10.5(1)F on Nexus 9800 switches and, from 10.5(3)F on N9364E-SG2 switches.

Image prefix	Example	Remarks
nxos64-s1-dpu	nxos64-s1-dpu.10.6.2.F.bin nxos64-s1-dpu.10.6.3s.F.bin	Mandatory on N9324C-SE1U and N9348Y2C6D-SE1U Smart switches. Minimum software version for the N9300 Smart Switches is NX-OS Release 10.6(2)F. Beginning with NX-OS 10.6(2)F, only network mode is supported. Beginning with NX-OS Release 10.6(3s)F, these switches support two modes: network and DPU security mode. In the network mode, only the networking features are supported. In the DPU security mode, both network and DPU security features are supported. **Note** • Only disruptive upgrade is supported on the Smart switches. For more information about this image, refer to DPU image.
nxos64-msx	nxos64-msx.10.6.2n.F.bin	Mandatory on N9164E-NS4-O switch from NX-OS Release 10.6(2n). Currently, only disruptive upgrade is supported.

The Nexus 9000 Series switches support disruptive software upgrades and downgrades by default.

For information about the supported upgrade paths, see the CiscoNexus 9000 and 3000 Upgrade and ISSU Matrix.

Note

- Until NX-OS Release 10.5(2)F, we provided separate electronic programmable logic device (EPLD) image upgrades to enhance hardware functionality or to resolve known hardware issues. For more information on the EPLD image and the upgrade process, see the Cisco Nexus 9000 Series FPGA/EPLD Upgrade Release Notes.
- Until NX-OS Release 10.5(2)F, ISSU supports EPLD upgrades using the **install all nxos** *<nxos-image>* **epld** *<epld-image>* command during disruptive system (NX-OS) upgrade.
- While upgrading from pre-10.5(3)F releases to 10.5(3)F and later, first upgrade to 10.5(3) NX-OS using the **install all** *<nxos-image>* command. Then, after the NX-OS upgrade is complete, upgrade EPLD using the **install epld** command.
- Beginning with NX-OS Release 10.5(3)F, EPLD upgrade takes place during an ISSU system upgrade. To avoid EPLD upgrade, use the **skip-epld** option. Do not use the **epld** *<epld-image>* option as the EPLD image is bundled with the NX-OS images and a separate EPLD image is no longer provided.

DPU image

Beginning with NX-OS Release 10.6(3s)F, the DPU image (`nxos64-s1-dpu`) includes both the NX-OS and the DPU images for N9300 Smart Switches. With this image the switch functions as an NX-OS switch in network mode by default (feature service-acceleration is disabled by default). If you enable feature service-acceleration, it powers up the DPUs, and the switch can provide both network and DPU security mode features.

Upgrade Smart Switches

As Smart Switches support only disruptive upgrade, follow these instructions to perform a software upgrade from earlier versions to 10.6(3s)F:

1. Start the upgrade using the **install all nxos** *image-name* command.
2. Apply the service acceleration configuration. For configuration details, refer to the Cisco Smart Switches Configuration Guide for DPU Security Mode.

For more information, refer to the Upgrade the NX-OS software section.

SMU image

Another type of binary file is the software maintenance upgrade (SMU) package file. SMUs contain fixes for specific defects. They are created to respond to immediate issues and do not include new features. SMU package files are available for download and generally include the ID number of the resolved defect in the filename (for example, n9000-dk10.1.1.CSCab00001.gbin). For more information on SMUs, see the Cisco N9000 Series NX-OS System Management Configuration Guide.

Upgrade paths

This section provides information about the upgrade paths for Nexus 9000 Series devices and outlines the supported transitions between major and maintenance software releases.

For a list of specific releases from which you can perform a disruptive upgrade or a non-disruptive ISSU, see the Cisco Nexus 9000 Series NX-OS Release Notes for your particular release.

For ISSU compatibility for all releases and information about the upgrade paths, see the Cisco Nexus 9000 and 3000 Upgrade and ISSU Matrix.

In general, ISSU is supported from

- a major release to any associated maintenance release
- the last two maintenance releases to the next two major releases, and
- an earlier maintenance release to the next two major releases.

NX-OS software upgrade

Outlines the process and benefits of upgrading NX-OS software on Nexus switches, detailing disruptive, non-disruptive, and min-disruptive ISSU methods, their prerequisites, procedures, platform-specific considerations, and impact on network stability and downtime.

A software upgrade on a switch involves updating the operating system (OS) or firmware running on the switch and is performed to provide:

- Security patching: Addressing known vulnerabilities to protect the network from potential exploits.
- Bug fixes: Resolving software-related issues that may cause instability or unexpected behavior.
- Feature enhancement: Enabling new software-defined networking (SDN) capabilities, improved management protocols, or better traffic optimization features.

Upgrading a Nexus switch running NX-OS is a critical operation that requires careful planning to ensure network stability. The process typically involves, downloading the software image, verifying the compatibility between the current version and the target version, checking for hardware requirements, and utilizing the install all command, which automates the upgrade process and performs compatibility checks.

The two types of NX-OS software upgrade are:

- Disruptive upgrade
- Non-disruptive In-Service Software Upgrade (ND ISSU)

Disruptive upgrade

A disruptive upgrade is the default upgrade process that requires a reload or cold reboot of the Nexus device and any connected Fabric Extenders (FEX). This reload disrupts both the control plane and the data plane, causing downtime and traffic interruption.

Disruptive upgrades are necessary when non-disruptive upgrade prerequisites are not met, or when upgrading between incompatible images, which may result in configuration loss, for example, unified ports, breakout, and FEX configurations.

Disruptive upgrade is supported on all NX-OS platforms but causes downtime during the upgrade. For example, in Nexus 9000 Series with single supervisors, disruptive upgrade causes the supervisor CPU to reset and reload the new software, disrupting control plane and data plane during the process.

The reload causes disruptions to connected servers and hosts, but in vPC deployments, traffic is minimized by routing through the peer switch during upgrade.

Use the **install all nxos bootflash:**<filename> command for disruptive upgrade.

Example

```
install all nxos bootflash:nxos64-cs.10.6.2.F.bin
```

For a sample display for disruptive upgrade, refer to the Disruptive ISSU for CloudScale TOR and Silicon One EOR sections.

Non-disruptive In-Service Software Upgrade

Non-disruptive In-Service Software Upgrade (ND ISSU) allows upgrading the device software while the switch continues to forward traffic, minimizing or eliminating downtime.

Non-disruptive ISSU requires meeting specific prerequisites, such as compatible software releases, supported hardware platforms, and ISSU-ready topology and features. Few features and configurations are not supported during non-disruptive ISSU, and certain platforms or configurations only support disruptive ISSU.

Non-disruptive ISSU requires careful configuration, such as setting appropriate BGP graceful restart timers and avoiding configuration changes during the upgrade.

Non-disruptive ISSU is further classified as:

- Non-Disruptive ISSU for Cloud Scale switches
  - Enhanced ISSU (default)

    Enhanced ISSU uses Linux containers (LXC) to run virtual instances of supervisor modules and line cards, enabling the upgrade with minimal impact. With Enhanced ISSU, the software runs inside a separate Linux container for the supervisor and line card. On the TOR, two additional containers are created and brought up as the standby supervisor and standby line card respectively.

    The virtual instances or linus containers communicate with each other using an emulated Ethernet connection. In the normal state, only two Linux containers are instantiated: vSup1, a virtual SUP container in an active role, and vLC, a virtual linecard container.

    Use the **install all nxos bootflash:** *<filename>* **non-disruptive** command for disruptive upgrade.

    Example:

    ```
    install all nxos bootflash:nxos64-cs.10.6.2.F.bin non-disruptive.
    ```

    > **Note**
    >
    > For switches in LXC boot mode, the enhanced LXC upgrade falls back to standard ND ISSU because the target image kernels are likely different from the current image. In standard ISSU, the CPU reset domain uses a kexec system call to boot into the new image after NX-OS components save their state. Kexec allows a new Linux kernel-based image to be loaded and booted from the currently running kernel.

    For samples of display for Non-disruptive ISSU, refer to the Non-Disruptive ISSU for Cloudscale TOR section.

- Min-disruptive ISSU for Silicon One switches
  - Standard ISSU only

    Beginning with NX-OS Release 10.6(3)F, min-disruptive (MD) ISSU, or minimally disruptive ISSU, is introduced only on Nexus 9364E-SG2 Silicon One-based TORs. While performing MD ISSU, any minimal packet disruption is displayed to the user.

    The drop-time information is also shown in the output of the `show install all min-disruptive drop-time` command.

    ```
    switch# show install all min-disruptive drop-time
    Estimated traffic outage time for current upgrade : about 65 ms
    ```

    While performing a min-disruptive ISSU, increase the routing protocol graceful restart timer to a higher value. The recommended value for graceful restart timer is 240 seconds, although this value can be adjusted based on the configuration scale.

If network topology changes during MD ISSU cause few SUP apps not to proceed with the upgrade, then MD ISSU is aborted and the upgrade becomes disruptive. Furthermore, if any SUP app responds with failure, MD ISSU is aborted and the upgrade becomes disruptive.

The output of the **show install all status** command now shows min-disruptive.

Use the **install all nxos bootflash:** *<filename>* **min-disruptive** command for disruptive upgrade.

Example

```
install all nxos bootflash:nxos64-s1.10.6.2.F.bin min-disruptive.
```

For a sample of display for min-disruptive ISSU, refer to the Min-Disruptive ISSU on Silicon One TOR section.

Prerequisites for NX-OS software upgrade

Use the information in this section to ensure that you verify the upgrade paths, there are no active configuration sessions or user configurations during the upgrade, confirm connectivity to the remote file server for image transfer, and validate spanning tree criteria for ISSU to enable a smooth NX-OS software upgrade.

The prerequisites that you must meet before upgrading the NX-OS software are:

- Verify the recommended upgrade paths between releases are specified in the Cisco Nexus 9000 and 3000 Upgrade and ISSU Matrix.
- Ensure that no user–who has access to the device or the network–is configuring the device or the network during the upgrade. You cannot configure a device during an upgrade. To verify that you have no active configuration sessions, use the **show configuration session summary** command.
- Save, commit, or discard any active configuration sessions before upgrading or downgrading the NX-OS software image on your device. While upgrading NX-OS software on a device with dual supervisors, the active supervisor module cannot switch over to the standby supervisor module if there is an active configuration session.
- To transfer NX-OS software images to the Nexus switch through a file transfer protocol such as TFTP, FTP, SFTP, and SCP, verify that the Nexus switch can connect to the remote file server where the NX-OS software images are stored. If you do not have a router to route traffic between sub nets, ensure that the Nexus switch and the remote file server are on the same sub network. To verify connectivity to the remote server, transfer a test file using a file transfer protocol of your choice or use the ping command if the remote file server is configured to respond to ICMP Echo Request packets. Here is an example of using the **ping** command to verify connectivity to a remote file server `192.0.2.100`.

```
switch# ping 192.0.2.100 vrf management
PING 192.0.2.100 (192.0.2.100): 56 data bytes
64 bytes from 192.0.2.100: icmp_seq=0 ttl=239 time=106.647 ms
64 bytes from 192.0.2.100: icmp_seq=1 ttl=239 time=76.807 ms
64 bytes from 192.0.2.100: icmp_seq=2 ttl=239 time=76.593 ms
64 bytes from 192.0.2.100: icmp_seq=3 ttl=239 time=81.679 ms
64 bytes from 192.0.2.100: icmp_seq=4 ttl=239 time=76.5 ms

--- 192.0.2.100 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 76.5/83.645/106.647 ms
```

- For non-disruptive ISSU in spanning tree topology, before running the **show spanning-tree issu-impact** command, the criteria that you must verify include
  - no topology change must be active in any STP instance
  - Bridge Assurance (BA) should not be active on any port (except MCT and vPC peer link)

- there should be no Non-Edge Designated Forwarding port (except MCT and vPC peer link), and
- ISSU criteria must be met on the vPC peer switch.

A pre-upgrade generic checklist includes:

- Schedule the upgrade when your network is stable and steady.
- Avoid any power interruption, which could corrupt the software image, during the installation procedure.
- On devices with dual supervisor modules, both supervisor modules must have connections on the consoleports to maintain connectivity when switchovers occur during a software upgrade. For more information about your specific chassis, refer to the relevant Hardware Installation Guide.
- Perform the installation on the active supervisor module, not the standby supervisor module.

> **Note**
>
> For more information about configuration sessions, refer to the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide specific to your release.

Nexus health and configuration checks

This section recommends that the health and configuration check must be performed on the switch before performing an upgrade to avoid potential issues.

A Nexus health and configuration check is an automated diagnostic service that

- analyzes Nexus switch logs to identify issues and provides recommendations
- checks critical configurations such as virtual Port Channel (vPC), multicast, and Layer 3 uplinks, and
- ensures best practices and detects inconsistencies in complex environments.

We recommend performing a Nexus health and configuration check before performing an upgrade. The benefits include identification of potential issues, susceptible Field Notices, security vulnerabilities, and missing recommended configurations. For more information about the procedure, see Perform Nexus Health and Configuration Check.

NX-OS software upgrade guidelines

This section lists comprehensive guidelines, requirements, and special considerations for performing NX-OS software upgrades on Nexus 9000 series switches, including instructions such as generic, release-specific, and feature-specific.

Before attempting to upgrade to any software image, follow the guidelines and limitations listed under these sub sections to ensure compatibility, minimize disruptions, and maintain operational stability.

> **Note**
>
> For ISSU compatibility for all releases, see the Cisco Nexus 9000 and 3000 Upgrade and ISSU Matrix.

- Generic

- Software image and SMU
- EPLD
- Release specific
- Switch specific
- Disruptive and non-disruptive ISSU
- Feature specific
- Unsupported PIDs

Generic

The guidelines that apply to all upgrades irrespective of the releases are

- When you use **install all** with **no-reload** option, the saved configuration cannot be used before you reload the device. Saving configuration in this state can result in incorrect startup configuration once you reload the device with new version of NX-OS.
- During upgrade, while performing device reload, if ASCII replay is triggered without binary restore, primary key gets lost. The primary key must be reconfigured after device reload. Use the **key config-key ascii** command to reconfigure the primary key and avoid encryption issues. However, upgrade with binary restore retains the primary key after the reboot.
- ISSU is blocked if **boot poap enable** is configured.
- Make sure that both vPC peers are in the same mode (regular mode or enhanced mode) before performing a non-disruptive upgrade.

  Note

  vPC peering between an enhanced ISSU mode (boot mode lxc) configured switch and a non-enhanced ISSU mode switch is not supported.

- During an ISSU, the software reload process on the first vPC device locks its vPC peer device by using CFS messaging over the vPC communications channel. Only one device at a time is upgraded. When the first device completes its upgrade, it unlocks its peer device. The second device then performs the upgrade process, locking the first device as it does so. During the upgrade, the two vPC devices temporarily run different releases of NX-OS; however, the system functions correctly because of its backward compatibility support.
- ISSU is not supported when onePK is enabled. You can run the **show feature | include onep** command to verify that this feature is disabled before performing an ISSU or enhanced ISSU.
- Occasionally, while the switch is operationally up and running, the Device not found logs are displayed on the console. This issue is observed because the switch attempts to find an older ASIC version and the error messages for the PCI probe failure are enabled in the code. There is no functionality impact or traffic loss due to this issue.
- For secure POAP, ensure that DHCP snooping is enabled and set firewall rules to block unintended or malicious DHCP servers. For more information on POAP, see the Cisco Nexus 9000 Series Fundamentals Configuration Guide.
- When you upgrade from an earlier release to a NX-OS release that supports switch profiles, you have the option to move some of the running-configuration commands to a switch profile. For more information on configuration, see the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide.

- Guest Shell is disabled during an ISSU and reactivated after the upgrade. All applications running in the Guest Shell are affected.
- While performing an ISSU, VRRP and VRRPv3 display these messages:
  - If VRRPv3 is enabled:

```
2015 Dec 29 20:41:44 MDP-N9K-6 %$ VDC-1 %$ %USER-0-SYSTEM_MSG: ISSU ERROR:
 Service
"vrrpv3" has sent the following message: Feature vrrpv3 is configured. User
 can change
vrrpv3 timers to 120 seconds or fine tune these timers based on upgrade
time on all Vrrp
Peers to avoid Vrrp State transitions. – sysmgr
```

  - If VRRP is enabled:

```
2015 Dec 29 20:45:10 MDP-N9K-6 %$ VDC-1 %$ %USER-0-SYSTEM_MSG: ISSU ERROR:
 Service "vrrp-
eng" has sent the following message: Feature vrrp is configured. User can
 change vrrp
timers to 120 seconds or fine tune these timers based on upgrade time on
all Vrrp Peers to
avoid Vrrp State transitions. – sysmgr
```

- An error occurs when you try to perform an ISSU if you changed the reserved VLAN without entering the copy running-config save-config and reload commands.

Software image and SMU

- Beginning with NX-OS Release 10.6(2n), `-msx` image (for example, `nxos64-msx.10.6.2n.F.bin`) is introduced for N9164E-NS4-O switch. Currently, only disruptive upgrade to any future releases is supported on this switch.
- Beginning with NX-OS Release 10.5(1)F, s1 image is introduced specifically for Nexus 9800 switches.

  Upgrade of Nexus 9800 switches from earlier releases that have cs image format to the s1 image format in NX-OS Release 10.5(1)F and later is supported.
- Loading an unsupported image on Nexus 9800 platform switches cause the switch to be stuck. Only a power cycle can reset it.
- The **install all** command is the recommended method for software upgrades because it performs configuration compatibility checks and BIOS upgrades automatically. In contrast, changing the boot variables and reloading the device bypasses these checks and the BIOS upgrade and therefore is not recommended.
- You can detect an incomplete or corrupt NX-OS software image prior to performing an upgrade by verifying the MD5, SHA256 or SHA512 checksum of the software image. To verify the MD5 checksum of the software image, run the **show file bootflash:** <*IMAGE-NAME*>**md5sum** command and compare the resulting value to the published MD5 checksum for the software image on the *Software Download* website. To verify the SHA512 checksum of the software image, run the **show file bootflash:** <*IMAGE-NAME*>**sha512sum** command and compare the resulting value to the published SHA512 checksum for the software image on the *Software Download* website.
- The **install all** command is the recommended method for software upgrades because it performs configuration compatibility checks and BIOS upgrades automatically. In contrast, changing the boot variables and reloading the device bypasses these checks and the BIOS upgrade and therefore it is not recommended.

EPLD

- From NX-OS Release 10.6(1)F, while installing nx-os using the **install all nx-os** command on switches affected by secure boot vulnerability, if the IO FPGA version of the device is lower than the Fixed IO FPGA version, EPLD upgrade does not take place. To upgrade the FPGA, use the **install epld** command. For more information about switches affected by secure boot vulnerability and Fixed IO FPGA version, refer to *Table 1* in the FPGA/EPLD Upgrade Procedure to Address Secure Boot Vulnerability document.
- From NX-OS Release 10.5(3)F, while installing nx-os using the **install all nx-os** command on switches affected by secure boot vulnerability, EPLD upgrade does not take place. To upgrade the FPGA, use the **install epld** command. For more information about switches affected by secure boot vulnerability and Fixed IO FPGA version, refer to *Table 1* in the FPGA/EPLD Upgrade Procedure to Address Secure Boot Vulnerability document.
- ISSU supports EPLD image upgrades using **install all nxos** *<nxos-image>* **epld** *<epld-image>* command, during disruptive system (NX-OS) upgrade. Beginning with NX-OS Release 10.5(3)F, do not use the **epld** *<epld_image>* option as the EPLD image is bundled with the NXOS images and a separate EPLD image is no longer provided.

Release specific

- Upgrade from release 10.4(6)F to 10.5(1)F, 10.5(2)F, or 10.5(3)F releases is not supported and can result in configuration loss or its corruption. To upgrade from 10.4(6)M or 10.4(7)M release to 10.6(x) releases, the recommended path is to first upgrade to 10.5(4)M and then to 10.6(x). See CSCwr21007 in the Cisco Nexus 9000 Series NX-OS Release Notes, Release 10.4(6)M and Release 10.4(7)M.
- ISSU is blocked when the delay configuration is present in track list Boolean/weight.
- If the IPv6 ND timeouts during ISSU, then the IPv6 BFD session may flap after the ISSU.

Switch specific

- During an ISSU on a Nexus 9300 Series switch, all First-Hop Redundancy Protocols (FHRPs) will cause the other peer to become active if the node undergoing the ISSU is active.
- Beginning with NX-OS Release 10.6(2)F, Nexus 9300-GX2, H2R, and H1 series switches support non-disruptive ISSU on switches that have MACsec-enabled interfaces.
- While performing non-disruptive ISSU from NX-OS Release 10.4(6)M to 10.6(1)F and later releases, on Nexus 9300-FX switches and line cards, IGMP traffic is forwarded on vPC legs towards the vPC pair. When there are multiple FEX devices on the vPC peer undergoing ISSU, multicast traffic loss can occur during the upgrade of the FEX devices. To resolve this, configure the **ip igmp group-timeout 450** command on all VLANs that carry IGMP traffic across the vPC peer link.
- Non-disruptive ISSU is not supported on interfaces with 2.5G or 5G speed on N9K-C93108TC-FX3P platform. For more information, refer to CSCwq38959.
- ISSU with with FCoE (Fiber Channel over Ethernet)/FC (Fiber Channel) NPV (N-port Virtualization) is supported on some Nexus 9000 switches. An ISSU allows you to upgrade the device software while the switch continues to forward traffic. You can perform an in-service software upgrade (ISSU), also known as a nondisruptive upgrade, for some Nexus 9000 switches. The default upgrade process is disruptive. Using the nondisruptive option helps ensure a nondisruptive upgrade.

  Fibre Channel N-port Virtualization (NPV) can co-exist with VXLAN on different fabric uplinks but on same or different front panel ports on the Nexus 93180YC-FX, N9K-C9336C-FX2-E, and N9k-C93360YC-FX2 switches.

Disruptive and non-disruptive ISSU

- When upgrading Nexus 9300-FX2 switch from NX-OS Release 10.5(3)F to any later releases, only disruptive upgrade is supported, and ND ISSU is not supported when the system is enabled with routing template security group. However, ND ISSU is supported from NX-OS Release 10.6(1)F.

- While performing ND ISSU, if a router is configured with BGP prefix peers, prefix-peer-timeout (default value - 30s) should be greater than GR timer (default value - 120s), to allow the prefix peers to resume the connection after ISSU.
- The recommended routing protocol graceful restart timer is 240 seconds and nve source-interface hold-down-time is 400 seconds. For higher scale deployments adjust the graceful restart timer as required.
- It is recommended to set **disable-fka** on VFC interfaces in E or F mode, when invoking ND native ISSU on switch mode testbed. If not, it can be disruptive.
- Beginning from NX-OS Release 10.2(8)M onwards, Nexus 9300-FX3 supports non-disruptive upgrade.
- When performing ND ISSU using BGP non-default hold timers, ensure that the BGP graceful-restart timer is at least 180 seconds or higher.
- If there is a VRF scale, for a non-disruptive ISSU under each VRF, you must configure graceful restart timer to 300 seconds.
- OpenFlow and LACP fast timer rate configurations are not supported for Non-Disruptive ISSU.

Feature specific

- From NX-OS Release 10.6(1)F, on Nexus modular switches, if the Backplane diagnostic test fails and a BACKPLANE_AUTHENTICATION_FAIL syslog appears, do not perform an upgrade or a system reload.
- While upgrading from an earlier release to 10.5(3)F or later releases, as a part of sFlow ISSU Consistency Checker, pre and post configuration files are created in the booflash. To remove the snapshot files, use the **clear system internal sflow consistency pss-snapshot** command. However, if the snapshot files are removed, the **show system internal sflow consistency issu-pss** command does not provide the expected output. For more information, refer to Cisco Nexus 9000 Series NX-OS System Management Configuration Guide.
- Enhanced ISSU is not supported with IPFM.

Unsupported PIDs

The table displays the list of unsupported PIDs from various NX-OS Releases.

Unsupported PIDs	NX-OS Release
N9332C and N9364C	10.6(1)F
N9K-C92348GC-X	10.6(1)F
9700-EX line cards	10.6(1)F

Note

N9K-X97160YC-EX line card is supported.

MACsec non-disruptive ISSU

MACsec non-disruptive ISSU allows software upgrades on MACsec-enabled interfaces without disrupting encrypted traffic, provided specific conditions are met.

Media Access Control Security (MACsec) non-disruptive ISSU is a feature that

- allows in-service software upgrades on switches with MACsec-enabled interfaces
- ensuring continued encryption and authentication of Ethernet traffic during the upgrade process, and
- is based on IEEE 802.1X-2014 REV of the MACsec Key Agreement (MKA) protocol. This protocol makes provision for MACsec-enabled switches to support ISSU by incorporating an additional parameter set.

MACsec is a Layer 2 security feature that provides encryption and authentication for traffic traversing Ethernet links. Nexus 9300-GX2, H2R, and H1 switches supports non-disruptive ISSU on switches that have MACsec-enabled interfaces.

Note

Non-Disruptive ISSU is only supported on MACsec PSK-enabled interface.

To perform a non-disruptive ISSU on MACsec-enabled interfaces, the conditions that must be met include:

- The MACsec interfaces must use XPN cipher suites; otherwise, the upgrade is aborted.
- Both peers must run supported software versions with MKA version 2 or above. If any switch runs an older software version with MKA REV1, ISSU is aborted.

Note

After completing ISSU, MACsec control plane transitions from the suspended to the secured state.

During the upgrade process, the MACsec interface can enter a new state called `suspended`. When an interface is in suspended state, it continues to encrypt outgoing data packets and decrypt incoming data packets on the interface. However, control packets may not be delivered sometimes when the interface control plane has a downtime. Due to this, a suspended interface does not support:

- Rekey results from policy timer expiry or packet number exhaustion. Packet number exhaustion-based rekey should not be triggered when using XPN cipher suites. XPN cipher suites provide a much wider window in which rekeying should occur.
- MACsec configuration changes after the interface enters a suspended state.

Note

Any key expiry triggered after an interface moves to suspended state is handled only when the interface exits suspension and moves back to secured state.

Two new commands are introduced at the MACsec policy level to support this feature. For more information, refer to the *Configuring a MACsec policy* section in Cisco Nexus 9000 Series NX-OS Security Configuration Guide.

System messages for MACsec non-disruptive ISSU

A few system messages or logs are added to notify when an interface is suspended and when it exits suspension.

- ```
  switch %CTS-5-CTS_SUSPENSION_START: MACsec control plane operation suspended
   for interface: Ethernet1/18
  ```

  - This system log indicates that an interface has moved to suspended state.

- ```
  switch %CTS-5-CTS_SUSPENSION_REQUESTED: Requesting suspension of MACsec control
   plane operation for interface: Ethernet1/18
  ```

  - This system log indicates that the peer interface has requested suspension to be started.

- ```
  switch %CTS-5-CTS_SUSPENSION_STOP: MACsec control plane operation resumed for
   interface: Ethernet1/18
  ```

  - This system log indicates that the suspension has ended and the interface is back to its original state.

ISSU platform support

Explains platform support for standard and enhanced ISSU, including the specific conditions that cause enhanced upgrades to revert to full, disruptive processes.

The tables in this section summarize which Nexus platforms support standard and enhanced ISSU, the software release when support was introduced, and any features not supported for non-disruptive upgrades.
  ```

> **Note**
>
> Enhanced ISSU cannot be supported if there is kernel update in the target release without reloading the container. The system prompts the `Host kernel is not compatible with target image. Full ISSU will be performed and control plane will be impacted`. message. In effect, the system performs non-disruptive ISSU instead of enhanced ISSU.

ISSU for N9300-FX platform switches

ISSU Type	Release and Supported Platforms	Features Not Supported with Non-disruptive ISSU
Standard	NX-OS Release 9.3(1) and 9.3(2): None Beginning with NX-OS Release 9.3(3): • N9336C-FX2 • N93240YC-FX2 • N93240YC-FX2Z • N9348GC-FXP • N93108TC-FX • N93180YC-FX Beginning with NX-OS Release 10.2(8)M: • N93180YC-FX3 • N93180YC-FX3S Beginning with NX-OS Release 10.3(1)F: • N93180YC-FX3H Beginning with NX-OS Release 10.4(1)F: • 9348GC-FX3 • 9348GC-FX3PH Beginning with NX-OS Release 10.4(2)F: • 93108TC-FX3	

ISSU Type	Release and Supported Platforms	Features Not Supported with Non-disruptive ISSU
		Standard ISSU is disruptive for N9300-FX platform switches configured with features such as • Segment Routing • TRM Feature, and • MACsec **Note** Beginning with NX-OS Release 10.2(1), Standard ISSU is non-disruptive for N9300-FX platform switches configured with Straight-Through FEX and Dual-Homed FEX. Beginning with NX-OS Release 10.3(3)F, standard ISSU is not supported on N93180YC-FX3 and FX3S platform switches. Beginning with NX-OS Release 10.4(1)F, standard ISSU is not supported on N9300-FX and 9300-FX2 platform switches. **Note** Beginning with NX-OS Release 10.4(2)F, non-disruptive ISSU is supported for segment routing traffic engineering (SR-TE) features with BGP as underlay only on N9300 platform switches. For more information about the features that are not supported, see NX-OS software upgrade guidelines on page 14
Enhanced		

ISSU Type	Release and Supported Platforms	Features Not Supported with Non-disruptive ISSU
	NX-OS Release 9.3(1), 9.3(2), and 9.3(3): None Beginning with NX-OS Release 9.3(5): • N9336C-FX2 • N93240YC-FX2 • N93216TC-FX2 • N93360YC-FX2 • N93240YC-FX2Z • N9348GC-FXP • N93108TC-FX • N93180YC-FX Beginning with NX-OS Release 10.1(1), Enhanced ISSU is supported on the following platforms with FC/FCoE features: • N93360YC-FX2 • N93180YC-FX Beginning with NX-OS Release 10.2(8)M, Enhanced ISSU is supported on the following platforms: • N93180YC-FX3 • N93180YC-FX3S Beginning with NX-OS Release 10.3(1)F: • N93180YC-FX3H ` Beginning with NX-OS Release 10.4(1)F: • 9348GC-FX3 • 9348GC-FX3PH Beginning with NX-OS Release 10.4(2)F: • 93108TC-FX3 Beginning with NX-OS Release 10.2(2)F, Enhanced ISSU is supported on the following platform with FC/FCoE features: • N9K-C9336C-FX2-E	

ISSU Type	Release and Supported Platforms	Features Not Supported with Non-disruptive ISSU
		Enhanced ISSU is disruptive for N9300-FX platform switches configured with features such as • Segment Routing • TRM Feature, and • MACsec **Note** In NX-OS Release 9.3(x), Enhanced ISSU on N93360YC-FX2 and N93180YC-FX with FC/FCoE features will be disruptive. **Note** Beginning with NX-OS Release 10.2(1), Enhanced ISSU is non-disruptive for N9300-FX platform switches configured with Straight-Through FEX and Dual-Homed FEX. Beginning with NX-OS Release 10.3(3)F, only the LXC mode is supported on N9300-FX3 and 9300-FX3S switches, which allows you to perform enhanced non-disruptive ISSU with minimal downtime. Beginning with NX-OS Release 10.4(1)F, only the LXC mode is supported on N9300-FX and 9300-FX2 switches, which allows you to perform enhanced non-disruptive ISSU with minimal downtime. **Note** Beginning with NX-OS Release

ISSU Type	Release and Supported Platforms	Features Not Supported with Non-disruptive ISSU
		10.4(2)F, non-disruptive ISSU is supported for segment routing traffic engineering (SR-TE) features with BGP as underlay only on N9300 platform switches. For more information about the features that are not supported, see NX-OS software upgrade guidelines on page 14.

ISSU for N9300-GX platform switches

ISSU Type	Release and Supported Platforms	Features Not Supported with Non-disruptive ISSU
Standard	Beginning with NX-OS Release 10.2(8)M: N9364C-GX N9316D-GX N93600CD-GX	• TRM Feature • Segment Routing **Note** Beginning with NX-OS Release 10.4(2)F, non-disruptive ISSU is supported for segment routing traffic engineering (SR-TE) features with BGP as underlay only on N9300 platform switches. For more information about the features that are not supported, see NX-OS software upgrade guidelines on page 14.
Enhanced		

ISSU Type	Release and Supported Platforms	Features Not Supported with Non-disruptive ISSU
	Beginning with NX-OS Release 10.2(8)M: N9364C-GX N9316D-GX N93600CD-GX Beginning with NX-OS Release 10.2(2)F, Enhanced ISSU is supported on N9300-GX2B platform switches. Beginning with NX-OS Release 10.2(3)F, Enhanced ISSU is supported on N9300-GX2A platform switches. Beginning with NX-OS Release 10.3(3)F, only the LXC mode is supported on N9300-GX switches, which allows you to perform enhanced non-disruptive ISSU with minimal downtime.	• TRM Feature • Segment Routing **Note** Beginning with NX-OS Release 10.4(2)F, non-disruptive ISSU is supported for segment routing traffic engineering (SR-TE) features with BGP as underlay only on N9300 platform switches. For more information about the features that are not supported, see NX-OS software upgrade guidelines on page 14.

ISSU for N9300-HX platform switches

ISSU Type	Release and Supported Platforms	Features Not Supported with Non-disruptive ISSU
Enhanced	Boot mode LXC ISSU is supported by default on the following Nexus C9300-HX platform switches: • N9K-C9332D-H2R - Beginning with NX-OS Release 10.4(1)F • N9K-C93400LD-H1 - Beginning with NX-OS Release 10.4(2)F • N9K-C9364C-H1 - Beginning with NX-OS Release 10.4(3)F **Note** ND-ISSU from any 10.4(x) or 10.5(x) releases to 10.5(3)F release is not supported on N9K-C9364C-H1.	• SR L2EVPN • ISIS and OSPF underlay • vPC configuration with segment-routing • Egress Peer engineering • Segment routing and GRE co-existence • MACsec

ISSU for N9300-SG2 platform switches

Outlines ISSU support for N9300-SG2 platform switches, including supported NX-OS releases, platform compatibility, and features not supported with non-disruptive ISSU.

ISSU Type	Release and Supported Platforms	Features Not Supported with Non-disruptive ISSU
Min-disruptive	Beginning with NX-OS Release 10.6(3)F, Min-disruptive Standard ISSU is supported on N9364E-SG2-O and 9364E-SG2-Q platform switches.	• TRM Feature • Segment Routing • MACsec

ISSU for N9400 platform switches

Outlines supported ISSU types for N9400 platform switches, lists the NX-OS release and platforms with Enhanced ISSU support, and identifies features not compatible with non-disruptive ISSU.

ISSU Type	Release and Supported Platforms	Features Not Supported with Non-disruptive ISSU
Enhanced	Beginning with NX-OS Release 10.4(1)F, Enhanced ISSU is supported on Nexus C9408 platform switch.	• TRM Feature • Segment Routing • MACsec

Upgrade the NX-OS software

Learn how to upgrade the Nexus 9000 Series NX-OS software by preparing the switch, verifying image checksums, ensuring adequate space, copying and installing the correct image file, and validating the software version post-upgrade.

Use this procedure to upgrade to the latest NX-OS 10.6(x) release.

Note

- By default, the software upgrade process is disruptive.
- If an error message appears during the upgrade, the upgrade fails because of the reason indicated. For more information about possible causes and solutions, see the Cisco Nexus 9000 Series NX-OS Troubleshooting Guide.

Before you begin

- Read the release notes for the software image file for any exceptions to this upgrade procedure. For more information about software image file, see the Cisco Nexus 9000 Series NX-OS Release Notes.

Procedure

1. Log in to the device on the console port connection.

2. Ensure that the required space is available for the image file to be copied.

```
switch# dir bootflash:
16384    Oct 30 17:05:32 2020  lost+found/
1964291584    Dec 08 19:44:33 2020  nxos.10.1.1.bin
...
Usage for bootflash://sup-local
 4825743360 bytes used
16312102912 bytes free
21137846272 bytes total
```

> **Note**
>
> We recommend that you have the image file for at least one previous release of the NX-OS software on the device to use if the new image file does not load successfully.

a) If you need more space on the active supervisor module, delete unnecessary files to make space available.

```
switch# delete bootflash:nxos.9.2.1.bin
```

b) Verify that there is space available on the standby supervisor module.

```
switch# dir bootflash://sup-standby/
16384    Oct 30 17:05:32 2020  lost+found/
1964291584    Dec 08 19:44:33 2020  nxos.10.1.1.bin
...
Usage for bootflash://sup-standby
4825743360 bytes used
16312102912 bytes free
21137846272 bytes total
```

c) Optional: If you need more space on the standby supervisor module, delete any unnecessary files to make space available.

```
switch# delete bootflash://sup-standby/nxos.9.2.1.bin
```

3. Log in and choose the software image file for your device from the Software Download website, and download it to a file server.
4. Copy the software image to the active supervisor module using a transfer protocol. You can use FTP, TFTP, SCP, or SFTP.

```
switch# copy scp://user@scpserver.cisco.com//download/nxos64.10.2.1.F.bin
bootflash:nxos64.10.2.1.F.bin
```

> **Note**
>
> The compaction feature is deprecated from NX-OS Release 10.5(2)F

For software images requiring compaction, you must use SCP, HTTP, or HTTPS as the source and bootflash or USB as the destination. This example uses SCP and bootflash:

```
switch# copy scp://user@scpserver.cisco.com//download/nxos64.10.2.1.F.bin
bootflash:nxos64.10.2.1.F.bin compact vrf management use-kstack

user1@10.65.42.196's password:
nxos64.10.2.1.F.bin 100% 1887MB 6.6MB/s 04:47
Copy complete, now saving to disk (please wait)...
Copy complete.
```

The **compact** keyword compacts the NX-OS image before copying the file to the supervisor module.

> **Note**
>
> Software image compaction is only supported on SCP, HTTP, or HTTPS. If you attempt compaction with any other protocol, the system returns the following error:

```
Compact option is allowed only with source as scp/http/https and destination
as bootflash or usb
```

> **Note**
>
> Compacted images are not supported with LXC boot mode.

> **Note**
>
> Software image compaction is only supported on Nexus 9300-series platform switches.

a) You can detect an incomplete or corrupt NX-OS software image prior to performing an upgrade by verifying the MD5, SHA256 or SHA512 checksum of the software image. To verify the MD5 checksum of the software image, run the **show file bootflash:** *<IMAGE-NAME>***md5sum** command and compare the resulting value to the published MD5 checksum for the software image on Software Download website. To verify the SHA512 checksum of the

software image, run the **show file bootflash:** *<IMAGE-NAME>***sha512sum** command and compare the resulting value to the published SHA512 checksum for the software image on Cisco's Software Download website.

```
switch# show file bootflash:nxos.10.1.1.bin md5sum
2242a7f876f1304118fd175c66f69b34
```

```
switch# show file bootflash:nxos.10.1.1.bin sha512sum
7f25cce57ca137a79211fb3835338aae64acf9b021b75cec5d4156e873b4274ca4f98e9a74fe4c8961
f5ace99ed65f3826650599369f84ab07265d7c5d61b57f
```

b) You can detect an incomplete or corrupt NX-OS software image prior to performing an upgrade by verifying the MD5, SHA256 or SHA512 checksum of the software image. To verify the MD5 checksum of the software image, run the **show file bootflash:** *<IMAGE-NAME>***md5sum** command and compare the resulting value to the published MD5 checksum for the software image on Cisco's Software Download website. To verify the SHA512 checksum of the software image, run the **show file bootflash:** *<IMAGE-NAME>***sha512sum** command and compare the resulting value to the published SHA512 checksum for the software image on the Software Download website.

```
switch# show file bootflash:nxos64.10.2.1.F.bin md5sum
c49660952215822afd30bb7958a0765a
```

```
switch# show file bootflash:nxos64.10.2.1.F.bin sha256sum
2a64efbb381fabbb52054af74cf3efda1691772a49a70ddd35550431cadecf8e
```

```
switch# show file bootflash:nxos64.10.2.1.F.bin sha512sum
3bf6a771aa4a192a8e1383e348b26bb48356a9774d74ca39edbf771824848 3b3391942c8103de8104deea8fda212266e70bd736220cff34943bb8e35943297 5
```

5. Check the impact of upgrading the software before actually performing the upgrade.

```
switch# show install all impact nxos bootflash:nxos64.10.2.1.F.bin
```

During the compatibility check, the ISSU-related messages listed in this table can appear in the **Reason** field.

Reason Field Message	Description
Incompatible image for ISSU	The NX-OS image to which you are attempting to upgrade does not support ISSU.
Default upgrade is not hitless	By default, the software upgrade process is disruptive. You must configure the **non-disruptive** option to perform an ISSU.

6. Save the running configuration to the startup configuration.

```
switch# copy running-config startup-config
```

7. Upgrade the NX-OS software using the **install all nxos bootflash:***filename* [**no-reload** | **non-disruptive** | **non-interruptive** | **min-disruptive** | **serial**| **skip-epld** | **skip-bios-upgrade** | **skip-kernel-upgrade**] command.

```
switch# install all nxos bootflash:nxos64.10.5.3.F.bin
```

> **Note**
>
> From NX-OS Release 10.5(3)F, EPLD upgrade can be performed with either the no-reload or the non-disruptive option.

The available options are:

- **no-reload**–Exits the software upgrade process before the device reloads.

  > **Note**
  >
  > When you use **install all** with **no-reload** option, no additional configuration changes can be made before you reload the device. Saving configuration in this state can result in incorrect startup configuration once you reload the device with the new version of NX-OS (along with EPLD from 10.5(3)F). From Release 10.5(3)F onwards, though the EPLD and BIOS are programmed but not upgraded, a switch reload is required for them to take effect.

- **non-disruptive**–Performs an in-service software upgrade (ISSU) to prevent the disruption of data traffic. (By default, the software upgrade process is disruptive.) From Release 10.5(3)F onwards, though the EPLD and BIOS are programmed but not upgraded, a switch reload is required for them to take effect.

  > **Note**
  >
  > This option is not available on the Nexus 9364E-SG2 switches.

- **min-disruptive**–Performs an in-service software upgrade (ISSU) with minimal disruption of data traffic. This option is only available on Nexus 9364E-SG2 switches.
- **non-interruptive**–Upgrades the software without any prompts. This option skips all error and sanity checks.
- **serial**–Upgrades the I/O modules in Nexus 9500 Series switches one at a time. (By default, the I/O modules are upgraded in parallel, which reduces the overall upgrade time. Specifically, the I/O modules are upgraded in parallel in this order: the first half of the line cards and fabric modules, the second half of the line cards and fabric modules, the first system controller, the second system controller.)
- **skip-epld**–Installs only the nxos image, and not the epld image. This option is available from 10.5(3)F.
- **skip-bios-upgrade**–Installs only the nxos image, and skips the BIOS upgrade. This option is available from 10.5(3)F.
- **skip-kernel-upgrade**–Installs only the nxos image, and skips the kernel upgrade. This option is available from 10.5(3)F.

Note

- If you enter the **install all** command without specifying a filename, the command performs a compatibility check, notifies you of the modules that will be upgraded, and confirms that you want to continue with the installation. If you choose to proceed, it installs the NX-OS software image that is currently running on the switch and upgrades the BIOS of various modules from the running image, if necessary.
- Starting from 10.5(3)F, during the image upgrade, you can also apply SMUs so the SMU is installed with the new image using the **install all nxos** *<nxos image>* **package** *<smu package>* **non-disruptive non-interruptive** command.

```
install all nxos nxos64-cs.10.5.3.F.bin.upg package nxos64-cs.CSCab123456
-1.0.0-10.5.3.lib32_64_n9000.rpm non-disruptive non-interruptive
```

8. Optional: Post upgrade actions.
   a) Display the entire upgrade process.

```
switch# show install all status
```

   b) Log in and verify that the device is running the required software version.

```
switch# show version
```

   c) If necessary, install the relevant licenses to ensure that the required features are available on the device. For more information about the licenses, see the Cisco NX-OS Licensing Options Guideand Cisco Nexus 9000 and 3000 Series NX-OS Smart Licensing Using Policy User Guide.

In-Service Software Upgrade

This section provides details about in-service software upgrade (ISSU), a process that allows you to upgrade the device without disrupting the traffic, minimizing or eliminating downtime. It also covers supported ISSU scenarios and platforms, including standard and enhanced ISSU for Top-of-Rack switches, and explains configuration requirements, traffic impact, and limitations for N9000 Series switches.

An in-service software upgrade (ISSU) is an upgrade that

- allows you to upgrade the device software while the switch continues to forward traffic,

  reduces or eliminates the downtime typically caused by software upgrades, and

  is also known as non-disruptive upgrade.

You can perform an ISSU or non-disruptive upgrade for some switches. (See the ISSU platform support on page 20 for a complete list of supported platforms.)

- The default upgrade process is disruptive. Therefore, ISSU needs to be enabled using the command-line interface (CLI), as described in the configuration section of this document.
- Using the non-disruptive option helps ensure a non-disruptive upgrade. The guest shell is disabled during the ISSU process and it is later reactivated after the upgrade.

- Enhanced ISSUs are supported for some N9000 Series switches.

ISSU scenarios and platform support

The supported ISSU scenarios include

- performing standard ISSU on Top-of-Rack (ToR) switches with a single supervisor, and
- performing enhanced ISSU on Top-of-Rack (ToR) switches with a single supervisor

Details for each scenario are described below.

Performing standard ISSU on Top-of-Rack (ToR) switches with a single Supervisor

The ToR N9300 platform switches are the NX-OS switches with single supervisors. Performing ISSU on the N9000 Series switches causes the supervisor CPU to reset and to load the new software version. After the CPU loads the updated version of the NX-OS software, the system restores the control plane to the previous known configuration and the runtime state and it gets in-sync with the data plane, thereby completing the ISSU process.

The data plane traffic is not disrupted during the ISSU process. In other words, the data plane forwards the packets while the control plane is being upgraded, any servers that are connected to the N9000 Series switches do not see any traffic disruption. The control plane downtime during the ISSU process is approximately less than 120 seconds.

Performing enhanced ISSU on Top-of-Rack (ToR) switches with a single Supervisor

Note

Enhanced ISSU is not supported if there are any underlying kernel differences. In effect, the system performs non-disruptive ISSU instead of enhanced ISSU. The system prompts the message: `Host kernel is not compatible with target image. Full ISSU will be performed and control plane will be impacted.`

The NX-OS software normally runs directly on the hardware. However, configuring enhanced or container-based ISSU on single supervisor ToRs is accomplished by creating virtual instances of the supervisor modules and the line cards. With enhanced ISSU, the software runs inside a separate Linux container (LXC) for the supervisors and the line cards. A third container is created as part of the ISSU procedure, and it is brought up as a standby supervisor.

The virtual instances (or the Linux containers) communicate with each other using an emulated Ethernet connection. In the normal state, only two Linux containers are instantiated: vSup1 (a virtual SUP container in an active role) and vLC (a virtual linecard container). Enhanced ISSU requires 16G memory on the switch.

During the software upgrade with enhanced ISSU, the supervisor control plane stays up with minimal switchover downtime disruption and the forwarding state of the network is maintained accurately during the upgrade. The supervisor is upgraded first and the line card is upgraded next.

The data plane traffic is not disrupted during the ISSU process. The control plane downtime is less than 6 seconds.

Note

In-service software downgrades (ISSDs), also known as non-disruptive downgrades, are not supported.

For information on ISSU and high availability, see the Cisco N9000 Series NX-OS High Availability and Redundancy Guide.

NX-OS upgrade history

This section provides details about how Nexus 9000 switches track and display a comprehensive history of all upgrade activities. Viewing this upgrade history helps with troubleshooting and understanding the sequence and status of past upgrades performed on the switch.

During the life of a Nexus 9000 switch, many upgrade procedures can be performed. Upgrades can occur for maintenance purposes or to update the operating system to obtain new features. Over time, switches can be updated on numerous occasions. Viewing the types of upgrades and when they occurred can help in troubleshooting issues or simply understanding the history of the switch.

Nexus 9000 switches log all upgrade activity performed over time providing a comprehensive history of these events. The stored upgrade history types are:

- Cisco NX-OS System Upgrades
- Electronic Programmable Logic Device (EPLD) Upgrades, and
- Software Maintenance Upgrade (SMU) Installations

View the NX-OS upgrade history by entering the **show upgrade history** command. The output displays any upgrade activity that previously occurred on the switch and defines the start and end times for each event. The following is an example output of the **show upgrade history** command:

```
switch# show upgrade history
     TYPE            VERSION               DATE                STATUS

NXOS system image  10.5(3)         06 Mar 2025 03:08:17     Installation End

NXOS system image  10.5(3)        06 Mar 2025 03:01:41    Installation started
NXOS system image  10.5(3)         06 Mar 2025 02:16:25     Installation End

NXOS EPLD          epld.img        06 Mar 2025 02:11:43     EPLD Upgrade
completed
NXOS EPLD         epld.img        06 Mar 2025 02:00:22    EPLD Upgrade started
NXOS system image  10.5(3)        06 Mar 2025 01:52:46    Installation started
NXOS system image  10.5(3)         06 Mar 2025 01:19:05     Installation End

NXOS EPLD          epld.img        06 Mar 2025 01:19:04     EPLD Upgrade
completed
NXOS EPLD         epld.img        06 Mar 2025 01:07:42    EPLD Upgrade started
NXOS system image  10.5(3)        06 Mar 2025 01:00:48    Installation started
```

View the NX-OS upgrade history details by entering the **show upgrade history details** command. The output displays user login details (user name/session ID) under LOGIN column on the switch along with upgrade history. Here is an sample output of the **show upgrade history details** command.

```
switch# show upgrade history details
   TYPE            VERSION                  DATE                  LOGIN
                  STATUS
NXOS system image  10.5(3)                 06 Mar 2025 03:08:17
username/10.1.1.25     Installation End
NXOS system image  10.5(3)                 06 Mar 2025 03:01:41
username/10.1.1.25     Installation started
NXOS system image  10.5(3)                 06 Mar 2025 02:16:25
username/10.1.1.25     Installation End
NXOS EPLD          epld.img                06 Mar 2025 02:11:43
            EPLD Upgrade completed
NXOS EPLD          epld.img                06 Mar 2025 02:00:22
            EPLD Upgrade started
```

```
NXOS system image  10.5(3)                 06 Mar 2025 01:52:46
username/10.1.1.25     Installation started
NXOS system image  10.5(3)                 06 Mar 2025 01:19:05
username/10.1.1.25     Installation End
NXOS EPLD          epld.img                06 Mar 2025 01:19:04
            EPLD Upgrade completed
NXOS EPLD          epld.img                06 Mar 2025 01:07:42
            EPLD Upgrade started
NXOS system image  10.5(3)                 06 Mar 2025 01:00:48
username/10.1.1.25     Installation started
NXOS EPLD          n9000-epld.10.5.1.F.img 05 Mar 2025 23:29:18
username/10.1.1.25     EPLD Upgrade completed
NXOS EPLD          n9000-epld.10.5.1.F.img 05 Mar 2025 23:17:59
username/10.1.1.25     EPLD Upgrade started
NXOS EPLD          epld.img                05 Mar 2025 22:55:11
            EPLD Upgrade completed
NXOS EPLD          epld.img                05 Mar 2025 22:37:36
            EPLD Upgrade started
```

Examples of upgrade types

Presents sample displays for various ISSU upgrade types on NX-OS platforms, including disruptive, non-disruptive, and min-disruptive upgrades for Silicon One EOR, CloudScale TOR, and related modules.

This section provides you sample displays of various upgrade types:

- Disruptive ISSU for Silicon One EOR
- Disruptive ISSU for CloudScale TOR
- Non-Disruptive ISSU CloudScale TOR
- Min-Disruptive ISSU Silicon One TOR

For more information about various types of upgrade in NX-OS, refer to NX-OS software upgrade on page 11.

Disruptive ISSU for Silicon One EOR

```
=====================
switch# show module
Mod Ports                  Module-Type                            Model
     Status
--- ----- ------------------------------------------------ ---------------------
 --------
1    36   36x400G QSFP56-DD Ethernet Module               N9K-X9836DM-A
    ok
2    36   36x400G QSFP56-DD Ethernet Module               N9K-X9836DM-A
    ok
3    36   36x400G QSFP56-DD Ethernet Module               N9K-X9836DM-A
    ok
4    36   36x400G QSFP56-DD Ethernet Module               N9K-X9836DM-A
    ok
5    36   36x400G QSFP56-DD Ethernet Module               N9K-X9836DM-A
    ok
6    36   36x400G QSFP56-DD Ethernet Module               N9K-X9836DM-A
    ok
7    36   36x400G QSFP56-DD Ethernet Module               N9K-X9836DM-A
    ok
8    36   36x400G QSFP56-DD Ethernet Module               N9K-X9836DM-A
    ok
```

```
27   0    Supervisor Module                                  N9K-C9800-SUP-B
    active *
28   0    Supervisor Module                                  N9K-C9800-SUP-B
    ha-standby

Mod  Sw                      Hw    Slot
---  ----------------------- ------ ----
1    10.6(2)                 1.0    LC1
2    10.6(2)                 1.0    LC2
3    10.6(2)                 1.0    LC3
4    10.6(2)                 1.0    LC4
5    10.6(2)                 1.0    LC5
6    10.6(2)                 1.0    LC6
7    10.6(2)                 1.0    LC7
8    10.6(2)                 1.0    LC8
27   10.6(2)                 1.0    SUP1
28   10.6(2)                 1.0    SUP2


Mod  MAC-Address(es)                         Serial-Num
---  --------------------------------------  ----------
1    68-87-c6-94-25-b8 to 68-87-c6-94-26-df  FOC2542N39J
2    20-cf-ae-4b-7a-d0 to 20-cf-ae-4b-7b-f7  FOC2533N3VZ
3    14-a2-a0-5a-26-18 to 14-a2-a0-5a-27-3f  FOC2549NKWJ
4    d4-eb-68-9d-71-40 to d4-eb-68-9d-72-67  FOC2537NA00
5    20-cf-ae-e5-82-68 to 20-cf-ae-e5-83-8f  FOC2537N9V2
6    c4-4d-84-e0-ec-18 to c4-4d-84-e0-ed-3f  FOC2543P5D5
7    20-cf-ae-e5-98-60 to 20-cf-ae-e5-99-87  FOC2537N9W3
8    e8-d3-22-7f-69-c0 to e8-d3-22-7f-6a-e7  FOC2620NSU3
27   d4-7f-35-fd-55-00 to d4-7f-35-fd-55-07  FOC2826N2LW
28   d4-7f-35-fd-54-98 to d4-7f-35-fd-54-9f  FOC2826N2MF

Mod  Online Diag Status
---  ------------------
1    Pass
2    Pass
3    Pass
4    Pass
5    Pass
6    Pass
7    Pass
8    Pass
27   Pass
28   Pass
LFM Ports                  Module-Type                          Model
     Status
--- ----- ------------------------------------------------ ----------------------
 --------
19   0    8-slot Fabric Module                             N9K-C9808-FM-A
    ok
20   0    8-slot Fabric Module                             N9K-C9808-FM-A
    ok
21   0    8-slot Fabric Module                             N9K-C9808-FM-A
    ok
22   0    8-slot Fabric Module                             N9K-C9808-FM-A
    ok
23   0    8-slot Fabric Module                             N9K-C9808-FM-A
    ok
24   0    8-slot Fabric Module                             N9K-C9808-FM-A
    ok
25   0    8-slot Fabric Module                             N9K-C9808-FM-A
    ok
```

```
26   0    8-slot Fabric Module                              N9K-C9808-FM-A
    ok

Mod  Sw                      Hw    Slot
---  --------------------- ------ ----
19   NA                      1.0    FM1
20   NA                      1.0    FM2
21   NA                      1.0    FM3
22   NA                      1.0    FM4
23   NA                      1.0    FM5
24   NA                      1.0    FM6
25   NA                      1.0    FM7
26   NA                      1.0    FM8

LFM MAC-Address(es)                          Serial-Num
---  --------------------------------------  ----------
19   NA                                       FOC2535N6HZ
20   NA                                       FOC2540N0ZZ
21   NA                                       FOC2535N6GC
22   NA                                       FOC2540N0WA
23   NA                                       FOC2525NP5C
24   NA                                       FOC2540N101
25   NA                                       FOC2534N308
26   NA                                       FOC2546N2GP

LFM  Online Diag Status
---  ------------------
19   Pass
20   Pass
21   Pass
22   Pass
23   Pass
24   Pass
25   Pass
26   Pass
switch# install all nxos bootflash:nxos64-s1.10.6.2.IMG9.0.222.F.bin.tmp
Installer will perform compatibility check first. Please wait.
Installer is forced disruptive

Verifying image bootflash:/nxos64-s1.10.6.2.IMG9.0.222.F.bin.tmp for boot
variable "nxos".
[####################] 100% -- SUCCESS

Verifying EPLD/FPGA image //bootflash/nxos64-s1.10.6.2.IMG9.0.222.F.bin.tmp.
[####################] 100% -- SUCCESS

Verifying image type.
[####################] 100% -- SUCCESS

Preparing "bios" version info using image
bootflash:/nxos64-s1.10.6.2.IMG9.0.222.F.bin.tmp.
[####################] 100% -- SUCCESS

Preparing "nxos" version info using image
bootflash:/nxos64-s1.10.6.2.IMG9.0.222.F.bin.tmp.
[####################] 100% -- SUCCESS

Performing module support checks.
[####################] 100% -- SUCCESS

Notifying services about system upgrade.
[####################] 100% -- SUCCESS
```

```
Compatibility check is done:
Module  bootable          Impact  Install-type  Reason
------  --------  --------------  ------------  ------
     1       yes      disruptive         reset  default upgrade is not hitless

     2       yes      disruptive         reset  default upgrade is not hitless

     3       yes      disruptive         reset  default upgrade is not hitless

     4       yes      disruptive         reset  default upgrade is not hitless

     5       yes      disruptive         reset  default upgrade is not hitless

     6       yes      disruptive         reset  default upgrade is not hitless

     7       yes      disruptive         reset  default upgrade is not hitless

     8       yes      disruptive         reset  default upgrade is not hitless

    27       yes      disruptive         reset  default upgrade is not hitless

    28       yes      disruptive         reset  default upgrade is not hitless


Images will be upgraded according to following table:
Module       Image                 Running-Version(pri:alt)
New-Version  Upg-Required
------  ----------  ----------------------------------------
--------------------  ------------
     1       lcn9k                                     10.6(2)
10.6(3)           yes
     1        bios     v01.05(12/03/2024):v01.00(03/09/2020)
v01.05(12/03/2024)            no
     2       lcn9k                                     10.6(2)
10.6(3)           yes
     2        bios     v01.05(12/03/2024):v01.02(03/26/2022)
v01.05(12/03/2024)            no
     3       lcn9k                                     10.6(2)
10.6(3)           yes
     3        bios     v01.05(12/03/2024):v01.00(03/09/2020)
v01.05(12/03/2024)            no
     4       lcn9k                                     10.6(2)
10.6(3)           yes
     4        bios     v01.05(12/03/2024):v00.01(03/09/2020)
v01.05(12/03/2024)            no
     5       lcn9k                                     10.6(2)
10.6(3)           yes
     5        bios     v01.05(12/03/2024):v00.01(03/09/2020)
v01.05(12/03/2024)            no
     6       lcn9k                                     10.6(2)
10.6(3)           yes
     6        bios     v01.05(12/03/2024):v01.00(03/09/2020)
v01.05(12/03/2024)            no
     7       lcn9k                                     10.6(2)
10.6(3)           yes
     7        bios     v01.05(12/03/2024):v01.00(03/09/2020)
v01.05(12/03/2024)            no
     8       lcn9k                                     10.6(2)
```

```
10.6(3)           yes
     8        bios     v01.05(12/03/2024):v01.00(03/09/2020)
v01.05(12/03/2024)           no
    27        nxos                                   10.6(2)
10.6(3)           yes
    27        bios     v01.02(03/18/2025):v01.01(03/18/2024)
v01.02(03/18/2025)           no
    28        nxos                                   10.6(2)
10.6(3)           yes
    28        bios     v01.02(03/18/2025):v01.01(03/18/2024)
v01.02(03/18/2025)           no


FPGA microcode will be upgraded according to following table:
Module  Type  EPLD             Running-Version  Flashed-Version*   New-Version
  Upg-Required
------  ----  -------------     ---------------  ----------------   -----------
  ------------
     1    LC  MI FPGA            0x10014          0x10014           0x10014
               No
     1    LC  IO FPGA            0x10033          0x10033           0x10033
               No
     2    LC  MI FPGA            0x10014          0x10014           0x10014
               No
     2    LC  IO FPGA            0x10033          0x10033           0x10033
               No
     3    LC  MI FPGA            0x10014          0x10014           0x10014
               No
     3    LC  IO FPGA            0x10033          0x10033           0x10033
               No
     4    LC  MI FPGA            0x10014          0x10014           0x10014
               No
     4    LC  IO FPGA            0x10033          0x10033           0x10033
               No
     5    LC  MI FPGA            0x10014          0x10014           0x10014
               No
     5    LC  IO FPGA            0x10033          0x10033           0x10033
               No
     6    LC  MI FPGA            0x10014          0x10014           0x10014
               No
     6    LC  IO FPGA            0x10033          0x10033           0x10033
               No
     7    LC  MI FPGA            0x10014          0x10014           0x10014
               No
     7    LC  IO FPGA            0x10033          0x10033           0x10033
               No
     8    LC  MI FPGA            0x10014          0x10014           0x10014
               No
     8    LC  IO FPGA            0x10033          0x10033           0x10033
               No
    19    FM  MI FPGA            0x10006          0x10006           0x10006
               No
    20    FM  MI FPGA            0x10006          0x10006           0x10006
               No
    21    FM  MI FPGA            0x10006          0x10006           0x10006
               No
    22    FM  MI FPGA            0x10006          0x10006           0x10006
               No
    23    FM  MI FPGA            0x10006          0x10006           0x10006
               No
    24    FM  MI FPGA            0x10006          0x10006           0x10006
               No
```

```
    25    FM  MI FPGA            0x10006           0x10006           0x10006
              No
    26    FM  MI FPGA            0x10006           0x10006           0x10006
              No
    27   SUP  MI FPGA            0x10001           0x10001           0x10001
              No
    27   SUP  IO FPGA            0x10010           0x10010           0x10010
              No
    28   SUP  MI FPGA            0x10001           0x10001           0x10001
              No
    28   SUP  IO FPGA            0x10010           0x10010           0x10010
              No
* If Running-Version and Flashed-Version are different it implies that the
system has not yet been reloaded for the new version to take effect


Additional info for this installation:
--------------------------------------

Service "sflow" in vdc 1: SFLOW feature is enabled. Upgrade will pause sFlow
report for about 1 ~ 2 minutes.



Switch will be reloaded for disruptive upgrade.
Do you want to continue with the installation (y/n)?  [n] y


Install is in progress, please wait.
[#                   ]   0%
Performing runtime checks.

Syncing image bootflash:/nxos64-s1.10.6.2.IMG9.0.222.F.bin.tmp to standby.
[#                   ]   0%
Syncing image bootflash:/nxos64-s1.10.6.2.IMG9.0.222.F.bin.tmp to standby.
[####################] 100% -- SUCCESS

Setting boot variables.
[#                   ]   0%
Setting boot variables.
[####################] 100% -- SUCCESS

Performing configuration copy.
[####################] 100%
Performing configuration copy.
[####################] 100% -- SUCCESS

Module 1: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#                   ]   0%
Module 1: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

Module 2: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#                   ]   0%
Module 2: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

Module 3: Refreshing compact flash and upgrading bios/loader/bootrom.
```

```
Warning: please do not remove or power off the module at this time.
[#                    ]   0%
Module 3: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

Module 4: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#                    ]   0%
Module 4: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

Module 5: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#                    ]   0%
Module 5: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

Module 6: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#                    ]   0%
Module 6: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

Module 7: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#                    ]   0%
Module 7: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

Module 8: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#                    ]   0%
Module 8: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

Module 19: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#                    ]   0%
Module 19: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

Module 20: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#                    ]   0%
Module 20: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

Module 21: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#                    ]   0%
Module 21: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS
```

```
Module 22: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#                    ]   0%
Module 22: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

Module 23: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#                    ]   0%
Module 23: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

Module 24: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#                    ]   0%
Module 24: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

Module 25: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#                    ]   0%
Module 25: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

Module 26: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#                    ]   0%
Module 26: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

Module 27: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#                    ]   0%
Module 27: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

Module 28: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#                    ]   0%
Module 28: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

[#                    ]   0%
[####################] 100% -- SUCCESS

All EPLD/FPGAs are up to date


Finishing the upgrade, switch will reboot in 10 seconds.
switch#
User Access Verification
switch login:


Cisco N9800(R) Series BIOS Ver 1.02 Primary
```

```
Intel(R) Xeon(R) CPU D-1573N @ 2.70GHz
Board Type 0x223 PID N9K-C9808 Serial FOX2545PT35
Switch G28
Chipguard Init
TAM Library Version: 5.1.1
Aikido FW Version: 2.7.16
Aikido Chip Serial Number:
5061D44815A08BFE6899C983D357D0A400000000000000000000000000000000
Aikido FPGA Version: 0x110
Aikido chip authentication successful
Aikido KEY Format: UEFI
Cisco PK Key
Cisco Key Key
Cisco DBX Key
Cisco OS Rel Key
Cisco OS MFG Dev Key

Board type  5
IOFPGA @ 0xa1400000
SLOT_ID @ 0x1b

NXOS Grub
disable watchdog
check_bootmode: grub: Continue grub
Trying to read config file /boot/grub/menu.lst.local from (hd0,gpt5)
Booting bootflash:/nxos64-s1.10.6.2.IMG9.0.222.F.bin.tmp
Trying diskboot
 Filesystem type is ext2fs, partition type gpt.


Image Signature verification was Successful.

Boot Time: 3/24/2026  16:30:53
[    2.349441] tpm tpm0: A TPM error (256) occurred attempting the self test
71+1 records in
71+1 records out
300548096 bytes (301 MB, 287 MiB) copied, 0.127147 s, 2.4 GB/s
mount: overlay mounted on /newroot/usr.
Installing klm_card_index
done
Setting nativeboot
Linking n9k mando SSD partition...
creating flash devices BOOT_DEV= nvme0n1
INIT: version 3.04 booting
Installing ata_piix module ... done.
Unsquashing rootfs ...
vm.min_free_kbytes = 368640
Enabling 8250 serial driver spurious INTs workaround
Installing isan procfs ... done.
is_lxc: is_titan_eor: is_stby: suffix: klm_ftrace:
/isanboot/lib/modules/klm_ftrace.o
Installing ftrace in non-lxc mode done
Installing SSE module with card index 21262 ... done.
Creating SSE device node 240 ... done.
17.19: card_index: 21262, card_type: 132
Loading I2C driver ... done.
Installing CCTRL driver for card_type 132 without NEED_GEM ... [   21.184436]
 ENABLING TOP LEVEL RP INTERRUPT
[   21.190465] Got RP P2PM intr, Becoming Active !!
21.20: FPGA test device /dev/xr-fpga -c 237 0 created
done.
Loading IGB driver ...
```

```
Loading Intel igb driver for S1 platforms...21.44: \nInstalling klm_ins_ixgbe
 start

Installing klm_ins_ixgbe ... done.

Checking all filesystems.
Extracting rpms from image...
/
Installing SPROM driver ... 21262 IS_N9K done.
Installing pfmsvcs module ...done.
Installing nvram module ... done.
Installing if_index module with port mode 6 ... done.
Installing fcfwd
Installing RNI lcnd ... done.
Installing lcnd ... done.
\n S090setup-eth sup_start
Installing psdev ...
Installing veobc module ... done.
Error: ipv4: Address already assigned.
Clean up previous pcap files present in tmp directory
Checking SR card
Card Index is 21262
64.47: Inserting obfl & mtd spi driver ...
old data: 0 new data: 9f000000
old data: 0 new data: 0
old data: 0 new data: a0901406
[   64.500854] Initializing panic NMI
[   64.508912] fpga_setup_probe
64.54: Done..Inserting obfl & mtd spi driver ...
Making OBFL character devices
old data: 9f000000 new data: 9f000000
old data: 0 new data: 0
old data: a0901406 new data: a0901406
mounting plog for N9k!
64.58: mounting plog for N9k!
64.60: Done..mounting plog for N9k!
mounting recovery for N9k!
64.62: mounting recovery for N9k!
Mounting OBFL pstore for mtd
Inserting kernel_services module ... done.
Making kernel_services character devices
cgroups initialized
        package cgroup-lite-1.15-r0.corei7_64 is already installed
        package util-linux-unshare-2.39.3-r0.corei7_64 is already installed
kernel.core_pattern = |/bin/sparsecore.sh %p %h %P %E
minicoredumper disabled for this platform...
exit code: 1
COPY the image to ORIG_LC_FS_PATH
/isanboot/bin/images/s1_gibraltar_xc_sdk.bin
/isanboot/bin/images/s1hal_n9000.bin
tune2fs 1.47.0 (5-Feb-2023)
Setting reserved blocks percentage to 0% (0 blocks)
Persistent log store is mounted successfully
Starting rpcbind daemon...done.
creating NFS state directory: done
starting 8 nfsd kernel threads: done
starting mountd: done
starting statd: done
Saving image for img-sync ...
Installing kheaders module ... done.
Loading system software
Installing local RPMS
```

```
Patch Repository Setup completed successfully
/logflash/stats_ssd/.ssd_overall_data file exist !!
moving /logflash/stats_ssd/.ssd_overall_data to
/logflash/stats_ssd/.ssd_lastboot_data
Starting crond: OK
chown: warning: '.' should be ':': 'root.root'
Stopping crond: OK
Creating /dev/mcelog
Starting mcelog daemon
Starting crond: OK
Remove pltfm_config from eltm.conf for SONE cards
Replacement complete. All system shutdown binaries now point to /sbin/rep_file.
INIT: Entering runlevel: 3
link S1 SDK libs for SUP
SDK Shell for Gibraltar
ASIC_GIBRALTAR is y
ASIC_IRON is
Moving Sone libs for gibraltar
Linking tctrl_usd for GROGU RP2
Moving tctrl_usd_dual_sup.conf to tctrl_usd.conf
..Done
Running S93thirdparty-script...
vlc_card_index 0
S26: sw_id 21262 vp 0
S26 C: sw_id 21262, Retaining libcrdcfgdata.so as is
CAUTION!!!!!!!CAUTION!!!!/bootflash/platform.inf File PRESENT
packagegroup-nxos-optional-rpms_pkg_list
Starting lem_drv driver
Inserting klm_lfm IS_LEM_CARD=y
LEM card. Insert driver
0x25
FM DIVF = 0x25
[  102.637118] Initializing delayed work
[  102.646884] Finished INIT function for lfm
LEM card. Make device
Executing /isan/etc/rc.d/rc.isan-start/S45s1drv start
done
klm_sone installed successfully
Installing marvell
pcie_proxy module installing for Marvell
Adding Marvel switch to pci_proxy module
dma proxy module installed
Adding Zenith FPGA to pci_proxy module
Netbroker support IS present in the kernel.
done
Executing Prune clis.
Removing exusd cli files if it is not exbl_card in prune_clis
ethernet switching mode Tue Mar 24 09:32:53 PDT 2026
prepare span CLI
[  117.286629] register_rdn_int : Sent First Heartbeat

User Access Verification
switch login:
```

Disruptive ISSU for CloudScale TOR

```
=======================

switch# show module
Mod Ports                 Module-Type                              Model
```

```
  Status
--- ----- ---------------------------------------------- ----------------------
--------
1    50   48x400G QSFP-DD + 2x10G SFP+ Ethernet Module     N9K-C9348D-GX2A

 ok
27   0    Virtual Supervisor Module                        N9K-C9348D-GX2A

 active *

Mod  Sw                      Hw     Slot
---  ----------------------- ------ ----
1    10.5(3)                 0.2030 NA
27   10.5(3)                 0.2030 VSUP


Mod  MAC-Address(es)                          Serial-Num
---  ---------------------------------------  ----------
1    4c-5d-3c-41-af-ae to 4c-5d-3c-41-b0-7b  FDO25210LJX
27   4c-5d-3c-41-af-ae to 4c-5d-3c-41-b0-7b  FDO25210LJX

Mod  Online Diag Status
---  ------------------
1    Pass
27   Pass

* this terminal session
switch# install all nxos bootflash:nxos64-cs.10.6.1.F.bin
Installer will perform compatibility check first. Please wait.
Installer is forced disruptive

Verifying image bootflash:/nxos64-cs.10.6.1.F.bin for boot variable "nxos".
[####################] 100% -- SUCCESS

Verifying EPLD/FPGA image //bootflash/nxos64-cs.10.6.1.F.bin.
[####################] 100% -- SUCCESS

Verifying image type.
[####################] 100% -- SUCCESS

Preparing "nxos" version info using image bootflash:/nxos64-cs.10.6.1.F.bin.
[####################] 100% -- SUCCESS

Preparing "bios" version info using image bootflash:/nxos64-cs.10.6.1.F.bin.
[####################] 100% -- SUCCESS

Performing module support checks.
[####################] 100% -- SUCCESS

Notifying services about system upgrade.
[####################] 100% -- SUCCESS



Compatibility check is done:
Module  bootable          Impact  Install-type  Reason
------  --------  --------------  ------------  ------
     1       yes      disruptive         reset  default upgrade is not hitless

    27       yes      disruptive         reset  default upgrade is not hitless
```

```
Images will be upgraded according to following table:
Module       Image                 Running-Version(pri:alt)
New-Version  Upg-Required
------  ----------  ----------------------------------------
--------------------  ------------
     1       lcn9k                                    10.5(3)
10.6(1)           yes
    27        nxos                                    10.5(3)
10.6(1)           yes
    27        bios     v01.10(11/28/2024):v01.03(07/14/2021)
v01.10(11/28/2024)            no


FPGA microcode will be upgraded according to following table:
Module  Type   EPLD             Running-Version   New-Version  Upg-Required
------  ----  -------------     ---------------   -----------  ------------
    27   SUP  MI FPGA                   0x08        0x10            Yes
    27   SUP  IO FPGA                   0x15        0x16            Yes
    27   SUP  MIFPGA2                   0x05        0x07            Yes

EPLD Upgrade may result in multiple modules going offline.

Switch will be reloaded for disruptive upgrade.
Do you want to continue with the installation (y/n)?  [n] y


Install is in progress, please wait.

Setting boot variables.
[####################] 100% -- SUCCESS

Performing configuration copy.
[####################] 100% -- SUCCESS

Module 1: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

Module 27: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

EPLD/FPGA upgrade can take upto  12  mins.

Performing EPLD/FPGA upgrade .
[#                   ]   0%
[####################] 100% -- SUCCESS
2026 Mar 24 10:56:55 switch %$ VDC-1 %$ %VMAN-2-ACTIVATION_STATE: Successfully
 deactivated virtual service 'guestshell+'


Finishing the upgrade, switch will reboot in 10 seconds.
switch# $
CISCO MODULE
BIOS Ver: 1.10
Switch G16
RC Revision:  02.09.00

Memory Information:
MRC Revision:00.50.00
Total  DRAM: 32768 MB
```

```
Memory TOLM: 80000000
PCIE   BASE: 80000000     Size : 10000000
PCI32  BASE: 90000000     Limit: FBFFFFFF
PCI64  BASE: 80000000000     Limit: 83FFFFFFFF
UC    START: 80000000000     End  : 84000000000
ME Operational Firmware Version: 06:3.0.3.214

DIMM Information:
Clock Speed: 1067MHz
Socket: 0x0 Channel: 0x0 Number: 0x0 Presence: Yes Size: 32GB
Socket: 0x0 Channel: 0x0 Number: 0x1 Presence: No
Socket: 0x0 Channel: 0x1 Number: 0x0 Presence: No
Socket: 0x0 Channel: 0x1 Number: 0x1 Presence: No

CPU Signature - 0x00050665: Version - 0x00050660
CPU - 1 : Cores - 6 : HTEn - 1 : HT - 2 : Features - 0xBFEBFBFF
MicroCode Version : 0x0E000015

Detected CISCO IOFPGA
Booting from Primary BIOS
 FPGA Revision          : 0x16
FPGA ID               :  0x21161472
FPGA Date             :  0x20220324
Power Debug Register1  : 0x0
Power Debug Register2  : 0x27
Reset Cause Register   : 0x80000000
Boot Ctrl Register     : 0xE0FF
FPGA Update Status     : 0x20

Cisco Aikdio Chip Init
TAM Library Version: 5.3.1
Aikido Version: 7.0.4
Aikido Chip Serial Number:
BB18EDE8618AC1C5DD21F2840C1F699700000000000000000000000000000000
Aikido FPGA Version: 0x10
Aikido chip authentication status is successful
Aikido KEY Format: UEFI
Found DEV KEY in the CISCO_DB_MFG Object

Detected CISCO MIFPGA
MIFPGA Subsystem Vendor ID 0x1172
MIFPGA Version        : 0x10
MIFPGA Date           : 0x20220912
MIFPGA Update Status  : 0x20
MIFPGA ID             : 0x20973490

Board Type: Petrus

Detected CISCO MPFPGA
MPFPGA Subsystem Vendor ID 0x1172
MPFPGA Version        : 0x7
MPFPGA Date           : 0x20220912
MPFPGA Update Status  : 0x20
MPFPGA ID             : 0x20973491

Bootable Disk is detected. Device Name: INTEL SSDSCKKB240G8K
fogbreaker BIOS: v01.10.0 Date: 11/28/2024 01:12:44
MPFPGA SPI Flash Micron(Numonyx) N25Q128
MIFPGA SPI Flash Micron(Numonyx) N25Q128

Board type  4
IOFPGA @ 0xe8000000
```

```
Set fan speed to 60%
Initializing fan controller...
check_bootmode: grub: Continue grub
Trying to read config file /boot/grub/menu.lst.local from (hd0,msdos1)
Trying to read config file /boot/grub/menu.lst.local from (hd1,gpt5)

Security Lock
Booting bootflash:/nxos64-cs.10.6.1.F.bin
Trying diskboot
 Filesystem type is ext2fs, partition type gpt.


Image Signature verification was Successful.

Boot Time: 3/24/2026  18:1:7

Security Lock
62+1 records in
62+1 records out
261570560 bytes (262 MB, 249 MiB) copied, 0.121876 s, 2.1 GB/s
mount: overlay mounted on /newroot/usr.
Installing klm_card_index
done
[    3.953788] sd 6:0:0:0: [sdb] No Caching mode page found
[    3.959738] sd 6:0:0:0: [sdb] Assuming drive cache: write through
Linking n9k flash devices
creating flash devices BOOT_DEV= sda
Replacing inittab for node - host
INIT: version 3.01 booting
Setting up sysV scripts for host node
Installing ata_piix module ... done.
Installing kheaders module ... done.
Unsquashing rootfs ...
is_lxc:1 is_titan_eor: is_stby: suffix: klm_ftrace:
/isanboot/lib/modules/klm_ftrace.o
Installing ftrace in lxc 0 mode done

Checking all filesystems.cgroups initialized
Generating ssh keys...Done ssh keys
Starting OpenBSD Secure Shell server: sshd ... done sshd.
Checking rpmstore swap in host context.
tune2fs 1.46.5 (30-Dec-2021)
Setting reserved blocks percentage to 0% (0 blocks)
INIT: Entering runlevel: 3
cp: cannot stat '/orig_fs/xr_cpa.img': No such file or directory
grep: /bootflash/platform.inf: No such file or directory


mount: overlay mounted on /usr.
Linking n9k flash devices
creating flash devices BOOT_DEV= sda
Replacing inittab for node - vsup
INIT: version 3.01 booting
Unsquashing rootfs ...
vm.min_free_kbytes = 184320
No TOR virtual chassis
skip untra_klms
Enabling 8250 serial driver spurious INTs workaround
Installing isan procfs isan_proc_name=isan26... done.
Installing SSE module with card index 21211 and vslot ID 26 ... done.
Creating SSE device node 241 ... done.
cg controllers already mounted at /var/cgroup
```

```
is_lxc:1 is_titan_eor: is_stby:0 suffix: klm_ftrace:
/isanboot/lib/modules/klm_ftrace.o
Inserting kernel_services module ... done.
Making kernel_services character devices
klm_ftrace already loaded. skip.
12.63: card_index: 21211, card_type: 129
Loading I2C driver ... done.
Installing CCTRL driver for card_type 129 in vslot ID 26 without NEED_GEM ...
 done.
Loading IGB driver ...
Checking all filesystems.
done.
Extracting rpms from image...
Installing SPROM driver ... 21211 IS_N9K done.
Installing pfmsvcs module ...done.
Installing nvram module ... done.
Installing if_index module with port mode 6 ... done.
Installing fcfwd
Installing RNI lcnd ... done
Running mode is N9K-ToR ...done
Installing LC netdev container_id=0 is_standby=0 ... done
Installing psdev module ... done.
Installing veobc module ... done.
Clean up previous pcap files present in tmp directory
Checking SR card
Card Index is 21211
SR Card detected.
Inserting eMMC module ...
Inserting OBFL module ... 23.93: eMMC Device found.
23.93: card has mmc
done.
23.94: Making OBFL character devices for mmc
grep: /bootflash/platform.inf: No such file or directory
                                                          23.98: SR_START
(9635840), size 7
mounting plog for N9k!
23.99: mounting plog for N9k!
24.00: Done..mounting plog for N9k!
Mounting OBFL for eMMC
Attach blkoops
24.49: Before mounting blkoops
24.50: After mounting blkoops
24.51: recreate_mmcftrace_rr_device
/
Calling install_tor for TOR
copying common rpm to ftp
COPY the image to ORIG_LC_FS_PATH


tune2fs 1.46.5 (30-Dec-2021)
Setting reserved blocks percentage to 0% (0 blocks)
Adding system startup for /etc/init.d/cgroups-init.
Persistent log store is mounted successfully
Loading system software
kernel.core_pattern = |/bin/sparsecore.sh %p %h %P %E
Starting rpcbind daemon...done.
Generating ssh keys...creating NFS state directory: done
starting 8 nfsd kernel threads: done
starting mountd: done
starting statd: done
Saving image for img-sync ...
exit code: 1
```

```
Done ssh keys
Starting OpenBSD Secure Shell server: sshd ... done sshd.
Installing local RPMS
Patch Repository Setup completed successfully
/logflash/stats_ssd/.ssd_overall_data file exist !!
moving /logflash/stats_ssd/.ssd_overall_data to
/logflash/stats_ssd/.ssd_lastboot_data
Starting crond: OK
Creating /dev/mcelog
Starting mcelog daemon
Stopping crond: OK
N9K CPA setup done
N9K TOR LC CPA setup done
Remove pltfm_config from eltm.conf for SONE cards
INIT: Entering runlevel: 3
Replacement complete. All system shutdown binaries now point to /sbin/rep_file.
Running S93thirdparty-script...
vlc_card_index 0
S26: sw_id 21211 vp 0
S26 B: sw_id 21211, Moving libcrdcfgdata_tah_tor.so to libcrdcfgdata.so
Starting crond: OK


Starting lem_drv driver
Netbroker support IS present in the kernel.
done
done
Disabling cisco netbroker hooks ... done
Skipping EMON in TOR ...
Executing Prune clis.
Removing exusd cli files if it is not exbl_card in prune_clis
ethernet switching mode Tue Mar 24 11:01:59 PDT 2026
prepare span CLI
2026 Mar 24 11:02:40 switch %$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 1 powered
 up (Serial number FDO25210LJX)


2026 Mar 24 11:02:42 switch %$ VDC-1 %$ %VDC_MGR-2-VDC_ONLINE: vdc 1 has come
 online

User Access Verification
switch login: 2026 Mar 24 11:05:14 switch %$ VDC-1 %$
%ASCII-CFG-2-CONFIG_REPLAY_STATUS: Ascii Replay Started.
2026 Mar 24 11:05:22 switch %$ VDC-1 %$ %ASCII-CFG-2-CONFIG_REPLAY_STATUS:
Ascii Replay Done.
2026 Mar 24 11:05:23 switch %$ VDC-1 %$ %ASCII-CFG-2-CONF_CONTROL: System ready
2026 Mar 24 11:05:49 switch %$ VDC-1 %$ %VMAN-2-GUESTSHELL_ENABLED: The guest
 shell has been enabled. The command 'guestshell' may be used to access it,
'guestshell destroy' to remove it.

switch login:
```

Non-Disruptive ISSU CloudScale TOR

```
========================
switch# show module
Mod Ports                  Module-Type                              Model
     Status
--- ----- ------------------------------------------------ ----------------------
 --------
1    108  96x10/25/32G + 12x40/100G Ethernet/FC Module     N9K-C93360YC-FX2
```

```
    ok
27   0    Virtual Supervisor Module                        N9K-C93360YC-FX2
    active *

Mod  Sw                      Hw    Slot
---  --------------------- ------ ----
1    10.6(2)                 1.0    NA
27   10.6(2)                 1.0    VSUP


Mod  MAC-Address(es)                          Serial-Num
---  ---------------------------------------  ----------
1    70-ea-1a-d1-69-fa to 70-ea-1a-d1-6a-a5  FDO23240P95
27   70-ea-1a-d1-69-fa to 70-ea-1a-d1-6a-a5  FDO23240P95

Mod  Online Diag Status
---  ------------------
1    Pass
27   Pass

switch# install all nxos bootflash:nxos64-cs.10.6.2.F.bin non-disruptive
Installer will perform compatibility check first. Please wait.

Verifying image bootflash:/nxos64-cs.10.6.2.F.bin for boot variable "nxos".
[####################] 100% -- SUCCESS

Verifying EPLD/FPGA image //bootflash/nxos64-cs.10.6.2.F.bin.
[####################] 100% -- SUCCESS

Verifying image type.
[####################] 100% -- SUCCESS

Preparing "nxos" version info using image bootflash:/nxos64-cs.10.6.2.F.bin.
[####################] 100% -- SUCCESS

Preparing "bios" version info using image bootflash:/nxos64-cs.10.6.2.F.bin.
[####################] 100% -- SUCCESS

Performing module support checks.
[####################] 100% -- SUCCESS

Notifying services about system upgrade.
[####################] 100% -- SUCCESS



Compatibility check is done:
Module  bootable          Impact  Install-type  Reason
------  --------  --------------  ------------  ------
     1       yes  non-disruptive       rolling
    27       yes  non-disruptive         reset



Images will be upgraded according to following table:
Module       Image                 Running-Version(pri:alt)
New-Version  Upg-Required
------  ----------  ----------------------------------------
--------------------  ------------
     1       lcn9k                                    10.6(1)
10.6(2)           yes
    27        nxos                                    10.6(1)
```

```
10.6(2)           yes
    27        bios     v05.53(01/22/2025):v05.38(06/12/2019)
v05.53(01/22/2025)            no


FPGA microcode will be upgraded according to following table:
Module  Type  EPLD             Running-Version  Flashed-Version*   New-Version
  Upg-Required
------  ----  -------------     ---------------  ----------------   -----------
  ------------
    27   SUP  MI FPGA            0x4               0x4                0x4
              No
    27   SUP  IO FPGA            0x16              0x16               0x16
              No
    27   SUP  MI FPGA2           0x3               0x3                0x3
              No
* If Running-Version and Flashed-Version are different it implies that the
system has not yet been reloaded for the new version to take effect


Additional info for this installation:
--------------------------------------

Service "vpc" in vdc 1: Vpc is enabled, Please make sure both Vpc peer switches
 have same boot mode using 'show boot mode' and proceed


Service "ptp" in vdc 1: During ISSU PTP protocol state and time synchronization
 will be disturbed. Time synchronization will resume after ISSU


Do you want to continue with the installation (y/n)?  [n] y


Install is in progress, please wait.
[#                   ]   0%
Notifying services about the upgrade.

Setting boot variables.
[#                   ]   0%
Setting boot variables.
[####################] 100% -- SUCCESS

Performing configuration copy.
[####################] 100%
Performing configuration copy.
[####################] 100% -- SUCCESS

Module 1: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#                   ]   0%
Module 1: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

Module 27: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#                   ]   0%
Module 27: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS
```

```
[#                    ]   0%
[####################] 100% -- SUCCESS

All EPLD/FPGAs are up to date
Upgrade can no longer be aborted, any failure will result in a disruptive
upgrade
[#                    ]   0%
Freeing memory in the file system.

Loading images into memory.
[##                   ]   5%
Loading images into memory.
[####################] 100% -- SUCCESS

Saving linecard runtime state.

Saving linecard runtime state.

Saving supervisor runtime state.

Saving supervisor runtime state.

Saving mts state.

Saving mts state.


Rebooting the switch to proceed with the upgrade.
All telnet and ssh connections will now be temporarily terminated.
switch# incrond[28730]: stopping service
63+1 records in
63+1 records out
265322496 bytes (265 MB, 253 MiB) copied, 0.178732 s, 1.5 GB/s
mount: overlay mounted on /newroot/usr.
Installing klm_card_index
done
Increase FPGA PCI timeout to 900ms
Linking n9k flash devices
creating flash devices BOOT_DEV= sda
Replacing inittab for node - host
INIT: version 3.01 booting
Setting up sysV scripts for host node
Installing ata_piix module ... done.
Unsquashing rootfs ...
is_lxc:1 is_titan_eor: is_stby: suffix: klm_ftrace:
/isanboot/lib/modules/klm_ftrace.o
Installing ftrace in lxc 0 mode done

Checking all filesystems.cgroups initialized
Generating ssh keys...Done ssh keys
Starting OpenBSD Secure Shell server: sshd ... done sshd.
Checking rpmstore swap in host context.
tune2fs 1.46.5 (30-Dec-2021)
Setting reserved blocks percentage to 0% (0 blocks)
Installing kheaders module ... done.
INIT: Entering runlevel: 3
cp: cannot stat '/orig_fs/xr_cpa.img': No such file or directory
grep: /bootflash/platform.inf: No such file or directory


mount: overlay mounted on /usr.
Increase FPGA PCI timeout to 900ms
```

```
Linking n9k flash devices
creating flash devices BOOT_DEV= sda
Replacing inittab for node - vsup
INIT: version 3.01 booting
Unsquashing rootfs ...
vm.min_free_kbytes = 138240
No TOR virtual chassis
skip untra_klms
Enabling 8250 serial driver spurious INTs workaround
Installing isan procfs isan_proc_name=isan26... done.
Installing SSE module with card index 21172 and vslot ID 26 ... done.
Creating SSE device node 241 ... done.
cg controllers already mounted at /var/cgroup
Inserting kernel_services module ... done.
Making kernel_services character devices
is_lxc:1 is_titan_eor: is_stby:0 suffix: klm_ftrace:
/isanboot/lib/modules/klm_ftrace.o
klm_ftrace already loaded. skip.
19.70: card_index: 21172, card_type: 103
Loading I2C driver ... done.
Installing CCTRL driver for card_type 103 in vslot ID 26 without NEED_GEM ...
 done.
Loading IGB driver ...
Checking all filesystems.done.

Installing SPROM driver ... 21172 Quick rpm extraction...
IS_N9K done.
Segment 6 already extracted! Continuing...
Installing pfmsvcs module ...done.
Installing nvram module ... done.
Installing if_index module with port mode 6 ... done.
Calling install_tor for TOR
copying common rpm to ftp
Installing fcfwd
Installing RNI lcnd ... done
Running mode is N9K-ToR ...done
Installing LC netdev container_id=0 is_standby=0 ... done
Installing psdev module ... done.
Installing veobc module ... done.
Clean up previous pcap files present in tmp directory
Checking SR card
Card Index is 21172
SR Card detected.
Inserting eMMC module ...
Inserting OBFL module ... 24.38: eMMC Device found.
24.38: card has mmc
done.
24.39: Making OBFL character devices for mmc
24.45: SR_START (2099200), size 3
mounting plog for N9k!
24.46: mounting plog for N9k!
24.49: Done..mounting plog for N9k!
Mounting OBFL for eMMC
grep: /bootflash/platform.inf: No such file or directory
                                                        Attach blkoops
25.03: Before mounting blkoops
25.81: After mounting blkoops
26.45: recreate_mmcftrace_rr_device
COPY the image to ORIG_LC_FS_PATH


 Adding system startup for /etc/init.d/cgroups-init.
```

```
tune2fs 1.46.5 (30-Dec-2021)
Setting reserved blocks percentage to 0% (0 blocks)
 KEXEC LXC ISSU bypass
Persistent log store is mounted successfully
Loading system software
kernel.core_pattern = |/bin/sparsecore.sh %p %h %P %E
 minicoredumper disabled for this platform...
Starting rpcbind daemon...pidof[8519]: can't read from 8511/stat

done.
Generating ssh keys...creating NFS state directory: done
starting 8 nfsd kernel threads: done
starting mountd: done
starting statd: done
Saving image for img-sync ...
exit code: 1
Installing local RPMS
Done ssh keys
Starting OpenBSD Secure Shell server: sshd ... done sshd.
Patch Repository Setup completed successfully
/logflash/stats_ssd/.ssd_overall_data file exist !!
moving /logflash/stats_ssd/.ssd_overall_data to
/logflash/stats_ssd/.ssd_lastboot_data
Starting crond: OK
TOR ISSU is TRUE. Touching /isan-upgrade/isan.bin at Tue Mar 24 10:11:08 PDT
2026
TOR ISSU is TRUE. /mnt/pss/upgrade/ present at Tue Mar 24 10:11:08 PDT 2026
touched /share/vmm/lxc_fallback at Tue Mar 24 10:11:08 PDT 2026
Creating /dev/mcelog
Starting mcelog daemon
Stopping crond: OK
fastboot or torissu
Starting crond: OK
N9K CPA setup done
N9K TOR LC CPA setup done
Remove pltfm_config from eltm.conf for SONE cards
INIT: Entering runlevel: 3
Replacement complete. All system shutdown binaries now point to /sbin/rep_file.
Running S93thirdparty-script...
vlc_card_index 0
S26: sw_id 21172 vp 0
S26 B: sw_id 21172, Moving libcrdcfgdata_tah_tor.so to libcrdcfgdata.so
packagegroup-nxos-optional-rpms_pkg_list


Starting lem_drv driver
done
Netbroker support IS present in the kernel.
done
Disabling cisco netbroker hooks ... done
Skipping EMON in TOR ...
Executing Prune clis.
Removing exusd cli files if it is not exbl_card in prune_clis
ethernet switching mode Tue Mar 24 10:11:17 PDT 2026
prepare span CLI

System is coming up ... Please wait ...
2026 Mar 24 10:12:28 switch %$ VDC-1 %$ %NVE-5-VDC_MAC_G_STATUS: Attempting
to get router-mac, current value N/A
2026 Mar 24 10:12:28 switch %$ VDC-1 %$ %ASCII-CFG-2-CONFIG_REPLAY_STATUS:
Bootstrap Replay Started.
2026 Mar 24 10:12:28 switch %$ VDC-1 %$ %USER-0-SYSTEM_MSG: Recover l3vm state
```

```
 - l3vm
2026 Mar 24 10:12:28 switch %$ VDC-1 %$ %ASCII-CFG-2-CONFIG_REPLAY_STATUS:
Bootstrap Replay Done.
2026 Mar 24 10:12:28 switch %$ VDC-1 %$ %NETSTACK-0-IP_EMERG_LOG:  netstack
[18083]  Recovery restart: Stateful Restart: init_iftype, Reset im_ext share
memory intact FALSE
2026 Mar 24 10:12:31 switch %$ VDC-1 %$ %NETSTACK-0-IP_EMERG_LOG:  netstack
[18083]  Recovery restart: ip_ha_recover_if_restart: ip_recover_iftype_all
.

2026 Mar 24 10:12:34 switch %$ VDC-1 %$ %NETSTACK-0-IP_EMERG_LOG:  netstack
[18083]  Registration with clis server complete
Continuing with installation process, please wait.
The login will be disabled until the installation is completed.

Status for linecard upgrade.

[####################] 100% -- SUCCESS

Performing supervisor state verification.

[####################] 100% -- SUCCESS

Supervisor non-disruptive upgrade successful


Install has been successful.

User Access Verification


Login incorrect
User Access Verification
switch login: admin
Password:
```

Min-Disruptive ISSU Silicon One TOR

```
======================

switch# install all nxos bootflash:nxos64-s1.10.6.2.F.bin.upg.tmp min-disruptive
 non-interruptive
Installer will perform compatibility check first. Please wait.

Verifying image bootflash:/nxos64-s1.10.6.2.F.bin.upg.tmp for boot variable
"nxos".
[####################] 100% -- SUCCESS

Verifying EPLD/FPGA image //bootflash/nxos64-s1.10.6.2.F.bin.upg.tmp.
[####################] 100% -- SUCCESS

Verifying image type.
[####################] 100% -- SUCCESS

Preparing "nxos" version info using image
bootflash:/nxos64-s1.10.6.2.F.bin.upg.tmp.
[####################] 100% -- SUCCESS

Preparing "bios" version info using image
bootflash:/nxos64-s1.10.6.2.F.bin.upg.tmp.
[####################] 100% -- SUCCESS
```

```
Performing module support checks.
[####################] 100% -- SUCCESS

Notifying services about system upgrade.
[####################] 100% -- SUCCESS



Compatibility check is done:
Module  bootable          Impact  Install-type  Reason
------  --------  --------------  ------------  ------
     1       yes  min-disruptive       rolling
    27       yes  min-disruptive         reset



Images will be upgraded according to following table:
Module       Image                 Running-Version(pri:alt)
New-Version  Upg-Required
------  ----------  ---------------------------------------
--------------------  ------------
     1       lcn9k                                    10.6(3)
10.6(3u)I9(1u)           yes
    27        nxos                                    10.6(3)
10.6(3u)I9(1u)           yes
    27        bios     v01.10(03/06/2025):v01.08(01/22/2025)
v01.10(03/06/2025)            no


FPGA microcode will be upgraded according to following table:
Module  Type  EPLD             Running-Version  Flashed-Version*   New-Version
  Upg-Required
------  ----  -------------     ---------------  ----------------   -----------
  ------------
    27   SUP  MI FPGA            0x1000a           0x1000a           0x1000a
              No
    27   SUP  IO FPGA            0x1000d           0x1000d           0x1000d
              No
    27   SUP  IO FPGA(TAM)       0x70016           0x70016           0x70016
              No
    27   SUP  IO FPGA(FB)        0x1000d           0x1000d           0x1000d
              No
* If Running-Version and Flashed-Version are different it implies that the
system has not yet been reloaded for the new version to take effect


Additional info for this installation:
--------------------------------------

Estimated traffic outage time  for current upgrade :  about 65 ms



Install is in progress, please wait.
[#                   ]   0%
Notifying services about the upgrade.

Setting boot variables.
[#                   ]   0%
Setting boot variables.
```

```
[####################] 100% -- SUCCESS

Performing configuration copy.
[####################] 100%
Performing configuration copy.
[####################] 100% -- SUCCESS

Module 1: Upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#                   ]   0%
Module 1: Upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

Module 27: Upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#                   ]   0%
Module 27: Upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[####################] 100% -- SUCCESS

[#                   ]   0%
[####################] 100% -- SUCCESS

All EPLD/FPGAs are up to date
Upgrade can no longer be aborted, any failure will result in a disruptive
upgrade
[#                   ]   0%
Freeing memory in the file system.

Loading images into memory.
[##                  ]   5%
Loading images into memory.
[####################] 100% -- SUCCESS

Saving linecard runtime state.

Saving linecard runtime state.

Saving supervisor runtime state.

Saving supervisor runtime state.

Saving mts state.

Saving mts state.


Rebooting the switch to proceed with the upgrade.
All telnet and ssh connections will now be temporarily terminated.
switch# incrond[18252]: stopping service
71+1 records in
71+1 records out
300548096 bytes (301 MB, 287 MiB) copied, 0.155314 s, 1.9 GB/s
mount: overlay mounted on /newroot/usr.
Installing klm_card_index
done
Linking n9k mando SSD partition...
creating flash devices BOOT_DEV= sda
Replacing inittab for node - host
INIT: version 3.04 booting
Setting up sysV scripts for host node
```

```
Installing ata_piix module ... done.
Unsquashing rootfs ...
is_lxc:1 is_titan_eor: is_stby: suffix: klm_ftrace:
/isanboot/lib/modules/klm_ftrace.o
Installing ftrace in lxc 0 mode done
        Model Number:       Micron_5400_MTFDDAV480TGA
        Serial Number:      MSY28450C90
        Firmware Revision:  D4CN005

Checking all filesystems.cgroups initialized
System is at the expected SSD partitioning format
Generating ssh keys...Done ssh keys
Starting OpenBSD Secure Shell server: sshd ... done sshd.
Checking rpmstore swap in host context.
tune2fs 1.47.0 (5-Feb-2023)
Setting reserved blocks percentage to 0% (0 blocks)
Installing kheaders module ... done.
INIT: Entering runlevel: 3


mount: overlay mounted on /usr.
Linking n9k mando SSD partition...
creating flash devices BOOT_DEV= sda
Replacing inittab for node - vsup
INIT: version 3.04 booting
Unsquashing rootfs ...
No TOR virtual chassis
skip untra_klms
Enabling 8250 serial driver spurious INTs workaround
Installing isan procfs isan_proc_name=isan26... done.
Installing SSE module with card index 21267 and vslot ID 26 ... done.
Creating SSE device node 240 ... done.
cg controllers already mounted at /var/cgroup
Inserting kernel_services module ... done.
Making kernel_services character devices
is_lxc:1 is_titan_eor: is_stby:0 suffix: klm_ftrace:
/isanboot/lib/modules/klm_ftrace.o
klm_ftrace already loaded. skip.
        Model Number:       Micron_5400_MTFDDAV480TGA
        Serial Number:      MSY28450C90
        Firmware Revision:  D4CN005
26.81: card_index: 21267, card_type: 152
Loading I2C driver ... done.
Installing CCTRL driver for card_type 152 in vslot ID 26 without NEED_GEM ...
 done.
26.88: Interrupt throttling disabled. No cctrl irq detected.
Loading IGB driver ...
Checking all filesystems.
Quick rpm extraction...
done.
Segment 6 already extracted! Continuing...
Segment 7 already extracted! Continuing...
Modify INotify fd
128
fs.inotify.max_user_instances = 2048
2048
Installing SPROM driver ... 21267 IS_N9K done.
Installing pfmsvcs module ...done.
Installing nvram module ... done.
Installing if_index module with port mode 6 ... done.
Calling install_tor for TOR
copying common rpm to ftp
```

```
Installing fcfwd
Installing RNI lcnd ... done
Running mode is N9K-ToR ...done
Installing LC netdev container_id=0 is_standby=0 ... done
Installing psdev module ... done.
Installing veobc module ... done.
Clean up previous pcap files present in tmp directory
Checking SR card
Card Index is 21267
old data: 9f000000 new data: 9f000000
old data: 0 new data: 0
old data: a0901406 new data: a0901406
33.23: Inserting obfl & mtd spi driver ...
[   33.245002] fpga_setup_probe
[   33.248280] Probe called for devide 0x17a
33.32: Done..Inserting obfl & mtd spi driver ...
Making OBFL character devices
old data: 9f000000 new data: 9f000000
old data: 0 new data: 0
old data: a0901406 new data: a0901406
mounting plog for N9k!
33.38: mounting plog for N9k!
33.42: Done..mounting plog for N9k!
mounting recovery for N9k!
33.44: mounting recovery for N9k!
Mounting OBFL pstore for mtd
System is at the expected SSD partitioning format
COPY the image to ORIG_LC_FS_PATH
/isanboot/bin/images/s1_gr2_xc_sdk.bin
/isanboot/bin/images/s1hal_n9000.bin
tune2fs 1.47.0 (5-Feb-2023)
Setting reserved blocks percentage to 0% (0 blocks)
KEXEC LXC ISSU bypass
Persistent log store is mounted successfully
Loading system software
kernel.core_pattern = |/bin/sparsecore.sh %p %h %P %E
minicoredumper disabled for this platform...
Starting rpcbind daemon...Generating ssh keys...done.
Saving image for img-sync ...
creating NFS state directory: done
starting 8 nfsd kernel threads: done
starting mountd: done
starting statd: done
exit code: 1
Installing local RPMS
Done ssh keys
Starting OpenBSD Secure Shell server: sshd ... done sshd.
Patch Repository Setup completed successfully
/logflash/stats_ssd/.ssd_overall_data file exist !!
moving /logflash/stats_ssd/.ssd_overall_data to
/logflash/stats_ssd/.ssd_lastboot_data
Starting crond: OK
TOR ISSU is TRUE. Touching /isan-upgrade/isan.bin at Thu Mar 19 09:47:17 PDT
2026
TOR ISSU is TRUE. /mnt/pss/upgrade/ present at Thu Mar 19 09:47:17 PDT 2026
touched /share/vmm/lxc_fallback at Thu Mar 19 09:47:17 PDT 2026
Creating /dev/mcelog
Starting mcelog daemon
Stopping crond: OK
Remove pltfm_config from eltm.conf for SONE cards
INIT: Entering runlevel: 3
Replacement complete. All system shutdown binaries now point to /sbin/rep_file.
```

```
Starting crond: OK
Deleting /usr/bin/xr_cpa
Deleting /usr/lib64/xr_cpa
Deleting /usr/local/etc/xr-cpa-data
Deleting /usr/lib64/dummy_libs
Deleting /usr/lib64/r78x_27.tar.gz
Run dc3_sensor as root
ASIC_GIBRALTAR is
ASIC_IRON is
Running S93thirdparty-script...
vlc_card_index 0
S26: sw_id 21267 vp 0
S26 B: sw_id 21267, Retaining libcrdcfgdata.so as is
packagegroup-nxos-optional-rpms_pkg_list


Starting lem_drv driver
Netbroker support IS present in the kernel.
done
done
Disabling cisco netbroker hooks ... done
Skipping EMON in TOR ...
Executing Prune clis.
Removing exusd cli files if it is not exbl_card in prune_clis
ethernet switching mode Thu Mar 19 09:47:24 PDT 2026
prepare span CLI


Continuing with installation process, please wait.
The login will be disabled until the installation is completed.

Status for linecard upgrade.
[####################] 100% -- SUCCESS

Performing supervisor state verification.
[####################] 100% -- SUCCESS

Supervisor min-disruptive upgrade successful


Install has been successful.

User Access Verification
switch login: admin
Password:

Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright (C) 2002-2026, Cisco and/or its affiliates.
All rights reserved.
```

Prerequisites for NX-OS software downgrade

This section provides an overview of the prerequisites for downgrading the NX-OS software, including verifying feature compatibility, addressing any incompatibilities, and managing active configuration sessions before initiating the downgrade process.

The prerequisites for downgrading the NX-OS software include

- Before downgrading from a NX-OS release that supports the Control Plane Policing (CoPP) feature to an earlier release that does not support the CoPP feature, verify compatibility using the **show incompatibility nxos bootflash:** *filename* command. If incompatibility exists, disable any features that are incompatible with the downgrade image before downgrading the software.
- Save, commit, or discard any active configuration sessions before downgrading the NX-OS software image on your device.

NX-OS software downgrade guidelines

This section lists the guidelines and limitations you must review before downgrading NX-OS software, including supported downgrade methods, feature-specific considerations, and platform restrictions to ensure a successful and supported downgrade process.

Read and follow these guidelines and limitations listed in this section before downgrading your NX-OS software from the current release to an earlier release.

- The only supported method of downgrading a Nexus 9000 Series switch is to utilize the **install all** command. Changing the boot variables, saving the configuration, and reloading the switch is not a supported method to downgrade the switch.

  Disable the Guest Shell if you need to downgrade from NX-OS Release 9.3(x) to an earlier release.

  - Performing an ISSU downgrade from NX-OS Release 9.3(x) to Release 7.0(3)I4(1) with an FCoE (Fiber Channel over Ethernet) NPV (N-port Virtualization) configuration causes the port channel to crash with a core file:

    ```
    [################ ] 38%2016 Apr 18 20:52:35 n93-ns1 %$ VDC-1 %$ %SYSMGR-2-
     SERVICE_CRASHED: Service "port-channel" (PID 14976) hasn't caught signal
     11 (core will be saved)
    ```

  - ISSU (non-disruptive) downgrade is not supported

- On Nexus 9500 switches with N9508-E2 Fabric module, downgrade from any 9.x or 10.x supported releases to any unsupported releases of 7.x is not supported.
- When downgrading from the NX-OS Release 9.3(x) to earlier releases, any ACL with the statistics per-entry command enabled and applied as RACL needs the statistics per-entry command removed from the running configuration before downgrading. Otherwise, the interfaces on which this ACL is applied as a RACL will be error disabled after the downgrade.
- Prior to downgrading a Nexus 9500-series switch, with -FX or -FX+EX line cards, from NX-OS Release 10.1(x) to earlier releases (9.2(x) or 7.x), the TCAM region that applies to NetFlow (ing-netflow) should be carved to zero (0) using the following command:

  hardware access-list tcam region ing-netflow 0

  The configuration change is required because the default ing-netflow TCAM region in 9.3(1) and onwards is 512 while the default in 9.2(x) and earlier is 0.

- When downgrading from the NX-OS Release 10.1(x) to a release prior to 9.3(x), make sure that the ACL TCAM usage for ingress features does exceed the allocated TCAM space in the absence of the label sharing feature. Label sharing is a new feature in NX-OS Release 9.3(x). Otherwise, interfaces with RACLs that could not fit in the TCAM will be disabled after the downgrade.
- Software downgrades should be performed using the **install all** command. Changing the boot variables, saving the configuration, and reloading the switch is not a supported method to downgrade the switch.
- This limitation applies to Nexus platform switches that support Trust Anchor Module (TAM):

The TACACS global key cannot be restored when downgrading from NX-OS Release 9.3(3) and higher to any earlier version. TAM was updated to version-7 in 9.3(3), but earlier NX-OS versions used TAM version-3.

- iCAM must be disabled before downgrading from Release 9.2(x) or Release 9.3(x) → 7.0(3)I7(1). Only Release 9.3(1) → Release 9.2(4) can be performed if iCAM is enabled.
- Beginning with NX-OS Release 9.3(3), new configuration commands exist for SRAPP (with sub-mode options for MPLS and SRTE). The SRAPP configuration on the switch running release 9.3(3) (or later) will not be present if the switch is downgraded to an earlier release.
- On devices with dual supervisor modules, both supervisor modules must have connections on the console ports to maintain connectivity when switchovers occur during a software downgrade. For more information about your specific chassis, see the relevant Hardware Installation Guide.
- NX-OS automatically installs and enables the guest shell by default. However, if the device is reloaded with a NX-OS image that does not provide guest shell support, the existing guest shell is automatically removed and a %VMAN-2-INVALID_PACKAGE message is issued. As a best practice, remove the guest shell with the **guestshell destroy** command before downgrading to an earlier NX-OS image.
- You must delete the switch profile (if configured) when downgrading from a NX-OS release that supports switch profiles to a release that does not. For more information, see the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide.
- Software downgrades are disruptive. In-service software downgrades (ISSDs), also known as non-disruptive downgrades, are not supported.
- While downgrading from the NX-OS Release 10.2(1)F or higher to an earlier release, the **install all** command is blocked when the delay config is present in track list Boolean/weight.
- While performing ISSD from NX-OS Release 10.2(3)F to NX-OS Release 10.2(2)F with **epbr L2** applied on interfaces, remove the policies from interfaces before performing ISSD to avoid the duplicate tracks issue.
- Beginning with NX-OS Release 10.2(3)F, if you have configured the **lldp chassis-id switch** command, then you must disable the command before performing ISSD.
- Beginning with 10.2(3)F, although application of ePBR policy to access ports is supported, downgrading with this configuration is not recommended.
- When feature ngmvpn is enabled and a disruptive downgrade is performed from NX-OS Release 10.3(2)F to NX-OS Release 10.3(1)F, although a few VRFs are missing from the show run output, this is only a display issue, and has no functional impact.
- When a switch is downgraded from NX-OS Release 10.3(3)F or 10.4(1)F to a version that supports both Native and LXC modes, the switch always goes to Native mode even if the upgrade was done from LXC mode. To keep the mode persistent after a downgrade, ensure that you perform the downgrade in the following sequence:

  Note

  - The following sections are applicable only to Nexus 9300-FX3 and 9300-GX platform switches.
  - When system comes up in native mode on downgrade, boot mode lxc is removed from configuration.

  - LXC mode upgrade/downgrade: For example,
    1. The switch is running on NX-OS Release 10.3(2)F in LXC mode.
    2. Upgrade the version to NX-OS Release 10.3(3)F (LXC mode).

3. Downgrade the version to NX-OS Release 10.3(2)F to the Native mode.
4. Execute the boot mode lxc configuration command, save the configuration, and reload the switch.
5. The switch comes up in NX-OS Release 10.3(2)F LXC mode.

- Native mode upgrade/downgrade:

  Example 1

  1. The switch (9300-FX3 or 9300-GX) is running on NX-OS Release 10.3(2)F in the Native mode.
  2. Upgrade the version to NX-OS Release 10.3(3)F (LXC mode), as these (9300-FX3 or 9300-GX) switches support only LXC mode.
  3. Downgrade to any earlier NX-OS Release [for example, 10.3(2)].
  4. The switch comes up in NX-OS Release 10.3(2)F in Native mode.

  Example 2

  1. The switch (Nexus 9300-FX or 9300-FX2) is running on NX-OS Release 10.3(2)F in the Native mode.
  2. Upgrade the version to NX-OS Release 10.4(1)F (LXC mode), as these switches support only LXC mode.
  3. Downgrade to any earlier NX-OS Release [for example, 10.3(2)].
  4. The switch comes up in NX-OS Release 10.3(2)F in Native mode.

- When you downgrade from NX-OS Release 10.4(2)F to any earlier releases until 10.3(2)F (included), N9K-C9400-SW-GX2A Sup card ports 2xSFP Eth10/1-2 are not supported.
- When Nexus 9400 has feature security-group enabled, then downgrade from 10.5(1)F release to a lower release is not possible as feature security-group is not supported on 9400 in 10.4(3)F release and earlier.
- Downgrade of Nexus 9800 switches that have s1 image format from NX-OS Release 10.5(1)F to earlier releases that have cs image format is supported. However, as the s1 image is not present in earlier releases, the downgrade should be done to the cs image from the corresponding release.
- If you are on NX-OS Release 10.5(2)F, where DSVNI with IPv6 underlay feature is supported, and downgrade to a release that does not support this feature, then this leads to traffic loss for DSVNI VLANs. To avoid this, ensure that you remove the IPv6 underlay with downstream VNI configuration before downgrading.
- During downgrade, where both source and target images support Type-6 encryption, while performing device reload, if ASCII replay is triggered without binary restore, primary key gets lost. The primary key must be reconfigured after device reload. Use the **key config-key ascii** command to reconfigure the primary key and avoid encryption issues. However, downgrade with binary restore retains the primary key after the reboot, provided both source and target images support Type-6 encryption.

  If you downgrade the system from an image that supports Type-6 encryption to an image that does not support Type-6 encryption, compatibility check fails.
- When a switch is downgraded from NX-OS Release 10.6(1)F to any lower releases, the **allow feature-set fex** configuration gets added to the running configuration file.

Downgrade to an earlier software release

Use this procedure to safely downgrade a Nexus 9000 switch to an earlier NX-OS release by verifying software and hardware compatibility, copying the correct image file, disabling unsupported features, saving configurations, and

performing the downgrade using the install all command. Always review the release notes for any specific exceptions or prerequisites before starting the process.

The downgrade of Nexus 9000 switches involves using the supported **install all** command to revert to an earlier NX-OS software release, ensuring compatibility by checking for software and hardware incompatibilities, disabling unsupported features, and following a structured procedure including saving configurations and reloading the switch.

Downgrade on Nexus switches is always disruptive.

> **Note**
>
> Downgrade from releases 10.5(1)F, 10.5(2)F, or 10.5(3)F to 10.4(6)F and later 10.4(x) releases are not supported and can result in configuration corruption. If downgrade is needed, upgrade first to 10.5(4)M or later releases and then downgrade to 10.4(6)F and later 10.4(x) releases. See CSCwr21007 in the Cisco Nexus 9000 Series NX-OS Release Notes, Release 10.4(6)M.

> **Note**
>
> If an error message appears during the downgrade, the downgrade fails due to the indicated reason. See the appropriate version of the Cisco Nexus 9000 Series NX-OS Troubleshooting Guide for a list of possible causes and solutions.

Before you begin

Read the release notes for the software image file for any exceptions to this downgrade procedure. For information about software image file, see the appropriate version of the Cisco Nexus 9000 Series NX-OS Release Notes.

Procedure

1. Log in to the device on the console port connection.
2. Verify the software image file and copy it to the active supervisor module.
   a) Verify that the image file for the downgrade is present on the active supervisor module bootflash.

      Example

      ```
      switch# dir bootflash:
      ```

      If the software image file is not present, log in and choose the software image file for your device from the http://software.cisco.com/download/navigator.html URL, and download it to a file server.

      > **Note**
      >
      > If you need more space on the active or standby supervisor module bootflash, use the **delete** command to remove unnecessary files.

   b) Copy the software image to the active supervisor module using a transfer protocol. You can use FTP, TFTP, SCP, or SFTP.

Example

```
switch# copy scp://user@scpserver.cisco.com//download/nxos.9.2.1.bin
bootflash:nxos.9.2.1.bin
```

3. Check for any incompatibilities.
   a) Check for any software incompatibilities.

   Example

   ```
   switch# show incompatibility-all nxos bootflash:nxos.9.2.1.bin
   Checking incompatible configuration(s)
   No incompatible configurations
   ```

   The resulting output displays any incompatibilities and remedies.

   > **Note**
   >
   > Incompatibility check does not flag if the configuration Layer 2 access ports configured on the switch when downgrading to older releases.

   b) Check for any hardware incompatibilities.

   Example

   ```
   switch# show install all impact nxos bootflash:nxos.9.2.1.bin
   ```

4. Mitigate incompatibilities.
   a) Disable any features that are incompatible with the downgrade image.
   b) Power off any unsupported modules.

   Example

   ```
   switch# poweroff module module-number
   ```

5. Save the running configuration to the startup configuration.

   Example

   ```
   switch# copy running-config startup-config
   ```

6. Downgrade the NX-OS software using the **install all nxos bootflash** *<nxos_image_to_downgrade>* command.

> **Note**
>
> If you enter the **install all** command without specifying a filename, the command performs a compatibility check, notifies you of the modules that will be upgraded, and confirms that you want to continue with the installation. If you choose to proceed, it installs the NXOS software image that is currently running on the switch and upgrades the BIOS of various modules from the running image if required.

7. Optional: Display the entire downgrade process.

   Example

   ```
   switch# show install all status
   ```

8. Optional: Log in and verify that the device is running the required software version.

   Example

   ```
   switch# show version
   ```

3 Transceiver Firmware

Topics:

- Transceiver firmware software
- Transceiver firmware software upgrades
- Firmware upgrade using NX-OS commands
- Parallel firmware download

Describes how the transceiver firmware or embedded software updates for N9000 optical modules are designed to optimize performance, address security vulnerabilities, and maintain compliance with current industry standards.

Transceiver firmware software

Describes a transceiver firmware with regard to N9000 switches.

A transceiver firmware in N9000 switches is a software that

- is embedded within the optical transceiver modules (optics) installed in the switch interfaces
- controls the operation of the optics, including signal transmission, reception, diagnostics, and performance optimization, and
- is specific to the transceiver model.

Beginning with NX-OS Release 10.6(2)F, transceiver firmware upgrade using NX-OS commands is supported and explained in this chapter.

Transceiver firmware software upgrades

Provides an introduction about transceiver firmware software upgrades and how this upgrade improves performance, addresses issues, ensures compliance, and supports efficient bulk updates for supported models.

Transceiver firmware upgrades are the software upgrades that

- enhance transceiver performance and resolve known issues
- are bundled with NX-OS releases,
- and apply only to supported transceiver models.

The transceiver firmware upgrade feature provides network administrators with the ability to

- update transceiver firmware to address bugs and security vulnerabilities
- enhance transceiver performance through firmware optimizations
- maintain compliance with latest industry standards, and
- perform bulk firmware updates across multiple interfaces.

One of the ways in which the transceiver firmware upgrade can be performed is by using the NX-OS command framework. This requires careful procedural adherence to avoid disruption and maintain network stability. For more information, see Firmware upgrade using NX-OS commands on page 70.

You can upgrade transceiver firmware faster by processing multiple downloads simultaneously. The process involves careful grouping of ports for parallel operations, but ports on the same Inter-Integrated Circuit (I2C) line are updated sequentially. For more information, see Parallel firmware download on page 94.

Firmware upgrade using NX-OS commands

This section provides details on downloading, installing, and managing firmware upgrades for CDB-supported transceivers across platforms, highlighting CMIS compliance, unified commands, and robust error handling.

This section includes the required details that allow the users to download, install, and manage firmware upgrades for all Command Data Block (CDB) supported transceivers or optical transceivers across multiple hardware platforms.

The key benefits include

- CMIS compliance as this upgrade supports standards such as CMIS 4.0, 5.0, and above for modern transceivers

- unified or consistent command syntax across all supported platforms, and
- robust error handling through comprehensive error detection and recovery mechanisms.

The sections that follow explain the concept, procedures, and sample scenarios for the transceiver firmware upgrade using NX-OS commands.

Parallel firmware download queues and port mapping

Learn how parallel queues manage firmware downloads across port ranges, including queue-to-port mapping and download behavior.

The system supports parallel firmware downloads with eight separate queues. Each queue handles downloads for a specific range of ports. You can download up to eight files at once, with one assigned to each queue.

Here is a list of port to queue mappings:

- Queue 1: Ports 1-8
- Queue 2: Ports 9-16
- Queue 3: Ports 17-24
- Queue 4: Ports 25-32
- Queue 5: Ports 33-40
- Queue 6: Ports 41-48
- Queue 7: Ports 49-56
- Queue 8: Ports 57-64

> **Note**
>
> The parallel download behavior is determined by the selected ports and queues. If the specified ports are in different queues, then the file download happens simultaneously. If the specified ports are in the same queue, then other ports have to wait in the queue until the first port completes the file download.

Assuming the download is executed for ports 19 to 25, download starts for port 19 from queue 3. Port 20 to 24 will be in queue 3 in a waiting state. Download for port 25 from queue 4 will also start simultaneously.

Firmware upgrade phases

This section provides a summary of each phase involved in the transceiver firmware upgrade process.

The firmware upgrade process for transceivers involves distinct phases to manage the transfer, activation, and permanence or persistence of new firmware.

Firmware upgrades for transceivers occur in four main phases:

- Download: Transfers the new firmware image from the source (such as the switch bootflash) to the transceiver's memory. The download can be disruptive or non-disruptive.
- Activate: Switches the transceiver from the current firmware to the newly downloaded version immediately, with no warning. This phase activates the new firmware and commits it by immediately making it operational and persistent or permanent without a separate commit step.

- Run: Activates the new firmware temporarily for testing without making it permanent or persistent. If not committed, the firmware reverts after a reload. This phase allows you to test the new firmware without making the firmware upgrade permanent.
- Commit: Makes the newly activated firmware persistent across reloads. This step is required only after the run phase, and not after the activate phase as activate executes commit automatically. If a commit is not performed, the transceiver reverts to the old firmware after a subsequent reboot or power cycle.

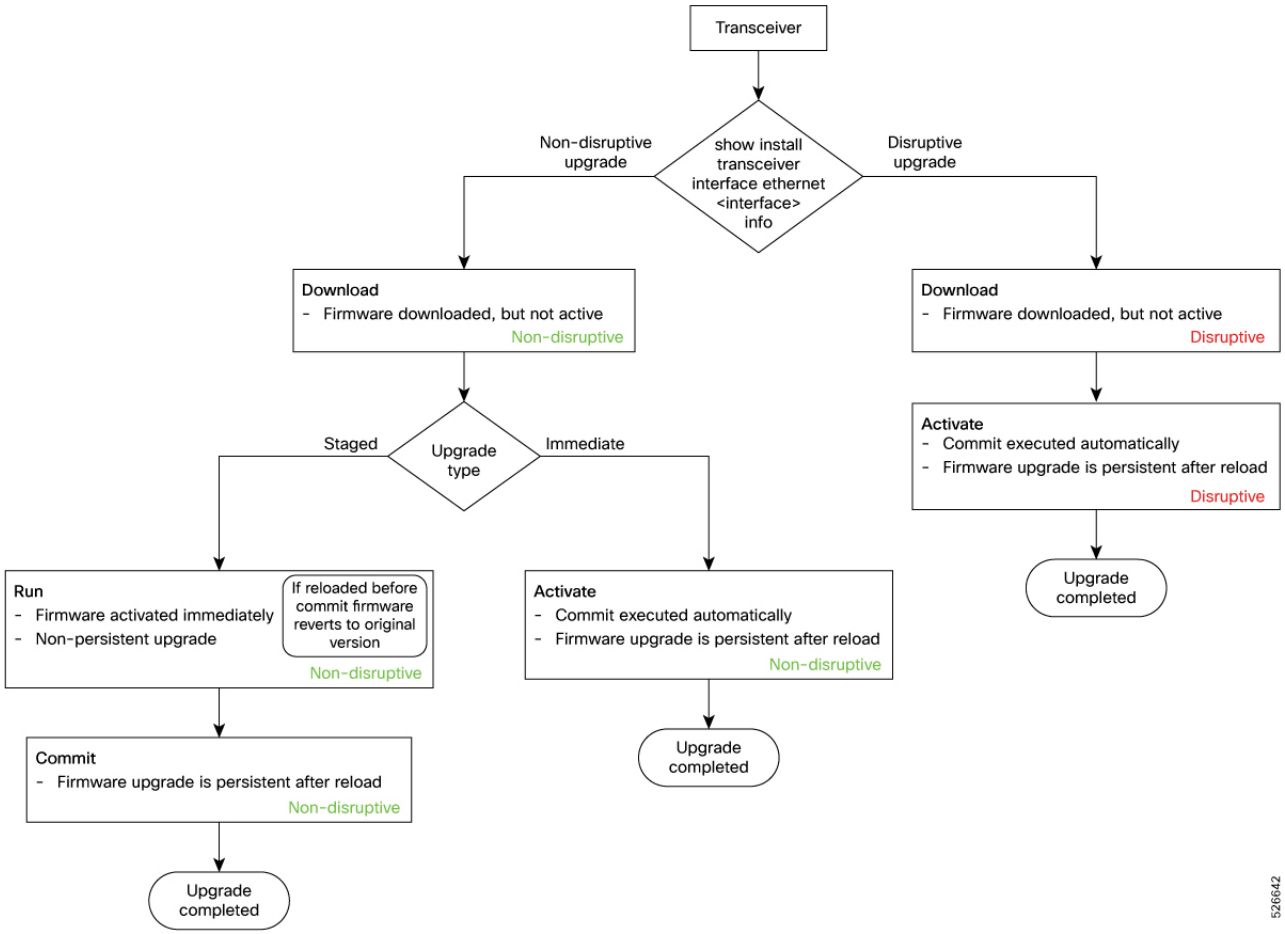
Firmware upgrade procedures

Based on whether the transceiver supports non-disruptive upgrade or disruptive upgrade, the upgrade procedures can be immediate or staged.

The procedure for firmware upgrade varies significantly depending on whether the transceiver supports non-disruptive upgrade or disruptive upgrade.

This is a flowchart that illustrates how the different phases are used for the transceiver upgrade. For more information about the procedure, see Upgrade firmware using NX-OS commands.

Figure 1: Firmware upgrade procedure



This table provides information about the upgrade phases recommended for transceivers based on the supported upgrade types.

Upgrade	Upgrade Type	Upgrade Phases
Disruptive	Immediate	1. Download 2. Activate
Non-disruptive	Staged	1. Download 2. Run 3. Commit
	Immediate	1. Download 2. Activate

Note

To determine whether an upgrade is disruptive, use the **show install transceiver interface ethernet** *<interface>* **info** command. If the answer to `Is Firmware Upgrade Service Affecting?` is

- `Yes`, then it indicates that the transceiver supports disruptive upgrade.
- `No`, then it indicates that the transceiver supports non-disruptive upgrade.

For sample scenarios, see Examples for firmware upgrade.

Guidelines and limitations for firmware upgrade using NX-OS commands

Guidelines and limitations for firmware upgrade using nx-os commands.

Read these operational limitations and precautions to avoid interruptions or failures when performing firmware upgrade using NX-OS commands.

- Beginning with Release 10.6(2)F, this feature is supported only on N9364E-SG2-Q and N9364E-SG2-O platforms and 800 G optics.
- Complete the firmware download before reloading or power cycling the system to avoid corruption. If the system restarts and the optic is not detected, restart the firmware download process to restore functionality.
- If the link fails to come up on a QDD-400G-ZRP-S Rev B transceiver running firmware version 61.24.16 configured with oFEC and dac-rate 1x1.25, upgrade the transceiver firmware on a switch running NX-OS 10.6(2)F or later, or use dac-rate 1x1 as a workaround; contact TAC for assistance, if needed.

Upgrade firmware using NX-OS commands

Outlines procedures for upgrading firmware on CDB-supported transceivers using NX-OS commands, covering disruptive and non-disruptive methods, required pre-requisites, command usage, and key considerations for different transceiver types.

Use the procedures provided in this section to upgrade the firmware of all Command Data Block (CDB) supported transceivers. For unsupported transceivers, the upgrade command displays an error.

For more information about ZR optics firmware upgrade, see Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide.

Note

- Different firmware files may be required for different transceiver types.
- Enter the **show install transceiver interface ethernet** *<interface>* **status** command after each state to determine current transceiver upgrade status and impact of next step.

Based on the upgrade types, there are three ways to upgrade the transceiver:

- Disruptive immediate upgrade of the transceiver
- Non-disruptive staged upgrade of the transceiver
- Non-disruptive immediate upgrade of the transceiver

Before you begin

The common pre-requisites for the three procedures are:

- Verify that the firmware file is correct and compatible with the transceiver model.
- Ensure that you have access to the switch command line interface and the ability to copy the firmware file from a remote location where it is stored (SCP, SFTP, and so on).

A detailed information about the three procedures are provided here.

Disruptive immediate upgrade of the transceiver

Procedure for transceivers that support only disruptive upgrade.

Use this procedure for transceivers that support only disruptive upgrade.

For more information about the different phases and upgrade procedures, see the Firmware upgrade procedures section before you perform this procedure.

For two sample scenarios, see Sample disruptive immediate upgrade with both banks valid and Sample disruptive immediate upgrade with one bank invalid on page 85.

Procedure

1. Get the firmware file, copy it on to the switch bootflash, and check the information of the firmware.
   a) Copy the firmware file to the switch under bootflash.

   Example

   ```
   switch# copy scp://<ip-address>/<location of
   firmware>/bank_trapp_qsfp-100g-sr-s_signed_rel_0.73.4.bin
   bootflash:///bank_trapp_qsfp-100g-sr-s_signed_rel_0.73.4.bin vrf management
    use
   ```

   b) Verify the information of the firmware including versions and default running images that reside in the module using the **show install transceiver interface ethernet** *<interface>* **info** command.

Example

```
switch# show install transceiver interface ethernet 1/7/1 info
Eth1/7/1
********************Bank A**********************
================================================
Version : 2.6.0 : Build Time: May 28 2025 15:56:10
Operational Status     : Not Running
Administrative Status  : Uncommitted
Validity Status        : Valid
------------------------------------------------
********************Bank B**********************
================================================
Version : 2.6.0 : Build Time: May 28 2025 15:56:10
Operational Status     : Running
Administrative Status  : Committed
Validity Status        : Valid
------------------------------------------------
Is Firmware Upgrade Service Affecting? Yes
------------------------------------------------
```

Note

In the show install output, `Yes` indicates that the transceiver supports disruptive upgrade and a `No` indicates that the transceiver supports non-disruptive upgrade as well.

Verify the output for other significant information such as:

- Image version – indicates the version of the image
- Operational Status – indicates the operational status of the image, whether it is running or not running
- Administrative Status - indicates whether the image is committed or not committed
- Image validity - indicates the validity of the image, whether it is valid or invalid

c) Verify the status using the **show install transceiver interface ethernet** *<interface>* **status** command.

Example

```
switch# show install transceiver interface ethernet 1/7/1 status
Eth1/7/1
Download Status:    Ready to start
Run Status:         Ready to start
Commit Status:      Ready to start
Activation Status:  Ready to start
```

When the status is `Ready to start`, proceed to the next step.

2. Download firmware on the transceiver module and verify the download status.

a) Download and install the firmware on the transceiver with the **install transceiver interface ethernet** *<interface>***download bootflash:** *<firmware-file-name>* command. The download is disruptive.

> **Note**
>
> Parallel firmware download is supported for Silicon One PIDs with range command support for non-ZR transceiver. If the same firmware file is used for all transceivers, use the same file across all devices. If different firmware files are required for different transceivers, copy those files under bootflash and specify the corresponding file names in the download command.

Example

```
switch# install transceiver interface ethernet 1/11 download
bootflash:tmg_bank_trappb_qdd-400g-bd_signed_dev.bin
```

b) Verify the status of the firmware download using the **show install transceiver interface ethernet** *<interface>* **status** command. Track the progress of the download using this command.

Example

```
switch# show install transceiver interface E1/16 status
Eth1/16:
Last download is completed successfully
Download Status:   Ready to start
Run Status:        Ready to start
Commit Status:     Ready to start
Activation Status: Ready to start
```

> **Note**
>
> When the download is in progress, you can execute the next command, which will be queued.

3. Activate the firmware using the **install transceiver interface ethernet** *<interface>* **activate** command. The new upgrade persists across reloads.

   Example

   ```
   switch# install transceiver interface E1/16 activate
   ```

   After performing this command, verify the progress using the **show install transceiver interface ethernet** *<interface>* **status** command.

4. Verify that the module is upgraded and the newly downloaded image is running and committed using the **show install transceiver interface ethernet** *<interface>* **info** command.

Non-disruptive staged upgrade of the transceiver

This is a procedure for transceivers that support non-disruptive upgrade and when you want the upgrade to be a staged upgrade.

Use this procedure for transceivers that support non-disruptive upgrade and when you want the upgrade to be a staged upgrade.

For more information about the different phases and upgrade procedures, see the Firmware upgrade procedures section before you perform this procedure.

For a sample scenario, see Sample non-disruptive staged upgrade on page 87.

Procedure

1. Get the firmware file, copy it on to the switch bootflash, and check the information of the firmware.
   a) Copy the firmware file to the switch under bootflash.

      Example

      ```
      switch# copy scp://<ip-address>/<location of
      firmware>/bank_trapp_qsfp-100g-sr-s_signed_rel_0.73.4.bin
      bootflash:///bank_trapp_qsfp-100g-sr-s_signed_rel_0.73.4.bin vrf management
       use
      ```

   b) Verify the information of the firmware including versions and default running images that reside in the module using the **show install transceiver interface ethernet** *<interface>* **info** command.

      Example

      ```
      switch# show install transceiver interface ethernet 1/7/1 info
      Eth1/7/1
      ********************Bank A**********************
      ================================================
      Version : 2.6.0 : Build Time: May 28 2025 15:56:10
      Operational Status      : Not Running
      Administrative Status   : Uncommitted
      Validity Status         : Valid
      ------------------------------------------------
      ********************Bank B**********************
      ================================================
      Version : 2.6.0 : Build Time: May 28 2025 15:56:10
      Operational Status      : Running
      Administrative Status   : Committed
      Validity Status         : Valid
      ------------------------------------------------
      Is Firmware Upgrade Service Affecting? Yes
      ------------------------------------------------
      ```

      Note

      In the show install output, Yes indicates that the transceiver supports disruptive upgrade and a No indicates that the transceiver supports non-disruptive upgrade as well.

      Verify the output for information such as

      - Image version – indicates the version of the image
      - Operational Status – indicates the operational status of the image, whether it is running or not running

- Administrative Status - indicates whether the image is committed or not committed
- Image validity - indicates the validity of the image, whether it is valid or invalid
- Is Firmware Upgrade Service Affecting? - Yes indicates that the transceiver supports disruptive upgrade and No indicates that the transceiver supports non-disruptive upgrade

c) Verify the status using the **show install transceiver interface ethernet** *<interface>* **status** command.

Example

```
switch# show install transceiver interface ethernet 1/7/1 status
Eth1/7/1
Download Status:    Ready to start
Run Status:         Ready to start
Commit Status:      Ready to start
Activation Status:  Ready to start
```

When the status is `Ready to start`, proceed to the next step.

2. Download firmware on the transceiver module and verify the download status.

a) Download and install the firmware on the transceiver with the **install transceiver interface ethernet** *<interface>***download bootflash:** *<firmware-file-name>* command. The download is disruptive.

> **Note**
>
> Parallel firmware download is supported for Silicon One PIDs with range command support for non-ZR transceiver. If the same firmware file is used for all transceivers, use the same file across all devices. If different firmware files are required for different transceivers, copy those files under bootflash and specify the corresponding file names in the download command.

Example

```
switch# dnf repolist
```

b) Verify the status of the firmware download using the **show install transceiver interface ethernet** *<interface>* **status** command. Track the progress of the download using this command.

Example

```
switch# show install transceiver interface E1/16 status
Eth1/16:
Last download is completed successfully
Download Status:   Ready to start
Run Status:        Ready to start
Commit Status:     Ready to start
Activation Status: Ready to start
```

> **Note**
>
> When the download is in progress, you can execute the next command, which will be queued.

3. Activate the firmware using the run and commit commands. Both commands are non-disruptive.
   a) Run the new firmware version using the **install transceiver interface ethernet** *<interface>* **run** command.

   > **Note**
   >
   > Do not reload before running the commit command. The upgrade is non-persistent. So, if you reload before using the commit command, the transceiver reverts to the version before the upgrade.

   Example

   ```
   switch# install transceiver interface E1/16 run
   ```

   b) Commit the downloaded firmware. Use the **install transceiver interface ethernet** *<interface>* **commit** command, so the module records the new firmware image, which is now active and is ready to be used on any future reset or reboot.

   Example

   ```
   switch# install transceiver interface E1/16 commit
   ```

   After performing the commit command, verify the progress using the **show install transceiver interface ethernet** *<interface>* **status** command.

4. Verify that the module is upgraded and the newly downloaded image is running and committed using the **show install transceiver interface ethernet** *<interface>* **info** command.

Non-disruptive immediate upgrade of the transceiver

This is a procedure for transceivers that support non-disruptive upgrade when you choose to perform an immediate upgrade.

Use this procedure for transceivers that support non-disruptive upgrade and when you want the upgrade to be an immediate upgrade.

For more information about the different phases and upgrade procedures, see the Firmware upgrade procedures section before you perform this procedure.

For a sample scenario, see Sample non-disruptive immediate upgrade on page 91.

Procedure

---

1. Get the firmware file, copy it on to the switch bootflash, and check the information of the firmware.
   a) Copy the firmware file to the switch under bootflash.

Example

```
switch# copy scp://<ip-address>/<location of
firmware>/bank_trapp_qsfp-100g-sr-s_signed_rel_0.73.4.bin
bootflash:///bank_trapp_qsfp-100g-sr-s_signed_rel_0.73.4.bin vrf management
 use
```

b) Verify the information of the firmware including versions and default running images that reside in the module using the **show install transceiver interface ethernet** *<interface>* **info** command.

Example

```
switch# show install transceiver interface ethernet 1/7/1 info
Eth1/7/1
********************Bank A**********************
================================================
Version : 2.6.0 : Build Time: May 28 2025 15:56:10
Operational Status     : Not Running
Administrative Status  : Uncommitted
Validity Status        : Valid
------------------------------------------------
********************Bank B**********************
================================================
Version : 2.6.0 : Build Time: May 28 2025 15:56:10
Operational Status     : Running
Administrative Status  : Committed
Validity Status        : Valid
------------------------------------------------
Is Firmware Upgrade Service Affecting? Yes
------------------------------------------------
```

Note

In the show install output, Yes indicates that the transceiver supports disruptive upgrade and a No indicates that the transceiver supports non-disruptive upgrade as well.

Verify the output for information such as

- Image version – indicates the version of the image
- Operational Status – indicates the operational status of the image, whether it is running or not running
- Administrative Status - indicates whether the image is committed or not committed
- Image validity - indicates the validity of the image, whether it is valid or invalid
- Is Firmware Upgrade Service Affecting? – Yes indicates that the transceiver supports disruptive upgrade and No indicates that the transceiver supports non-disruptive upgrade

c) Verify the status using the **show install transceiver interface ethernet** *<interface>* **status** command.

Example

```
switch# show install transceiver interface ethernet 1/7/1 status
Eth1/7/1
Download Status:    Ready to start
```

```
Run Status:          Ready to start
Commit Status:       Ready to start
Activation Status:   Ready to start
```

When the status is `Ready to start`, proceed to the next step.

2. Download firmware on the transceiver module and verify the download status.
   a) Download and install the firmware on the transceiver with the **install transceiver interface ethernet** *<interface>* **download bootflash:** *<firmware-file-name>* command. The download is non-disruptive.

> **Note**
>
> Parallel firmware download is supported for Silicon One PIDs with range command support for non-ZR transceiver. If the same firmware file is used for all transceivers, use the same file across all devices. If different firmware files are required for different transceivers, copy those files under bootflash and specify the corresponding file names in the download command.

Example

```
switch# install transceiver interface ethernet 1/11 download
bootflash:tmg_bank_trappb_qdd-400g-bd_signed_dev.bin
```

   b) Verify the status of the firmware download using the **show install transceiver interface ethernet** *<interface>* **status** command. Track the progress of the download using this command.

Example

```
switch# show install transceiver interface E1/16 status
Eth1/16:
Last download is completed successfully
Download Status:   Ready to start
Run Status:        Ready to start
Commit Status:     Ready to start
Activation Status: Ready to start
```

> **Note**
>
> When the download is in progress, you can execute the next command, which will be queued.

3. Activate the firmware using the run and commit commands. Both commands are non-disruptive.
   a) Run the new firmware version using the **install transceiver interface ethernet** *<interface>* **run** command.

> **Note**
>
> Do not reload before running the commit command. The upgrade is non-persistent. So, if you reload before using the commit command, the transceiver reverts to the version before the upgrade.

Example

```
switch# install transceiver interface E1/16 run
```

b) Commit the downloaded firmware. Use the **install transceiver interface ethernet** *<interface>* **commit** command, so the module records the new firmware image, which is now active and is ready to be used on any future reset or reboot.

Example

```
switch# install transceiver interface E1/16 commit
```

After performing the commit command, verify the progress using the **show install transceiver interface ethernet** *<interface>* **status** command.

4. Activate the firmware using the **install transceiver interface ethernet** *<interface>* **activate** command. The new upgrade persists across reloads.

Example

```
switch# install transceiver interface E1/16 activate
```

After performing this command, verify the progress using the **show install transceiver interface ethernet** *<interface>* **status** command.

5. Verify that the module is upgraded and the newly downloaded image is running and committed using the **show install transceiver interface ethernet** *<interface>* **info** command.

Examples for firmware upgrade

Scenarios of firmware upgrade

This section provides few examples for certain scenarios that you may encounter during the firmware upgrade.

Sample disruptive immediate upgrade with both banks valid

Sample scenario for the disruptive immediate upgrade of a transceiver firmware that supports only disruptive upgrade, and the transceiver displays both Bank A and B as valid.

Transceivers have either one or two banks, labeled as Bank A and B, to store the firmware image. Either of the banks can hold valid firmware images. During an upgrade, the new firmware is first downloaded to the inactive bank–the one not in use. After the download, the firmware is activated by switching the active bank to the newly updated bank, while the previously active bank becomes inactive. This method allows for a safer upgrade process by retaining the previous firmware until the new firmware is fully committed and finalized.

This section shows a sample scenario for the disruptive immediate upgrade of a transceiver firmware that supports only disruptive upgrade and has both Bank A and B status as valid.

For more information about the procedure, see Disruptive immediate upgrade of the transceiver on page 74.

Let us assume that v 3.2.0 is the current active version of your transceiver firmware and you want to upgrade to v 3.9.0.

Procedure

1. Enter the **info** command.

   Example

```
switch# show install transceiver interface ethernet 1/1 info
Eth1/1
********************Bank A**********************
================================================
Version                 : 3.1.0
Operational Status      : Not Running
Administrative Status   : Uncommitted
Validity Status         : Valid
------------------------------------------------
********************Bank B**********************
================================================
Version                  : 3.2.0
Operational Status       : Running
Administrative Status    : Committed
Validity Status          : Valid
------------------------------------------------
Is Firmware Upgrade Service Affecting? Yes
------------------------------------------------
```

   The current active image is v 3.2.0 and this resides in Bank B.

2. Next, enter the **download** command.

   Example

```
switch# install transceiver interface ethernet 1/1 download
bootflash:CISCO-EOPTOLINK_QDD-8X100G-FR_HW_1.0_FW_3.9.bin
Transceiver firmware download Command submitted, Please use status command
to check progress.
```

3. Enter the **status** command to verify the progress.

   Example

```
switch# show install transceiver interface ethernet 1/1 status
Downloading is in progress [5/100 Completed].
Eth1/1
Download Status:    In Progress
Run Status:         Ready to start
Commit Status:      Ready to start
Activation Status:  Ready to start
```

The download is in progress. After completion of the download, the status command output displays that the download is completed.

```
switch# show install transceiver interface ethernet 1/1 status
Last Download is completed successfully
Eth1/1
Download Status:    Ready to start
Run Status:         Ready to start
Commit Status:      Ready to start
Activation Status:  Ready to start
```

4. Enter the **info** command.

Example

```
switch# show install transceiver interface ethernet 1/1 info
Eth1/1
********************Bank A**********************
================================================
Version                 : 3.9.0
Operational Status      : Not Running
Administrative Status   : Uncommitted
Validity Status         : Valid
------------------------------------------------
********************Bank B**********************
================================================
Version                 : 3.2.0
Operational Status      : Running
Administrative Status   : Committed
Validity Status         : Valid
------------------------------------------------
Is Firmware Upgrade Service Affecting? Yes
------------------------------------------------
```

The new image, v 3.9.0, is downloaded to Bank A, the passive bank (which has the new but inactive image) .

5. Enter the **activate** command to activate the new image, v 3.9.0.

Example

```
switch# install transceiver interface ethernet 1/1 activate
Transceiver firmware Activate Command submitted, Please use status command
to check progress
```

6. Enter the **info** command.

Example

```
switch# show install transceiver interface ethernet 1/1 info
Eth1/1
********************Bank A**********************
================================================
Version                  : 3.9.0
Operational Status       : Running
Administrative Status    : Committed
Validity Status          : Valid
------------------------------------------------
********************Bank B**********************
================================================
```

```
Version                    : 3.2.0
Operational Status         : Not Running
Administrative Status      : Uncommitted
Validity Status            : Valid
------------------------------------------------
Is Firmware Upgrade Service Affecting? Yes
------------------------------------------------
```

Bank A is active now with the new image, v 3.9.0.

The transceiver firmware upgrade is complete.

Sample disruptive immediate upgrade with one bank invalid

Sample scenario for the disruptive immediate upgrade of a transceiver firmware that supports only disruptive upgrade, and the transceiver displays one bank as valid, either Bank A or Bank B.

Transceivers have either one or two banks to store the firmware image. The banks are labelled as Bank A and Bank B. This section shows a sample scenario for the disruptive immediate upgrade of a transceiver firmware that supports only disruptive upgrade and has only one Bank with a valid status, either Bank A or Bank B.

For more information about the procedure, see Disruptive immediate upgrade of the transceiver on page 74.

Let us assume that v 126.4.0 is the current active version of your transceiver firmware and you want to upgrade to v 126.55.0.

Procedure

---

1. Enter the **info** command.

   Example

```
switch# show install transceiver interface ethernet 1/23 info
Eth1/23
********************Bank A**********************
================================================
Version                          : 126.4.0 :
Operational Status               : Running
Administrative Status            : Committed
Validity Status                  : Valid
------------------------------------------------
********************Bank B**********************
================================================
Version                           : 0.0.0 :
Operational Status                : Not Running
Administrative Status             : Uncommitted
Validity Status                   : Invalid
------------------------------------------------
Is Firmware Upgrade Service Affecting? Yes
------------------------------------------------
```

   The current active image is v 126.4.0 and this resides in Bank A.

> **Note**
>
> As the answer to the question `Is Firmware Upgrade Service Affecting?` is `Yes`, the transceiver supports disruptive upgrade.

2. Next, enter the **download** command.

   Example

```
switch# install transceiver interface ethernet 1/23 download
bootflash:CISCO-INNOLIGHT_OSFP-800G-VR8_HW_1.10_FW_126.55.bin
Transceiver firmware download Command submitted, Please use status command
to check progress
```

3. Enter the **status** command to verify the progress.

   Example

```
switch# show install transceiver interface ethernet 1/23 status
Downloading is in progress [10/100 Completed].
Eth1/23
Download Status:      In Progress
Run Status:           Ready to start
Commit Status:        Ready to start
Activation Status:    Ready to start
```

   The download is in progress. After completion of the download, the status command output displays that the download is completed.

```
switch# show install transceiver interface ethernet 1/23 status
Last Download is completed successfully
Eth1/23
Download Status:   Ready to start
Run Status:        Ready to start
Commit Status:     Ready to start
Activation Status: Ready to start
```

4. Enter the **info** command.

   Example

```
switch# show install transceiver interface ethernet 1/23 info
Eth1/23
********************Bank A**********************
================================================
Version                 : 126.55.0
Operational Status      : Running
Administrative Status   : Committed
Validity Status         : Valid
------------------------------------------------
********************Bank B**********************
================================================
Version                 : 0.0.0
Operational Status      : Not Running
```

```
Administrative Status   : Uncommitted
Validity Status         : Invalid
------------------------------------------------
Is Firmware Upgrade Service Affecting? Yes
------------------------------------------------
```

The new image, v 126.55.0, has replaced the existing image as it has only one bank to store the firmware image. It is activated by default. However, still it is recommended to use the activate command.

> **Note**
>
> Bank B can be invalid in cases where the transceiver does not use Bank B.

5. Enter the activate command to activate the new image, v 126.55.0.

Example

```
switch# install transceiver interface ethernet 1/23 activate
Transceiver firmware Activate Command submitted, Please use status command
to check progress
```

6. Enter the **info** command.

Example

```
switch# show install transceiver interface ethernet 1/23 info
Eth1/23
********************Bank A**********************
================================================
Version                 : 126.55.0
Operational Status      : Running
Administrative Status   : Committed
Validity Status         : Valid
------------------------------------------------
********************Bank B**********************
================================================
Version                 : 0.0.0 :
Operational Status      : Not Running
Administrative Status   : Uncommitted
Validity Status         : Invalid
------------------------------------------------
Is Firmware Upgrade Service Affecting? Yes
------------------------------------------------
```

Bank A is active now with the new image, v 126.55.0. The transceiver firmware upgrade is complete.

Sample non-disruptive staged upgrade

Sample scenario for the non-disruptive staged upgrade of a transceiver firmware.

This section shows a sample scenario for the non-disruptive staged upgrade of a transceiver firmware.

For more information about the procedure, see Non-disruptive staged upgrade of the transceiver on page 76.

Let us assume that v 3.12.0 is the current active version of your transceiver firmware and you want to upgrade to v 3.15.0.

Procedure

1. Enter the **info** command.

   Example

```
switch# show install transceiver interface ethernet 1/1 info
Eth1/1
********************Bank A**********************
================================================
Version                 : 3.9.0
Operational Status      : Not Running
Administrative Status   : Uncommitted
Validity Status         : Valid
------------------------------------------------
********************Bank B**********************
================================================
Version                   : 3.12.0
Operational Status        : Running
Administrative Status     : Committed
Validity Status           : Valid
------------------------------------------------
Is Firmware Upgrade Service Affecting? No
------------------------------------------------
```

   The current active image is v 3.12.0 and this resides in Bank B.

   > **Note**
   >
   > As the answer to the question `Is Firmware Upgrade Service Affecting?` is `No`, the transceiver supports non-disruptive upgrade.

2. Next, enter the **download** command.

   Example

```
switch# install transceiver interface ethernet 1/1 download
bootflash:CISCO-EOPTOLINK_QDD-8X100G-FR_HW_1.0_FW_3.15.bin
Transceiver firmware download Command submitted, Please use status command
to check progress.
```

3. Enter the **status** command to verify the progress.

   Example

```
switch# show install transceiver interface ethernet 1/1 status
Downloading is in progress [5/100 Completed].
Eth1/1
Download Status:    In Progress
Run Status:         Ready to start
```

```
Commit Status:      Ready to start
Activation Status:  Ready to start
```

The download is in progress. After completion of the download, the status command output displays that the download is completed.

```
switch# show install transceiver interface ethernet 1/1 status
Last Download is completed successfully
Eth1/1
Download Status:    Ready to start
Run Status:         Ready to start
Commit Status:      Ready to start
Activation Status:  Ready to start
```

4. Enter the **info** command.

Example

```
switch# show install transceiver interface ethernet 1/1 info
Eth1/1
********************Bank A**********************
================================================
Version                  : 3.15.0
Operational Status       : Not Running
Administrative Status    : Uncommitted
Validity Status          : Valid
------------------------------------------------
********************Bank B**********************
================================================
Version                  : 3.12.0
Operational Status       : Running
Administrative Status    : Committed
Validity Status          : Valid
------------------------------------------------
Is Firmware Upgrade Service Affecting? No
------------------------------------------------
```

The new image, v 3.15.0, is downloaded to Bank A, the passive bank. This image is inactive.

5. Enter the run command to activate the new image, v 3.15.0.

Example

```
switch# install transceiver interface ethernet 1/1 run
Transceiver firmware Run Command submitted, Please use status command to check
 progress
```

6. Enter the **status** command to check the progress.

Example

```
switch# show install transceiver interface ethernet 1/1 status
Last Run is completed successfully
Eth1/1
Download Status:        Ready to start
Run Status:             Ready to start
Commit Status:          Ready to start
Activation Status:      Ready to start
```

The output displays that the run command is completed.

Now, enter the **info** command.

```
switch# show install transceiver interface ethernet 1/1 info
Eth1/1
********************Bank A**********************
================================================
Version                          : 3.15.0
Operational Status      : Running
Administrative Status   : Uncommitted
Validity Status         : Valid
------------------------------------------------
********************Bank B**********************
================================================
Version                 : 3.12.0
Operational Status      : Not Running
Administrative Status   : Committed
Validity Status         : Valid
------------------------------------------------
Is Firmware Upgrade Service Affecting? No
------------------------------------------------
```

Image v 3.15.0 in Bank A is running but not committed. Though the v 3.15.0 is active, it is temporarily active until the next reload or reset. It will be permanently or persistently active only after it is committed (next step). If not committed, then the existing image v 3.12.0 in bank B will become active after the next reset or reload operation.

7. Enter the **commit** command to make the activation persistent.

   Example

```
switch# install transceiver interface ethernet 1/1 commit
Transceiver firmware Commit Command submitted, Please use status command to
check progress
```

8. Enter the **status** command.

   Example

```
switch# show install transceiver interface ethernet 1/1 status
Last Commit is completed successfully
Eth1/1
Download Status:      Ready to start
Run Status:           Ready to start
Commit Status:        Ready to start
Activation Status:    Ready to start
```

9. Enter the **info** command

   Example

```
switch# show install transceiver interface ethernet 1/1 info
Eth1/1
********************Bank A**********************
================================================
Version                 : 3.15.0
Operational Status      : Running
Administrative Status   : Committed
Validity Status         : Valid
```

```
------------------------------------------------
********************Bank B**********************
================================================
Version                 : 3.12.0
Operational Status      : Not Running
Administrative Status   : Uncommitted
Validity Status         : Valid
------------------------------------------------
Is Firmware Upgrade Service Affecting? No
------------------------------------------------
```

Bank A is active now with the new image, v 3.15.0, and the upgrade is persistent. Image v 3.12.0 in Bank B is inactive even after a reload.

The transceiver firmware upgrade is complete.

Sample non-disruptive immediate upgrade

Sample scenario for the non-disruptive immediate upgrade of a transceiver firmware.

This section shows a sample scenario for the non-disruptive immediate upgrade of a transceiver firmware.

For more information about the procedure, see Non-disruptive immediate upgrade of the transceiver on page 79.

Let us assume that v 3.9.0 is the current active version of your transceiver firmware and you want to upgrade to v 3.12.0.

Procedure

---

1. Enter the **info** command.

   Example

```
switch# show install transceiver interface ethernet 1/1 info
Eth1/1
********************Bank A**********************
================================================
Version                 : 3.9.0
Operational Status      : Running
Administrative Status   : Committed
Validity Status         : Valid
------------------------------------------------
********************Bank B**********************
================================================
Version                 : 3.2.0
Operational Status      : Not Running
Administrative Status   : Uncommitted
Validity Status         : Valid
------------------------------------------------
Is Firmware Upgrade Service Affecting? No
------------------------------------------------
```

   The current active image is v 3.9.0 and this resides in Bank A.

> **Note**
>
> As the answer to the question `Is Firmware Upgrade Service Affecting?` is `No`, the transceiver supports non-disruptive upgrade.

2. Next, enter the **download** command.

   Example

```
switch# install transceiver interface ethernet 1/1 download
bootflash:CISCO-EOPTOLINK_QDD-8X100G-FR_HW_1.0_FW_3.12.bin
Transceiver firmware download Command submitted, Please use status command
to check progress.
```

3. Enter the **status** command to verify the progress.

   Example

```
switch# show install transceiver interface ethernet 1/1 status
Downloading is in progress [5/100 Completed].
Eth1/1
Download Status:    In Progress
Run Status:         Ready to start
Commit Status:      Ready to start
Activation Status:  Ready to start
```

   The download is in progress. After completion of the download, the status command output displays that the download is completed.

```
switch# show install transceiver interface ethernet 1/1 status
Last Download is completed successfully
Eth1/1
Download Status:    Ready to start
Run Status:         Ready to start
Commit Status:      Ready to start
Activation Status:  Ready to start
```

4. Enter the **info** command.

   Example

```
switch# show install transceiver interface ethernet 1/1 info
Eth1/1
********************Bank A**********************
================================================
Version                 : 3.9.0
Operational Status      : Running
Administrative Status   : Committed
Validity Status         : Valid
------------------------------------------------
********************Bank B**********************
================================================
Version                 : 3.12.0
Operational Status      : Not Running
```

```
Administrative Status   : Uncommitted
Validity Status         : Valid
------------------------------------------------
Is Firmware Upgrade Service Affecting? No
------------------------------------------------
```

The new image, v 3.12.0, is downloaded to Bank B, the passive bank. This image is inactive.

5. Enter the **activate** command to activate the new image, v 3.12.0.

Example

```
switch# install transceiver interface ethernet 1/1 activate
Transceiver firmware Activate Command submitted, Please use status command
to check progress
```

6. Enter the **info** command

Example

```
switch# show install transceiver interface ethernet 1/1 info
Eth1/1
********************Bank A**********************
================================================
Version                  : 3.9.0
Operational Status       : Not Running
Administrative Status    : Uncommitted
Validity Status          : Valid
------------------------------------------------
********************Bank B**********************
================================================
Version                    : 3.12.0
Operational Status         : Running
Administrative Status      : Committed
Validity Status            : Valid
------------------------------------------------
Is Firmware Upgrade Service Affecting? No
------------------------------------------------
```

Bank B is active now with the new image, v 3.12.0. Image v 3.9.0 in Bank A is inactive.

The transceiver firmware upgrade is complete.

Firmware upgrade for a range of interfaces or parallel firmware download

Firmware Parallel download and range of interfaces.

This is a sample output for the parallel firmware download and information command for a range of interfaces.

```
switch# show install transceiver interface E1/59,E1/60-64 info
install transceiver interface E1/15-18 download
bootflash:CISCO-EOPTOLINK_QDD-2X400G-LR4_HW_1.2_FW_3.13.bin
show install transceiver interface  E1/15-18 status
install transceiver interface E1/15-18 run
install transceiver interface E1/15-18 commit
install transceiver interface E1/15-18 activate
```

Parallel firmware download

This section provides an overview of the parallel firmware download feature, which accelerates transceiver firmware upgrades by enabling simultaneous downloads, and outlines the necessary steps, guidelines, and precautions for efficiently copying, downloading, running, and committing firmware on switch transceiver ports.

The parallel firmware download feature reduces the time needed for upgrading transceiver firmware in network switches by allowing multiple downloads to occur simultaneously.

Firmware download guidelines and limitations

Review these guidelines and limitations before performing firmware downloads and related operations on transceivers.

Understand the supported firmware download workflow and operational steps before starting the update.

- Group ports correctly for parallel processing.
- Although different port groups can be processed in parallel, ports on the same Inter-Integrated Circuit (I2C) line download sequentially.
- Use this list of commands to perform these operations:
  - Start: Initiates the firmware download to the transceiver.

    Note

    For single-bank transceivers, this overwrites the existing firmware. Execute the run and commit commands even for single-bank transceivers. For dual-bank transceivers, the new firmware is downloaded to the inactive bank.

  - Run: Activates the newly downloaded firmware image.

    Note

    For dual-bank transceivers, this command switches the newly downloaded firmware to the active bank and moves the previously active firmware to the inactive bank.

  - Commit: Finalizes the firmware update by ensuring that the active firmware image is used during future resets or reboots.

Follow these operational limitations and precautions to avoid interruptions or failures during firmware updates.

- Monitor system resources to ensure that firmware threads do not interfere with higher-priority operations such as USD threads.
- Run the **info** command per port because it does not support range input, unlike **start**, **commit**, and **run**.
- Perform all physical OIR (Online Insertion and Removal) actions outside the firmware download process. Restart the firmware download process if necessary.
- Complete the firmware download before reloading or power cycling the system to avoid corruption. If the system restarts and the optic is not detected, restart the firmware download process to restore functionality.

Copy Transceiver Firmware to Switch and Line Card

Copy the transceiver firmware file to the switch bootflash and then to the line card temporary directory before you initiate the firmware update process on transceiver ports.

This is a preparatory step for updating transceiver firmware. Different firmware files may be needed for distinct transceiver types. Name each file appropriately on the Line Card (LC). Then, refer to it as LC throughout the document.

Before you begin

Ensure you have access to the switch and a network location such as an SCP server where the firmware file is stored. You must have root access on the line card to copy files to the LC.

Procedure

1. Use the **copy** command to copy the firmware file to the switch's bootflash.

   Example

   ```
   switch# copy scp://192.0.2.1/<location of
   firmware>/bank_trapp_qsfp-100g-sr-s_signed_rel_0.73.4.bin
   bootflash:///bank_trapp_qsfp-100g-sr-s_signed_rel_0.73.4.bin vrf man use
   ```

   This command copies the firmware file from an SCP server to the switch bootflash memory.

2. Use the **run bash sudo su** and **scp** commands to gain root access and copy the firmware file from the switch to the Line Card (LC).

   Example

   ```
   run bash sudo su
               bash# scp -r
   /bootflash/bank_trapp_qsfp-100g-sr-s_signed_rel_0.73.4.bin
   root@lc1:/var/sysmgr/tmp/firmware.bin
   ```

   This command transfers the firmware file from the switch bootflash to the specified Line Card temporary directory. If different transceiver types need separate firmware files, copy and name each file so they are easily identified.

3. Use the **rlogin** and **ls -lrt** commands to log in to the Line Card and verify the firmware file.

   Example

   ```
   rlogin lc1
               root@lc1:~# ls -lrt /var/sysmgr/tmp/firmware.bin
               -rw-r--r-- 1 root root 831200 May 7 17:54
   /var/sysmgr/tmp/firmware.bin
   ```

   This step confirms that the firmware file is present and accessible on the Line Card device.

The transceiver firmware file is successfully copied to the switch bootflash and the Line Card temporary directory, and its presence is verified.

What to do next

Proceed to download, run, and commit the firmware on the transceiver ports as described in Download, Run, and Commit Transceiver Firmware.

Download, Run, and Commit Transceiver Firmware

Download, verify the status, run, and commit the pre-staged firmware on the specified transceiver ports.

Download the pre-staged firmware to the specified transceiver ports, verify the status, run the firmware, and then commit it.

Perform this task after you successfully copy the firmware file to the switch and line card. The task requires direct interaction with the transceiver hardware.

Before you begin

Before initiating the firmware download:

- Ensure the firmware file is copied to the switch and line card as described in "Copy Transceiver Firmware to Switch and Line Card".
- Verify that the firmware file is correct and compatible with the transceiver model.
- Verify that the file paths and names are accurate.

Procedure

1. Use the **debug hardware internal linkctrl transceiver firmware fp-port 1-10,6,15-16 start file-name** */var/sysmgr/tmp/firmware10.bin* command to initiate firmware download to a specified range of transceiver ports.

   Example

   ```
   switch# debug hardware internal linkctrl transceiver firmware fp-port
   1-10,6,15-16 start file-name /var/sysmgr/tmp/firmware10.bin
   ```

   Warning

   Executing firmware download commands requires caution. Using an incorrect firmware file can lead to a failed download and may corrupt the existing firmware. This is particularly critical for single bank transceivers, as they may require a firmware re-download to recover from a faulty state.

   Note

   During the firmware download process, the optical module will go offline automatically and will come back online once the download is completed. This behavior is part of the update process and does not require manual intervention.

2. Optional: Use the **debug hardware internal linkctrl transceiver firmware fp-port 14 start file-name** */var/sysmgr/tmp/firmware10.bin* command to initiate firmware download to a single specified transceiver port.

   Example

   ```
   switch# debug hardware internal linkctrl transceiver firmware fp-port 14 start
    file-name /var/sysmgr/tmp/firmware10.bin
   ```

Use this command to start downloading firmware to a single specified port.

3. Use the **debug hardware internal linkctrl transceiver firmware fp-port 3 status** command to check the status of the firmware download process.

   Example

```
switch# debug hardware internal linkctrl transceiver firmware fp-port 3 status
front_port_num[3]  Status of Firmware Download

QUEUE content for the group of 8 ports
Port: 3, File: /var/sysmgr/tmp/firmware10.bin Op Type: Run
Port: 4, File: /var/sysmgr/tmp/firmware10.bin Op Type: Run

-> Run command is in queue

-> Downloading is not in progress

-> Run command is not in progress

-> Commit is not in progress

-> Download completed percentage: 100


Error occured in last five commands
----------------------------------------
2025-02-19 11:08:57.199684: FP: 17 Command executed: 101- Start Firmware
Download: Module Flags Byte-8 Val: 0x0 Cdb Status Val: 0x0 Failure Message:
Unable to open firmware file
2025-02-19 11:10:52.188798: FP: 17 Command executed: 101- Start Firmware
Download: Module Flags Byte-8 Val: 0x0 Cdb Status Val: 0x0 Failure Message:
Unable to open firmware file
2025-02-19 11:44:14.468031: FP: 17 Command executed: 101- Start Firmware
Download: Module Flags Byte-8 Val: 0x0 Cdb Status Val: 0x40 Failure Message:
 CDB Status failed

Below values are updated after every start command
--------------------------------------------------
Command 101 Module Flags Byte-8 Val 0x0 Cdb Status Val 0x40
```

   Use this command to check the progress of start, run, and commit operations and view the contents of the queue for the group of ports.

   Note

   Use the status command and look for the following messages to ensure the download completed successfully:

```
Download completed percentage: 100
  Downloading is not in progress
  No Command is in progress
```

4. Use the **debug hardware internal linkctrl transceiver firmware fp-port 1-10,6,15-16 run** command to run the downloaded firmware on the specified range of transceiver ports.

Example

```
switch# debug hardware internal linkctrl transceiver firmware fp-port
1-10,6,15-16 run
```

Use this command to run the downloaded firmware on the specified ports.

5. Use the **debug hardware internal linkctrl transceiver firmware fp-port 1-10,6,15-16 commit** command to commit the downloaded firmware to the specified range of transceiver ports.

Example

```
switch# debug hardware internal linkctrl transceiver firmware fp-port
1-10,6,15-16 commit
```

Use this command to commit the downloaded firmware to a range of ports, making the changes permanent.

The new firmware is successfully downloaded, run, and committed to the specified transceiver ports.

4 Optionality in NX-OS Software

Topics:

- Optionality in NX-OS software
- Modular packages
- NX-OS image boot modes
- Red Hat Package Managers
- Dandified YUM commands
- Configure an FTP server and set up a local FTP YUM repository
- Create user roles for install operation

Explains how the NX-OS modular package management allows for the selective addition, removal, and upgrade of specific features without altering the base software.

Optionality in NX-OS software

This section explains NX-OS software optionality, highlighting modular package management and how this capabily delivers flexibility, a leaner footprint, and asynchronous feature delivery.

Optionality is a feature in NX-OS software that

- uses modular packages for selective feature upgrades
- supports both base and full modes, and
- enables independent upgrade or removal of optional RPMs without service disruption.

Beginning with NXOS Release 9.2(1), NX-OS software image supports modular package management. NX-OS software now provides flexibility to add, remove, and upgrade the features selectively without changing the base NX-OS software.

Using modular NX-OS software provides several advantages:

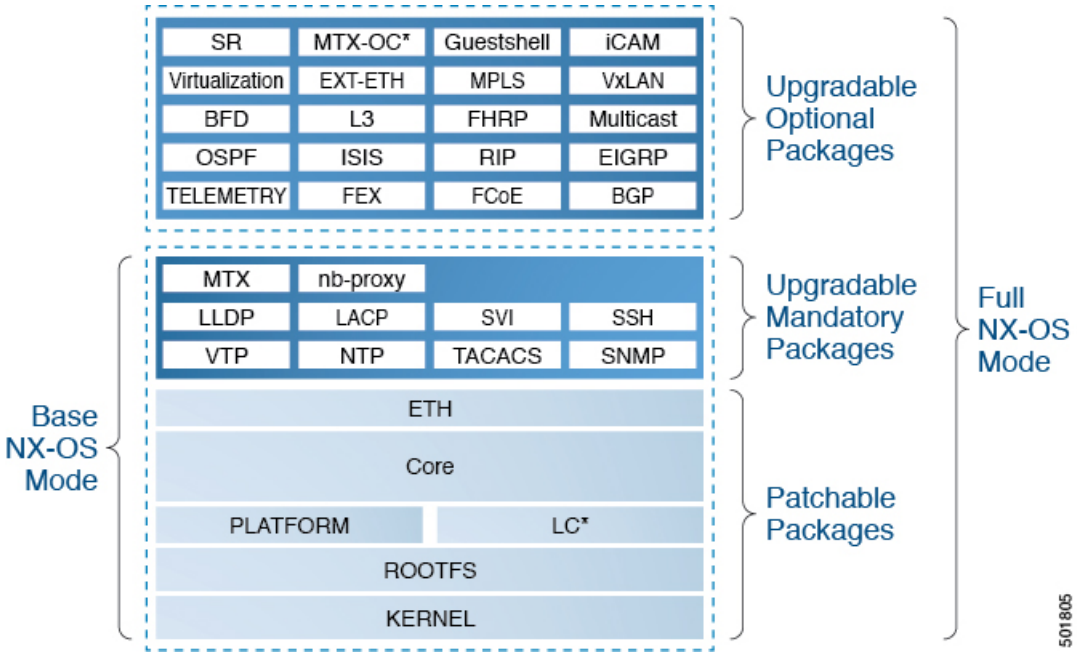
- Leaner NX-OS software
- Asynchronous delivery of the features and the fixes—provide quick fixes that are independent of the releases, including new features
- Reduced footprint of binaries and libraries at run time

Modes

NX-OS software is provisioned to boot the NX-OS software in two modes as described in the illustration:

- Base NX-OS mode
- Full NX-OS mode

Figure 2: Optionality in NX-OS software



- Base NX-OS mode contains:
  - Upgradable mandatory packages

  - Patchable packages

- Full NX-OS mode contains:
  - Upgradable optional packages
  - Upgradable mandatory packages
  - Patchable packages

> **Note**
>
> The default mode is full NX-OS mode.

In base NX-OS mode, basic Layer 2 and Layer 3 features are available. All dynamic routing features (for example, BGP, OSPF, EIGRP, RIP, and ISIS) and other optional feature RPMs are not available by default. You have to install the optional feature RPMs on top of the base image.

In full NX-OS mode, all feature RPMs are installed during boot time when Ethernet plugin is activated by the plug-in manager. There is no change in the user behavior as compared to the previous releases.

Modular packages

This section explains the architecture and management of modular packages in NX-OS, which are independent software components (RPMs) that allow for feature-specific upgrades and runtime removal.

A modular package is a software package that

- enables independent upgrades within the same release
- allows runtime removal without impacting system startup or other functions, and
- requires feature APIs to be used only after package installation.

The NX-OS software image is traditionally constructed with the packaging that forms a Linux distribution. It makes upgrading certain packages difficult as each package is large in size. This section describes a new package management for the NX-OS software image. Beginning with NX-OS Release 9.2(1), some NXOS features are considered as optional, for example, BGP, OSPF, VXLAN, MPLS, Segment Routing.

Each modular package has the following important characteristics:

- Upgrade functionality—The modular packages can be independently upgraded. The modular packages should be used from the same release as performing upgrades on these packages across multiple releases is not supported.
- Optionality—The modular packages are optional, for example, these packages can be removed or uninstalled at run time. The removal of the modular packages does not affect bringing-up the system and it does not affect any other functionality of the switches.

> **Note**
>
> All APIs exported by the modular package should be used only after the installation of the feature.

RPM and DNF

RPM (Red Hat Package Manager) is the package management system used for packaging in the Linux Standard Base (LSB). The RPM command options are grouped into three subgroups for:

- Querying and verifying packages
- Installing, upgrading, and removing packages
- Performing miscellaneous functions

rpm is the command name for the main command that is used with RPM, whereas .rpm is the extension that is used for the RPM files.

Dandified YUM (Yellowdog Updater, Modified) or DNF is an open source command-line tool for RPM based Linux systems. It allows users and system administrators to easily install, update, remove, or search software packages on the systems. DNF adds the automatic updates and the package management, including dependency management, to the RPM systems. In addition to understanding the installed packages on a system, DNF works with the repositories that are collections of the packages and they are typically accessible over a network connection.

NX-OS image boot modes

This section defines the base and full boot modes for NX-OS and provides the commands to switch between them. It explains how to reset the switch to install either a minimal set of core features or the complete software image.

You can boot the NX-OS image in base or full mode. The full boot mode installs the complete NX-OS software which is similar to the software of the previous releases. This is the default boot mode. The base boot mode has no optional RPMs installed.

To use the command line option, perform one of these steps depending on the mode.

- Use the **install reset nxos base** option to install the NX-OS image in the base boot mode using the VSH prompt. After reload, the switch is in the base mode with no optional packages installed.
- Use the **install reset nxos full** option to install the NX-OS image in the full boot mode using the VSH prompt. After reload, the switch is in the full mode with the optional packages automatically installed.

For more information, see *Using install commands for feature RPM operation* section.

Red Hat Package Managers

Learn about Red Hat Package Managers (RPMs) and how NX-OS that is built on a Linux kernel leverages the RPM framework to provide Optionality—the ability to customize the switch's software footprint by adding or removing components without needing to replace the entire system image.

You can upgrade or downgrade Red Hat Package Manager (RPM) to a new software version using NX-OS install commands or DNF commands. An upgradable RPM can be optional or mandatory.

> **Note**
>
> During the boot-up process of NX-OS, signed RPMs remain in memory while the image extraction stage takes place. However, this method is not the most efficient in terms of memory consumption. As of NX-OS Release 10.4(3)F, after the system reaches a stable state and adequate SSD space is accessible, the RPMs are transferred from memory to persistent storage. This feature is supported on N9K-C92348GC-X and all Nexus 9300 TOR switches.

See the following sections for more information about optional and mandatory RPMs.

Format of the RPM

Outlines the RPM file format and naming conventions for NX-OS feature packages, detailing the significance of each component—name, version, release, and architecture—and explaining how these elements indicate compatibility, feature changes, and distribution specifics.

This section describes the general format and naming convention of RPM files for NX-OS features.

The general format of a RPM is `<name>-<version>-<release>.<arch>.rpm`. The same format is followed for NX-OS feature RPMs.

- Name: package name, for example, BGP
- Version in <x.y.x.b> format: <major.minor.patch.build_number>, for example, 2.0.1.0
- Release: The branch from which the RPM is created, for example, 9.2.1
- Arch: The architecture type of the RPM, for example, lib32_n9000

This table provides more information on the naming convention, for example, fex-2.0.0.0-9.2.1.lib32_n9000.rpm.

Table 3: RPM naming convention

RPM Naming Convention Example: fex-2.0.0.0-9.2.1.lib32_n9000.rpm	Description
fex	Indicates the name of the component.
2	Indicates that the RPM is not backward compatible. Configuration loss takes place during an upgrade.
0	Indicates the incremental API changes/command changes/Schema changes with backward compatibility. It is applicable to the new features on top of the existing capabilities. No configuration is lost during an upgrade.
0	Indicates a bug fix without any functionality change. No configuration is lost during an upgrade.
0	This number tracks how many times the component has changed during the development cycle of a release. This value will be 0 for all the release images.

RPM Naming Convention Example: fex-2.0.0.0-9.2.1.lib32_n9000.rpm	Description
9.2.1	Indicates the release number or the distribution version for the RPM. It aligns to the NVR format. Since the feature RPM is only applicable to a NXOS release, this field has NXOS release version number present.
lib32_n9000	Indicates the architecture type of the RPM.

Optional RPMs and their associated features

The optional RPMs are the RPMs that can be installed to enable the features without affecting the native NX-OS behavior or they can be removed using the **install deactivate** command from the switch.

Optional RPMs, for example, EIGRP are not a part of the base software. They can be added, upgraded, and removed as required using either **dnf** or **install** commands from the switch.

This table contains the list of the optional RPMs and their associated features.

Table 4: List of optional RPMs and their associated features

Package Name	Associated Features
APP HOSTING	feature app-hosting
BGP	feature bgp
BFD	feature bfd
Container-tracker	feature container-tracker
EIGRP	feature eigrp
Ext-Eth	• feature openflow • feature evb • feature imp • feature netflow • feature sla_sender • feature sla_responder • feature sla twamp-server • feature sflow
EXT_ETH_LOWMEM	• feature evb • feature netflow
FCoE	• feature-set fcoe • feature-set fcoe-npv
FEX	feature-set fex

Package Name	Associated Features
FHRP	• feature hsrp • feature vrrpv3
HW TELEMETRY	feature hw telemetry
iCAM	feature icam
ISIS	feature isis
MPLS	• feature mpls segment-routing • feature mpls evpn
Multicast	• feature pim • feature pim6 • feature msdp • feature ngmvpn
NIA	NA
NXSDK	NA
OSPF	• feature ospf • feature ospfv3
RIP	feature rip
SDAA	NA
Services	feature catena
SR	feature mpls segment-routing traffic-engineering
TELEMETRY	feature telemetry
Virtualization	NA
VM Tracker	feature vmtracker
VXLAN	• feature nv overlay • feature fabric forwarding

Guidelines for NX-OS feature RPM installation

This topic describes the guidelines for installing NX-OS feature RPMs using system repositories and supported commands.

The NX-OS system RPM repositories are present in NX-OS Series switches for RPM management.

Note

Avoid manually copying the RPMs to system repositories. Instead use the install or DNF commands.

Table 5: RPM repositories that are present in the switches

Repository Name	Repository Path	Description
groups-repo	/rpms	Part of the bundled NX-OS image. It is used to keep all the RPMs that are bundled as part of the NX-OS image. All RPMs based in this repository are known as base RPMs.
localdb	/bootflash/.rpmstore/patching/localrepo	Used for RPM persistency. When a user adds a NX-OS feature RPM as part of **install add** command, the RPM is copied to this location and it is persisted during the reloads. User has the responsibility to clean the repository. To add a RPM to this repository, use **install add** command. To remove a RPM from this repository, use **install remove** command. DNF commands can be used to populate the repository too. The maximum space for the repository is 200Mb along with the patching repository for Nexus 9000 Series switches except Nexus 3000 Series switches. For Nexus 3000 Series switches, the maximum space for the repository is 20 Mb only.
patching	/bootflash/.rpmstore/patching/patchrepo	Used for RPM persistency. When a user adds a NX-OS patch RPM to the switch, the patch RPM is copied to this repository.
thirdparty	/bootflash/.rpmstore/thirdparty	Used for RPM persistency when a user adds a third party RPM.

The `groups-repo` and `localdb` repositories hold the NX-OS feature RPMs that should be installed during the system boot or during activation. DNF commands or **install** command can be used for the installation or the removal of these RPMs.

The listed rules are applied to the feature RPM installation procedure during boot or install time:

- Only RPMs with the same NX-OS release number should be selected for the installation.

- Base RPMs cannot be added to the `localdb` repository.

Guidelines for third-party RPM installation

Guidelines for installing third-party RPM packages on the device.

In releases prior to 10.1(x), you can install any third-party package on the device, even if it is not provided or signed by Cisco.

Starting with release 10.1(x) any third-party package that is not signed by Cisco is not allowed to be installed on the device. However, if you wish to bypass this and install the software, you can configure the device to enable the third-party software installation. The configuration persists as a normal configuration and can be verified by using the **running-config** command. Following this configuration, you can install any third-party software with the known risks.

Install command options for feature and third-party RPMs

Use the reference table for using install commands for the feature RPM operations.

Table 6: Reference for install commands for the feature RPM operations

Command	Description

install reset	This operation removes all the patches, persisted configurations, upgraded packages, third-party installed packages, unsaved configurations, and reloads the switch's previous mode (Full/Base) with the default packages.
	The **install reset** command also performs write erase operation. The following message is displayed at the prompt:
	`switch(config)# install reset`
	`================================================`
	`WARNING!!This operation will remove all pactches, upgraded packages, persisted etc configs, third party packages installed, startup configuration(write erase) and reload the switch with default packages.`
	`================================================`
	`Do you want to proceed with reset operation? (y/n)? [n]`

install reset nxos base	This operation installs NX-OS in base mode by removing all patches, upgraded packages, persisted etc configurations, third-party packages installed, startup configuration (write erase), and reloads the switch with the default packages.
install reset nxos full	This operation installs NX-OS with full mode by removing all patches, upgraded packages, persisted etc configs, third-party packages installed, startup configuration (write erase), and reloads the switch with the default packages (with mandatory and optional RPMs).

Command	Description
install add <>	Adds an RPM file to the respective repository and updates the repository (**patch/feature/third-party**).
install activate <*rpm name*>	Installs an RPM that is present in the repository.
install commit <*rpm name*>	Used for the patch RPMs. Makes the patch persist during the reload.
install deactivate <*rpm name*>	Uninstalls an RPM. Beginning with NX-OS Release 10.1(1), when you use this command to deactivate RPMs, the options to either downgrade to the base version of RPM or to uninstall RPM appear. You can select the option that you desire and the operation will proceed.
install remove <*rpm name*>	Removes an RPM file from the repository and updates the repository.
sh install active	Displays the list of the installed RPMs in the system apart from base rootfs RPMs. (features/patch/third-party).
sh install inactive	Displays the list of the RPMs that are present in the repository but they are not installed.
sh install packages	Lists all the RPMs that are installed including rootfs RPMs.

Command	Description

[no] system software allow third-party	Beginning with NX-OS Release 10.1(1) the third-party RPM installations are not allowed to be installed on the device by default. This command bypasses this restriction and configures the device to enable the third-party software installation.
	
	The following command shows the error message when you activate third-party RPM without applying the third-party configuration:
	
	`switch(config)# **install activate pbwMonitor-1.0-1.5.0.x86_64.rpm**`
	
	`Install operation 193 failed because package is not signed by Cisco.Enable TPS installation using 'system software allow third-party' CLI at Tue Nov 17 04:23:10 2020`
	
	The following command shows activating third-party RPM installations after applying the configuration:
	
	`switch(config)# **system software allow third-party**`
	`switch(config)# 2020 Nov 17 04:25:41 switch %$ VDC-1 %$`
	`%USER-2-SYSTEM_MSG: <<%PATCH-INSTALLER-2-TPS_FEATURE_ENABLED>> User has enabled TPS installation - patch_installer`
	
	`switch(config)# **install activate pbwMonitor-1.0-1.5.0.x86_64.rpm**`
	`[####################] 100%`
	`Install operation 194 completed successfully at Tue Nov 17 04:25:58 2020`
	
	The following command shows disabling the third-party configuration:
	
	`switch(config)# **no system software allow third-party**`
	`switch(config)# 2020 Nov 17 04:27:17 switch %$ VDC-1 %$`
	`%USER-2-SYSTEM_MSG: <<%PATCH-INSTALLER-2-TPS_FEATURE_DISABLED>> User has disabled TPS installation - patch_installer`

> **Note**
>
> If you are using ISSU or upgrading to NX-OS Release 10.1.1 release from an earlier version, you must manually apply the third-party configuration within the first 30 minutes after the upgrade to ensure the third-party RPMs get installed.

Use install commands for digital signature support

Use the install commands for digital signature support.

Use the install commands for digital signature support.

Procedure

1. Import and add a GPG (GNU Privacy Guard) key from a file located in the bootflash using the **install add bootflash:**<keyfile> **gpg-key** command.

   Example

   ```
   switch# install add bootflash:RPM-GPG-KEY-puppetlabs gpg-key
   [####################] 100%
   Install operation 304 completed successfully at Thu Jun 19 16:40:28 2018
   ```

   Release RPMs are signed with GPG (GNU Privacy Guard) key. The public GPG key is present at **/etc/pki/rpm-gpg/arm-Nexus9k-rel.gpg**. To add other public keys from different sources, use the steps in this section.

2. Use one of the two steps to verify whether the RPM file is a signed or non-signed file.
   a) Verify that the package is a signed file using the **install verify package** <package-name> command.
   b) Verify that the RPM file is a signed file using the **install verify bootflash:**<RPM file> command.

   Example

   ```
   switch# install verify bootflash:vxlan-2.0.0.0-9.2.1.lib32_n9000.rpm

   RSA signed
   switch#
   ```

Query all installed RPMs

Query all the installed RPMs using the **show install packages** command.

Perform this step to query all the installed RPMs

Procedure

Query all the installed RPMs using the **show install packages** command.

Example

```
switch# show install packages

Boot Image:
NXOS Image: bootflash:/nxos.9.2.1.bin

--------------------------------------------------
Installed Packages
attr.x86_64 2.4.47-r0.0 installed Unsigned
aufs-util.x86_64 3.14+git0+b59a2167a1-r0.0 installed Unsigned
base-files.n9000 3.0.14-r89.0 installed Unsigned
base-passwd.lib32_x86 3.5.29-r0.1.0 installed Unsigned
bash.lib32_x86 4.3.30-r0.0 installed Unsigned
bfd.lib32_n9000 2.0.0.0-9.2.1 installed Signed
bgp.lib32_n9000 2.0.0.0-9.2.1 installed Signed
binutils.x86_64 2.25.1-r0.0 installed Unsigned
bridge-utils.x86_64 1.5-r0.0 installed Unsigned
busybox.x86_64 1.23.2-r0.0 installed Unsigned
busybox-udhcpc.x86_64 1.23.2-r0.0 installed Unsigned
bzip2.x86_64 1.0.6-r5.0 installed Unsigned
ca-certificates.all 20150426-r0.0 installed Unsigned
cgroup-lite.x86_64 1.1-r0.0 installed Unsigned
chkconfig.x86_64 1.3.58-r7.0 installed Unsigned
container-tracker.lib32_n9000 2.0.0.0-9.2.1 installed Signed
containerd-docker.x86_64 0.2.3+gitaa8187dbd3b7ad67d8e5e3a15115d3eef43a7ed1-r0.0
installed Unsigned
core.lib32_n9000 2.0.0.0-9.2.1 installed Signed
coreutils.lib32_x86 8.24-r0.0 installed Unsigned
cpio.x86_64 2.12-r0.0 installed Unsigned
cracklib.lib32_x86 2.9.5-r0.0 installed Unsigned
cracklib.x86_64 2.9.5-r0.0 installed Unsigned
createrepo.x86_64 0.4.11-r9.0 installed Unsigned
cronie.x86_64 1.5.0-r0.0 installed Unsigned
curl.lib32_x86 7.60.0-r0.0 installed Unsigned
db.x86_64 6.0.30-r0.0 installed Unsigned
dbus-1.lib32_x86 1.8.20-r0.0 installed Unsigned
dhcp-client.x86_64 4.3.2-r0.0 installed Unsigned
dhcp-server.x86_64 4.3.2-r0.0 installed Unsigned
switch#
```

Install RPMs using the one-step procedure

The commands for both install and upgrade RPMs are the same. Use this one-step procedure to install the RPMs.

The commands for both install and upgrade RPMs are the same. Use this one-step procedure to install the RPMs.

Procedure

1. Install and activate the RPM using the **install add** <*rpm*> **activate** command.

   Example

   ```
   switch# install add bootflash:chef.rpm activate
   Adding the patch (/chef.rpm)
   [####################] 100%
   Install operation 868 completed successfully at Tue May  8 11:20:10 2018
   ```

```
Activating the patch (/chef.rpm)
[####################] 100%
Install operation 869 completed successfully at Tue May  8 11:20:20 2018
```

2. Verify the output of the **show install active** command.

Example

```
switch# show install active
Boot Image:
        NXOS Image: bootflash:/nxos.9.2.1.bin

Active Packages:
bgp-2.0.1.0-9.2.1.lib32_n9000
chef-12.0.0alpha.2+20150319234423.git.1608.b6eb10f-1.el5.x86_64

Active Base Packages:
        lacp-2.0.0.0-9.2.1.lib32_n9000
        lldp-2.0.0.0-9.2.1.lib32_n9000
        mtx-device-2.0.0.0-9.2.1.lib32_n9000
        mtx-grpc-agent-2.0.0.0-9.2.1.lib32_n9000
        mtx-infra-2.0.0.0-9.2.1.lib32_n9000
        mtx-netconf-agent-2.0.0.0-9.2.1.lib32_n9000
        mtx-restconf-agent-2.0.0.0-9.2.1.lib32_n9000
        mtx-telemetry-2.0.0.0-9.2.1.lib32_n9000
        ntp-2.0.0.0-9.2.1.lib32_n9000
        nxos-ssh-2.0.0.0-9.2.1.lib32_n9000
        snmp-2.0.0.0-9.2.1.lib32_n9000
        svi-2.0.0.0-9.2.1.lib32_n9000
        tacacs-2.0.0.0-9.2.1.lib32_n9000
        vtp-2.0.0.0-9.2.1.lib32_n9000
```

Install RPMs using the two-step procedure

The commands for both install and upgrade RPMs are the same. Use this two-step procedure to install the RPMs.

The commands for both install and upgrade RPMs are the same. Use this two-step procedure to install the RPMs.

Procedure

1. Install the RPM using the **install add** <*rpm*> command.

Example

```
switch# install add bootflash:vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm

[####################] 100%
Install operation 892 completed successfully at Thu Jun  7 13:56:38 2018
```

2. Verify using the **show install inactive** command.

Example

```
switch(config)# show install inactive | grep vxlan
vxlan-2.0.1.0-9.2.1.lib32_n9000
```

3. Activate the RPM using the **install activate** <*rpm*> command.

Example

```
switch# install activate vxlan

[####################] 100%
Install operation 891 completed successfully at Thu Jun  7 13:53:07 2018
```

4. Verify using the **show install active** command.

Example

```
switch# show install active | grep vxlan
vxlan-2.0.0.0-9.2.1.lib32_n9000
switch# show install inactive | grep vxlan
switch#
```

Upgrade the RPMs

Perform this procedure to upgrade the RPMs using a single command.

The commands for both install and upgrade RPMs are the same. Perform this procedure to upgrade the RPMs:

Procedure

1. Install the RPM using the **install add** <*rpm*>**activate upgrade** command.

Example

```
switch(config)# install add bootflash:bgp-2.0.2.0-9.2.1.lib32_n9000.rpm
activate upgrade

Adding the patch (/bgp-2.0.2.0-9.2.1.lib32_n9000.rpm)
[####################] 100%
Install operation 870 completed successfully at Tue May 8 11:22:30 2018

Activating the patch (/bgp-2.0.2.0-9.2.1.lib32_n9000.rpm)
[####################] 100%
Install operation 871 completed successfully at Tue May 8 11:22:40 2018
```

2. Verify the output using the **show install active** command.

Example

```
switch(config)# show install active

Boot Image:
```

```
NXOS Image: bootflash:/nxos.9.2.1.bin

Active Packages:
bgp-2.0.2.0-9.2.1.lib32_n9000
chef-12.0.0alpha.2+20150319234423.git.1608.b6eb10f-1.el5.x86_64

Active Base Packages:
lacp-2.0.0.0-9.2.1.lib32_n9000
lldp-2.0.0.0-9.2.1.lib32_n9000
mtx-device-2.0.0.0-9.2.1.lib32_n9000
mtx-grpc-agent-2.0.0.0-9.2.1.lib32_n9000
mtx-infra-2.0.0.0-9.2.1.lib32_n9000
mtx-netconf-agent-2.0.0.0-9.2.1.lib32_n9000
mtx-restconf-agent-2.0.0.0-9.2.1.lib32_n9000
mtx-telemetry-2.0.0.0-9.2.1.lib32_n9000
ntp-2.0.0.0-9.2.1.lib32_n9000
nxos-ssh-2.0.0.0-9.2.1.lib32_n9000
snmp-2.0.0.0-9.2.1.lib32_n9000
svi-2.0.0.0-9.2.1.lib32_n9000
tacacs-2.0.0.0-9.2.1.lib32_n9000
vtp-2.0.0.0-9.2.1.lib32_n9000
```

Downgrade RPMs

Downgrade the RPMs using the one-step procedure with the required command attribute.

The downgrade procedure needs a special command attribute. Downgrade the RPMs using the one-step procedure.

Procedure

1. Downgrade the RPM using the **install add** <*rpm*>**activate downgrade** command.

   Example

```
switch(config)# install add bootflash:bgp-2.0.1.0-9.2.1.lib32_n9000.rpm
activate downgrade


Adding the patch (/bgp-2.0.1.0-9.2.1.lib32_n9000.rpm)
[####################] 100%
Install operation 872 completed successfully at Tue May 8 11:24:43 2018

Activating the patch (/bgp-2.0.1.0-9.2.1.lib32_n9000.rpm)
[####################] 100%
Install operation 873 completed successfully at Tue May 8 11:24:52 2018
```

2. Verify the output using the **show install active** command.

   Example

```
switch(config)# show install active
Boot Image:
 NXOS Image: bootflash:/nxos.9.2.1.bin

Active Packages:
```

```
 bgp-2.0.1.0-9.2.1.lib32_n9000
 chef-12.0.0alpha.2+20150319234423.git.1608.b6eb10f-1.el5.x86_64

Active Base Packages:
 lacp-2.0.0.0-9.2.1.lib32_n9000
 lldp-2.0.0.0-9.2.1.lib32_n9000
 mtx-device-2.0.0.0-9.2.1.lib32_n9000
 mtx-grpc-agent-2.0.0.0-9.2.1.lib32_n9000
 mtx-infra-2.0.0.0-9.2.1.lib32_n9000
 mtx-netconf-agent-2.0.0.0-9.2.1.lib32_n9000
 mtx-restconf-agent-2.0.0.0-9.2.1.lib32_n9000
 mtx-telemetry-2.0.0.0-9.2.1.lib32_n9000
 ntp-2.0.0.0-9.2.1.lib32_n9000
 nxos-ssh-2.0.0.0-9.2.1.lib32_n9000
 snmp-2.0.0.0-9.2.1.lib32_n9000
 svi-2.0.0.0-9.2.1.lib32_n9000
 tacacs-2.0.0.0-9.2.1.lib32_n9000
 vtp-2.0.0.0-9.2.1.lib32_n9000
switch(config)#
```

Uninstall the RPMs

Perform this procedure to uninstall the RPMs.

Perform this procedure to uninstall the RPMs.

Procedure

Downgrade to the base version of RPM, if one exists in the groups-repo (/rpms), or uninstall the RPM completely from the switch using the **install deactivate** <*rpm*> command.

- To downgrade to the base version, enter **y**.
- To completely uninstall the RPM, enter **n** in the command prompt.

Example

```
switch(config)# install deactivate bgp
Base RPM found. Do you want to downgrade to base version(y/n) [n] y
Downgrading to the base version
[####################] 100%
Install operation 190 completed successfully at Tue Nov 17 04:10:40 2020
```

Example

```
switch(config)# install deactivate bgp
Base RPM found. Do you want to downgrade to base version(y/n) [n] n

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

 WARNING!!
 This operation will remove 'bgp-3.0.0.0-9.4.1.lib32_n9000' related
configuration from running-configuration
 on successful completion. Update startup-configuration accordingly.
================================================================================
```

```
[####################] 100%
Install operation 9 completed successfully at Tue Nov 17 05:05:59 2020
```

Remove the RPMs

Perform this procedure to remove the RPMs.

Perform this procedure to remove the RPMs.

Procedure

Remove the RPM from the repository using the **install remove** <*rpm*> command.

Example

```
switch(config)# show install inactive | grep vxlan

vxlan-2.0.0.0-9.2.1.lib32_n9000
switch(config)# install remove vxlan

Proceed with removing vxlan? (y/n)?  [n] y
[####################] 100%
Install operation 890 Removal of base rpm package is not permitted  at Thu Jun
  7 13:52:15 2018
```

Dandified YUM commands

This section explains the Dandified YUM (DNF)commands that replaced the YUM commands beginning with NX-OS Release 10.1(x).

In Nexus 9000 switches, DNF (Dandified YUM) is the package manager used to manage modular software components (RPMs) within the NX-OS environment. Beginning with NX-OS Release 10.1(x), DNF replaced YUM as the primary tool for the Optionality feature, allowing you to install, upgrade, or remove specific features (such as Layer 3, BGP, or OSPF) without performing a full system binary upgrade or reloading the switch.

Note

DNF commands do not support CTRL+C. Install commands do support CTRL+C. If DNF commands are aborted using CTRL+C, manual cleanup must be performed using **/isan/bin/patching_utils.py --unlock**.

Package operations with DNF commands

This section describes how to perform package operations using the DNF commands.

This section describes how to perform package operations using the DNF commands.

This section describes how to perform package operations using the DNF commands:

Note

- DNF commands are accessed only from the BASH shell on the box and they are not allowed from the NX-OS VSH terminal.
- Make sure that as a sudo user, you have access to the super user privileges.

Find the base version RPM of the image

Use the **ls /rpms** command to find the base version RPM of the image.

The base RPM version is the pre-installed RPM that is archived in the system image.

Use the **ls /rpms** command to find the base version RPM of the image.

```
switch# ls /rpms

bfd-2.0.0.0-9.2.1.lib32_n9000.rpm
ins_tor_sdk_t2-1.0.0.0-9.2.0.77.lib32_n9000.rpm
mtx-netconf-agent-2.0.0.0-9.2.1.lib32_n9000.rpm
snmp-2.0.0.0-9.2.1.lib32_n9000.rpm
bgp-2.0.0.0-9.2.1.lib32_n9000.rpm
ins_tor_sdk_t3-1.0.0.0-9.2.0.77.lib32_n9000.rpm
mtx-restconf-agent-2.0.0.0-9.2.1.lib32_n9000.rpm
sr-2.0.0.0-9.2.1.lib32_n9000.rpm
container-tracker-2.0.0.0-9.2.1.lib32_n9000.rpm
isis-2.0.0.0-9.2.1.lib32_n9000.rpm
mtx-telemetry-2.0.0.0-9.2.1.lib32_n9000.rpm
svi-2.0.0.0-9.2.1.lib32_n9000.rpm
eigrp-2.0.0.0-9.2.1.lib32_n9000.rpm
lacp-2.0.0.0-9.2.1.lib32_n9000.rpm
nbproxy-2.0.0.0-9.2.1.lib32_n9000.rpm
tacacs-2.0.0.0-9.2.1.lib32_n9000.rpm
ext-eth-2.0.0.0-9.2.1.lib32_n9000.rpm
lldp-2.0.0.0-9.2.1.lib32_n9000.rpm
ntp-2.0.0.0-9.2.1.lib32_n9000.rpm
telemetry-2.3.4.0-9.2.1.lib32_n9000.rpm
fcoe-2.0.0.0-9.2.1.lib32_n9000.rpm
mcast-2.0.0.0-9.2.1.lib32_n9000.rpm
nxos-ssh-2.0.0.0-9.2.1.lib32_n9000.rpm
virtualization-2.0.0.0-9.2.1.lib32_n9000.rpm
fex-2.0.0.0-9.2.1.lib32_n9000.rpm
mpls-2.0.0.0-9.2.1.lib32_n9000.rpm
ospf-2.0.0.0-9.2.1.lib32_n9000.rpm
vtp-2.0.0.0-9.2.1.lib32_n9000.rpm
fhrp-2.0.0.0-9.2.1.lib32_n9000.rpm
mtx-device-2.0.0.0-9.2.1.lib32_n9000.rpm           repodata
                        vxlan-2.0.0.0-9.2.1.lib32_n9000.rpm
guestshell-2.0.0.0-9.2.1.lib32_n9000.rpm
mtx-grpc-agent-2.0.0.0-9.2.1.lib32_n9000.rpm
rip-2.0.0.0-9.2.1.lib32_n9000.rpm
icam-2.0.0.0-9.2.1.lib32_n9000.rpm
mtx-infra-2.0.0.0-9.2.1.lib32_n9000.rpm
services-2.0.0.0-9.2.1.lib32_n9000.rpm
```

Check the list of the installed RPMs

Use the **dnf list installed** command to query the feature and third party RPMs and grep a specific RPM.

Use the **dnf list installed** command to query the feature and third party RPMs and grep a specific RPM.

Here is an example for feature RPMs.

```
bash-4.2# dnf list installed | grep lib32_n9000

bfd.lib32_n9000                          2.0.0.0-9.2.1           @groups-repo
core.lib32_n9000                         2.0.0.0-9.2.1           installed
eth.lib32_n9000                          2.0.0.0-9.2.1           installed
guestshell.lib32_n9000                   2.0.0.0-9.2.1           @groups-repo
lacp.lib32_n9000                         2.0.0.0-9.2.1           installed
linecard2.lib32_n9000                    2.0.0.0-9.2.1           installed
lldp.lib32_n9000                         2.0.0.0-9.2.1           installed
mcast.lib32_n9000                        2.0.0.0-9.2.1           @groups-repo
mtx-device.lib32_n9000                   2.0.0.0-9.2.1           installed
mtx-grpc-agent.lib32_n9000               2.0.0.0-9.2.1           installed
mtx-infra.lib32_n9000                    2.0.0.0-9.2.1           installed
mtx-netconf-agent.lib32_n9000            2.0.0.0-9.2.1           installed
mtx-restconf-agent.lib32_n9000           2.0.0.0-9.2.1           installed
mtx-telemetry.lib32_n9000                2.0.0.0-9.2.1           installed
nbproxy.lib32_n9000                      2.0.0.0-9.2.1           installed
ntp.lib32_n9000                          2.0.0.0-9.2.1           installed
nxos-ssh.lib32_n9000                     2.0.0.0-9.2.1           installed
ospf.lib32_n9000                         2.0.0.0-9.2.1           @groups-repo
platform.lib32_n9000                     2.0.0.0-9.2.1           installed
snmp.lib32_n9000                         2.0.0.0-9.2.1           installed
svi.lib32_n9000                          2.0.0.0-9.2.1           installed
tacacs.lib32_n9000                       2.0.0.0-9.2.1           installed
tor.lib32_n9000                          2.0.0.0-9.2.0.77        installed
virtualization.lib32_n9000               2.0.1.0-9.2.1           @localdb
vtp.lib32_n9000                          2.0.0.0-9.2.1           installed
vxlan.lib32_n9000                        2.0.0.0-9.2.1           @groups-repo
...
```

Get details of the installed RPMs

Introduces the use of yum and dnf info commands to retrieve detailed information about installed RPMs, including package metadata, version, repository source, and description.

The **dnf info** <*rpmname*> command lists out the detailed information of the installed RPM.

dnf info vxlan

```
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
 protect-packages
groups-repo

                                                    | 1.1 kB     00:00 ...
localdb

                                                    |  951 B     00:00 ...
patching

                                                    |  951 B     00:00 ...
thirdparty

                                                    |  951 B     00:00 ...
Installed Packages
Name        : vxlan
Arch        : lib32_n9000
Version     : 2.0.0.0
Release     : 9.2.1
```

```
Size        : 6.4 M
Repo        : installed
From repo   : groups-repo
Summary     : Cisco NXOS VxLAN
URL         : http://cisco.com/
License     : Proprietary
Description : Provides VxLAN support
```

Install RPMs

Installing the RPMs downloads the RPMs and copies the respective program to the switches.

Installing the RPMs downloads the RPMs and copies the respective program to the switches. This is an example for installing the RPMs from a remote server (that is reachable in the network).

```
bash-4.3# dnf install http://10.0.0.2/modularity/rpms/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
 protect-packages
groups-repo
                                                                   | 1.1 kB
 00:00 ...
localdb
                                                                   |  951 B
 00:00 ...
localdb/primary
                                                                   |  886 B
 00:00 ...
localdb


       1/1
patching
                                                                   |  951 B
 00:00 ...
thirdparty
                                                                   |  951 B
 00:00 ...
Setting up Install Process
vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm
                                                         | 1.6 MB     00:00

Examining /var/tmp/yum-root-RaANgb/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm:
vxlan-2.0.1.0-9.2.1.lib32_n9000
Marking /var/tmp/yum-root-RaANgb/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm to be
installed
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

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

 Package                  Arch                          Version
                        Repository                                      Size
================================================================================
Installing:
vxlan                   lib32_n9000                  2.0.1.0-9.2.1
                /vxlan-2.0.1.0-9.2.1.lib32_n9000                     6.4 M
Transaction Summary
```

```
================================================================================
Install       1 Package

Total size: 6.4 M
Installed size: 6.4 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : vxlan-2.0.1.0-9.2.1.lib32_n9000

  1/1
starting pre-install package version mgmt for vxlan
pre-install for vxlan complete
starting post-install package version mgmt for vxlan
post-install for vxlan complete

Installed:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1



Complete!
```

This is an example for installing the RPMs from local bootflash.

sudo dnf install /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm

```
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
 protect-packages
groups-repo

                                                | 1.1 kB     00:00 ...
localdb

                                                |  951 B     00:00 ...
patching

                                                |  951 B     00:00 ...
thirdparty

                                                |  951 B     00:00 ...
Setting up Install Process
Examining /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm:
vxlan-2.0.1.0-9.2.1.lib32_n9000
Marking /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm as an update to
vxlan-2.0.0.0-9.2.1.lib32_n9000
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

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

 Package                                 Arch
           Version                                                  Repository
                                                                 Size
```

```
================================================================================
Updating:
 vxlan                                    lib32_n9000
           2.0.1.0-9.2.1
/vxlan-2.0.1.0-9.2.1.lib32_n9000                                      6.4 M

Transaction Summary
================================================================================
Upgrade       1 Package

Total size: 6.4 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating   : vxlan-2.0.1.0-9.2.1.lib32_n9000

                                                                      1/2
starting pre-install package version mgmt for vxlan
pre-install for vxlan complete
starting post-install package version mgmt for vxlan
post-install for vxlan complete
  Cleanup    : vxlan-2.0.0.0-9.2.1.lib32_n9000

                                                                      2/2

Updated:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1



Complete!
```

This is an example for installing the RPM if it is available in a repository.

```
dnf install eigrp
```

Upgrade RPMs

Examples for upgrading RPMs from remote server, local bootflash, or repository.

This topic provides examples for upgrading RPMs from a remote server (that is reachable in the network).

```
bash-4.3# dnf upgrade http://10.0.0.2/modularity/rpms/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
 protect-packages
groups-repo
                                                                    | 1.1 kB
 00:00 ...
localdb
                                                                    |  951 B
 00:00 ...
patching
                                                                    |  951 B
 00:00 ...
thirdparty
                                                                    |  951 B
 00:00 ...
```

```
Setting up Upgrade Process
vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm
                                                              | 1.6 MB     00:00

Examining /var/tmp/yum-root-RaANgb/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm:
vxlan-2.0.1.0-9.2.1.lib32_n9000
Marking /var/tmp/yum-root-RaANgb/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm as an
update to vxlan-2.0.0.0-9.2.1.lib32_n9000
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

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

 Package                   Arch                              Version
                         Repository                                        Size
================================================================================
Updating:
vxlan                     lib32_n9000                     2.0.1.0-9.2.1
                  /vxlan-2.0.1.0-9.2.1.lib32_n9000                     6.4 M
Transaction Summary
================================================================================
Upgrade       1 Package

Total size: 6.4 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
** Found 1 pre-existing rpmdb problem(s), 'yum check' output follows:
busybox-1.23.2-r0.0.x86_64 has missing requires of busybox-syslog
  Updating   : vxlan-2.0.1.0-9.2.1.lib32_n9000

  1/2
starting pre-install package version mgmt for vxlan
pre-install for vxlan complete
starting post-install package version mgmt for vxlan
post-install for vxlan complete
  Cleanup    : vxlan-2.0.0.0-9.2.1.lib32_n9000

  2/2

Updated:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1



Complete!
```

This is an example for upgrading the RPMs from local bootflash.

```
sudo dnf upgrade /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
 protect-packages
groups-repo
```

```
                                                    | 1.1 kB     00:00 ...
localdb
                                                    |  951 B     00:00 ...
patching
                                                    |  951 B     00:00 ...
thirdparty
                                                    |  951 B     00:00 ...
Setting up Upgrade Process
Examining /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm:
vxlan-2.0.1.0-9.2.1.lib32_n9000
Marking /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm as an update to
vxlan-2.0.0.0-9.2.1.lib32_n9000
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

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

 Package                                  Arch
           Version                                                 Repository
                                                                 Size
================================================================================
Updating:
vxlan                                   lib32_n9000
          2.0.1.0-9.2.1
/vxlan-2.0.1.0-9.2.1.lib32_n9000                                        6.4 M

Transaction Summary
================================================================================
Upgrade       1 Package

Total size: 6.4 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating   : vxlan-2.0.1.0-9.2.1.lib32_n9000

                                                                    1/2
starting pre-install package version mgmt for vxlan
pre-install for vxlan complete
starting post-install package version mgmt for vxlan
post-install for vxlan complete
  Cleanup    : vxlan-2.0.0.0-9.2.1.lib32_n9000

                                                                    2/2

Updated:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1
```

```
Complete!
```

This is an example for upgrading the RPMs if it is available in any repository.

```
dnf upgrade eigrp
```

Downgrade RPMs

This topic provides example for downgrading the RPMs from a remote server (that is reachable in the network).

This topic provides example for downgrading the RPMs from a remote server (that is reachable in the network).

```
sudo dnf
 downgrade vxlan-2.0.0.0-9.2.1.lib32_n9000

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
 protect-packages
Setting up Downgrade Process
groups-repo

                                                 | 1.1 kB     00:00 ...
localdb

                                                 |  951 B     00:00 ...
localdb/primary

                                                 | 1.3 kB     00:00 ...
localdb

                                                                     2/2
patching

                                                 |  951 B     00:00 ...
thirdparty

                                                 |  951 B     00:00 ...
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.0.0-9.2.1 will be a downgrade
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be erased
--> Finished Dependency Resolution

Dependencies Resolved

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

 Package                                    Arch
                        Version
    Repository                                              Size
================================================================================
Downgrading:
vxlan                                      lib32_n9000
                       2.0.0.0-9.2.1
    groups-repo                                            1.6 M
Transaction Summary
================================================================================
Downgrade     1 Package

Total download size: 1.6 M
Is this ok [y/N]: y
```

```
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : vxlan-2.0.0.0-9.2.1.lib32_n9000

                                                                1/2
starting pre-install package version mgmt for vxlan
pre-install for vxlan complete
starting post-install package version mgmt for vxlan
post-install for vxlan complete
  Cleanup    : vxlan-2.0.1.0-9.2.1.lib32_n9000

                                                                2/2

Removed:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1



Installed:
  vxlan.lib32_n9000 0:2.0.0.0-9.2.1



Complete!
```

This is an example for downgrading the RPMs from local bootflash.

```
dnf downgrade /bootflash/eigrp-2.0.0-9.2.1.lib32_n9000.rpm
```

This is an example for downgrading the RPMs if it is available in any repository.

```
dnf downgrade eigrp
```

Delete RPMs

Deleting the RPMs de-installs the RPMs and removes any configuration commands of the feature.

Deleting the RPMs de-installs the RPMs and removes any configuration commands of the feature. Use the **dnf erase** <*rpm*> command to delete the RPMs.

```
bash-4.2# sudo dnf erase vxlan

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
 protect-packages
Setting up Remove Process
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be erased
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
Package   Arch   Version  RepositorySize
================================================================================
Removing:
vxlan     lib32_n9000    2.0.1.0-9.2.1    @/vxlan-2.0.1.0-9.2.1.lib32_n9000
```

```
6.4 M
Transaction Summary
================================================================================
Remove1 Package

Installed size: 6.4 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
Erasing    : vxlan-2.0.1.0-9.2.1.lib32_n9000 1/1
starting pre-remove package version mgmt for vxlan
pre-remove for vxlan complete

Removed:
vxlan.lib32_n9000 0:2.0.1.0-9.2.1

Complete!
```

Support for DNF groups

The support for DNF groups is part of the package management.

The DNF group is a feature that

- simplifies the management of the packages for the administrators
- provides greater flexibility, and
- allows administrators to manage collections of packages as logical groups.

The support for DNF groups is part of the package management.

The administrators can group a list of packages (RPMs) into a logical group and they can perform various operations. The group commands that DNFsupports include

- **grouplist**
- **groupinfo**
- **groupinstall**
- **groupremove**, and
- **groupupdate**.

The DNF groups can be broadly classified as Layer 2, Layer 3, routing, and management.

Grouplist command for listing available package groups

In Linux, you can install a group of related packages using the **dnfgrouplist** command instead of installing individual packages.

In Linux, number of packages are bundled to particular group. Instead of installing individual packages with dnf, you can install particular group that will install all the related packages that belongs to the group. For example to list all the available groups, use the **dnf grouplist** command:

List all available package groups in Linux using the **dnfgrouplist** command.

```
bash-4.4# dnf grouplist
Last metadata expiration check: 0:00:00 ago on Fri 08 Mar 2024 12:26:33 PM
UTC.                             ] ---  B/s |   0  B     --:-- ETA
Available Groups:
   management
   routing
   L2
   L3
bash-4.4#
```

Groupmembers command for displaying package group contents

Use the **dnf groupinfo** command to display the description and the contents of a package group.

Display the description and the contents of a package group using the **dnf groupinfo** command.

The command lists out the feature members of the group.

```
bash-4.4# dnf groupinfo l2

Last metadata expiration check: 0:00:00 ago on Fri 08 Mar 2024 12:27:44 PM
UTC.                             ] ---  B/s |   0  B     --:-- ETA

Group: L2
 Mandatory Packages:
   lacp
   lldp
   svi
   vtp
bash-4.4#
```

Groupinstall command for install and upgrade of member RPM

This command is for both install and upgrade of the members RPM.

This command is for both install and upgrade of the members RPM. If the member is not installed, it installs the highest version available. If the member is already installed and higher RPM is available, it upgrades that member.

```
bash-4.4# dnf groupinstall l3
Last metadata expiration check: 0:00:00 ago on Fri 08 Mar 2024 12:38:05 PM
UTC.                             ] ---  B/s |   0  B     --:-- ETA
Not a redundant system. Nothing todo
Dependencies resolved.
================================================================================

 Group                                          Packages

================================================================================
Marking packages as installed by the group:
 @L3                                            bfd

Is this ok [y/N]: y
Complete!
Install operation 10 completed successfully at Fri Mar  8 12:38:08 2024.

[####################] 100%
```

Groupupdate command for updating package groups

Use the **dnf groupupdate** command to update any existing installed group packages.

Update any existing installed group packages using the **dnf groupupdate** command.

```
bash-4.4# dnf groupupdate l3


Last metadata expiration check: 0:00:00 ago on Wed 13 Mar 2024 12:30:11 PM
UTC.                                                 ] ---  B/s |   0  B
--:-- ETA
Dependencies resolved.


================================================================================================================$

 Group                                                          Packages
================================================================================================================$
Marking packages as installed by the group:
 @L3                                                            bfd
================================================================================================================$

 Package                         Arch
Version                                      Repository
      Size
================================================================================================================$
Installing group packages:
 bfd                             lib32_64_n9000
2.0.0.0-10.4.3                               groups-repo
     562 k

Transaction Summary
================================================================================================================$
Install  1 Package

Total size: 562 k
Installed size: 2.3 M
Is this ok [y/N]: y
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :

    1/1
  Running scriptlet: bfd-2.0.0.0-10.4.3.lib32_64_n9000

    1/1
starting pre-install package version mgmt for bfd
pre-install for bfd complete
  Installing       : bfd-2.0.0.0-10.4.3.lib32_64_n9000

    1/1
  Running scriptlet: bfd-2.0.0.0-10.4.3.lib32_64_n9000

    1/1
starting post-install package version mgmt for bfd
post-install for bfd complete
  Verifying        : bfd-2.0.0.0-10.4.3.lib32_64_n9000
```

```
    1/1

Installed:
  bfd.lib32_64_n9000 2.0.0.0-10.4.3

Complete!
Install operation 14 completed successfully at Wed Mar 13 12:30:23 2024.

[####################] 100%
bash-4.4#
```

Grouperase command for deleting groups

Use the **dnf grouperase** command to delete the groups or all the RPM members of the group.

Delete the groups or all the RPM members of the group using the **dnf grouperase** command.

```
bash-4.4# dnf grouperase l3
Dependencies resolved.
================================================================================
 Group                                          Packages

================================================================================
Marking packages as removed by the group:
 @L3                                            bfd

================================================================================
 Package                    Arch                                  Version
                            Repository                         Size
================================================================================
Removing:
 bfd                      lib32_64_n9000                     2.0.0.0-10.4.3
                         @System                          2.3 M

Transaction Summary
================================================================================
Remove  1 Package

Freed space: 2.3 M
Is this ok [y/N]: y
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :
                                                                  1/1
  Running scriptlet: bfd-2.0.0.0-10.4.3.lib32_64_n9000
starting pre-remove package version mgmt for bfd
pre-remove for bfd complete
  Erasing          : bfd-2.0.0.0-10.4.3.lib32_64_n9000
                                                                  1/1
  Running scriptlet: bfd-2.0.0.0-10.4.3.lib32_64_n9000
                                                                  1/1
starting post-remove package version mgmt for bfd
post-remove for bfd complete

    Verifying        : bfd-2.0.0.0-10.4.3.lib32_64_n9000
                                                                    1/1
```

```
Removed:
  bfd.lib32_64_n9000 2.0.0.0-10.4.3
                                                                              Complete!
Install operation 11 completed successfully at Fri Mar  8 12:38:41 2024.

[####################] 100%
bash-4.4#
```

Find repositories

The **dnf repolist all** command lists the repositories that the switch has along with the number of RPMs it has to those repositories.

```
bash-4.3# dnf repolist all

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
 protect-packages
groups-repo
                                                 | 1.1 kB     00:00 ...
localdb
                                                 |  951 B     00:00 ...
patching
                                                 |  951 B     00:00 ...
thirdparty
                                                 |  951 B     00:00 ...
repo id
                        repo name                                        status
groups-repo
                        Groups-RPM Database                              enabled: 37
localdb
                        Local RPM Database                               enabled:  6
patching
                        Patch-RPM Database                               enabled:  0
thirdparty
                        Thirdparty RPM Database                          enabled:  0
open-nxos
                        open-nxos                                        disabled
repolist: 43
```

Installed DNF version

To view the installed version of DNF use the **dnf --version** command.

```
dnf --version

3.4.3
Installed: rpm-5.4.14-r0.0.x86_64 at 2018-06-02 13:04
Built    : Wind River <info@windriver.com> at 2018-04-27 08:36
Committed: Wind River <info@windriver.com> at 2018-04-27
```

```
Installed: yum-3.4.3-r9.0.x86_64 at 2018-06-02 13:05
Built    : Wind River <info@windriver.com> at 2018-04-27 08:36
Committed: Wind River <info@windriver.com> at 2018-04-27
```

Mapping of NX-OS commands to the DNF commands

See the table for mapping the NX-OS commands to the DNF commands:

Table 7: Patching command reference

NX-OS Commands	DNF Commands
show install inactive	**dnf list --patch-only available**
show install active	**dnf list --patch-only installed**
show install committed	**dnf list --patch-only committed**
show install packages	**dnf list --patch-only**
show install pkg-info	**dnf info --patch-only**
show install log	**dnf history --show-patch-log** where log_cmd: ▪ opid= - Log that is specific to an operation ID. ▪ last - Shows the latest operation log. ▪ reverse - Shows the log in reverse order. ▪ detail - Show detailed log. ▪ from= - Shows logging from a specific operation ID.
clear install log	**dnf history --clear-patch-log=** where clear_log_cmd: ▪ all - Clears the complete log. ▪ - Clears the logs above this operation ID.
install add	**dnf install --add bootflash:/**
install remove	**dnf install --remove**
install remove inactive	**dnf install --remove all**

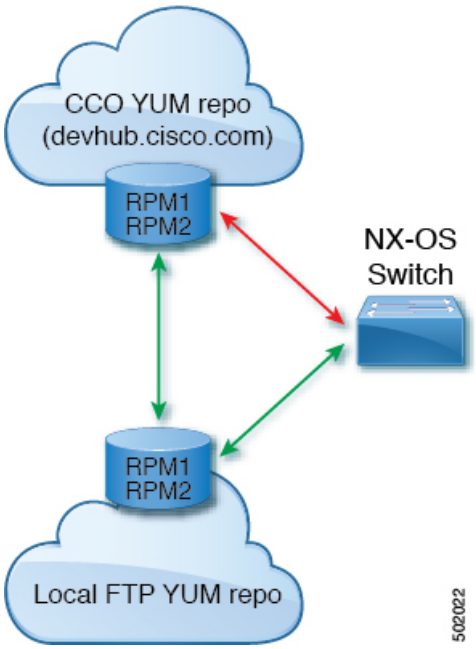
NX-OS Commands	DNF Commands
install activate	**dnf install --no-persist --nocommit** **Note** By default, all packages are activated and committed.
install deactivate	**dnf erase --nocommit** **Note** By default, all packages are deactivated and committed.
install commit	**dnf install --commit**
Install commit	**dnf install --commit all**

Configure an FTP server and set up a local FTP YUM repository

This section outlines the steps to create an FTP server and configure a local FTP YUM repository for managing NX-OS software packages.

For setting up a local FTP YUM repository, you have to first create an FTP server, create a local FTP YUM repository, and configure the NX-OS switch to reach the FTP server as outlined in this illustration.

Figure 3: Configure an FTP server and set up a local FTP YUM repository



Note

For NX-OS Release 10.1(1), visit https://devhub.cisco.com/artifactory/open-nxos/10.1.1/ for **open-nxos** repository.

Create an FTP server on Red Hat Enterprise Linux 7 (RHEL7) Virtual Machine

Complete the following steps to create an FTP server on Red Hat Enterprise Linux 7 (RHEL7) Virtual Machine (VM):

Procedure

1. Install vsftpd, an FTP server, using the **dnfinstall vsftpd** command.
2. Start the FTP server using the **systemctl start vsftpd** command.
3. Check the status of the FTP server using the **systemctl status vsftpd** command.
4. Provide access to the FTP services from the external systems and open port 21 using the **firewall-cmd --zone=public --permanent --add-port=21/tcp** command.
5. Add the FTP service using the **firewall-cmd --zone=public --permanent --add-service=ftp** command.
6. Reload the server using the **firewall-cmd --reload** command.
7. Host a file in the FTP server (for example, test.txt) and attempt Wget of that file using the**wget ftp://** < *ip of FTP server* > / *test.txt* command.

   Note

   The **/var/ftp/** directory is the default home directory of the FTP server.

Create a local FTP YUM repository

Complete the steps provided in this procedure to synchronize the external repository RPMs to the FTP server and create a local FTP YUM repository.

Procedure

1. Create a repository file using the **touch/etc/yum.repos.d/local.repo** command.

   Create a repository file under **/etc/yum.repos.d/** , for example, creates **local.repo** repository and adds the base URL.

   Example

   ```
   bash-4.3# touch /etc/yum.repos.d/local.repo
   ```

2. Edit the repository file and copy the localrepo details.using the **vim /etc/yum.repos.d/local.repo** command.

   > **Note**
   >
   > Modify the base URL to the required repository URL.

   Example

   ```
   bash-4.3# vim /etc/yum.repos.d/local.repo

   [localrepo]
   name=localrepo
   baseurl=
   https://devhub.cisco.com/artifactory/open-nxos/7.0-3-I2-1/x86_64/
   enabled=1
   gpgcheck=0
   sslverify=0
   ```

3. Verify the local repository data to proceed further using the **cat /etc/yum.repos.d/local.repo** command.

   Example

   ```
   bash-4.3# cat /etc/yum.repos.d/local.repo

   [localrepo]
   name=localrepo
   baseurl=
   https://devhub.cisco.com/artifactory/open-nxos/7.0-3-I2-1/x86_64/
   enabled=1
   gpgcheck=0
   sslverify=0
   ```

4. Check the reachability of the repository using the **dnf repolist** command.

   Example

   ```
   bash-4.3# dnf repolist
   Loaded plugins: fastestmirror, langpacks
   ```

```
Loading mirror speeds from cached hostfile
* base: mirror.dhakacom.com
* extras: mirror.dhakacom.com
* updates: mirror.dhakacom.com
repo id repo name status
base/7/x86_64 CentOS-7 - Base 9,911
extras/7/x86_64 CentOS-7 - Extras 313
localrepo localrepo 687
updates/7/x86_64 CentOS-7 - Updates 711
repolist: 11,622
```

5. Synchronize all the packages from the external repository to the FTP server home directory using the **nohup reposync -r** < *repo-name mentioned in the local.repo* > **-p** < *directory path to sync* > **&** command.

   Example

   ```
   nohup reposync -r localrepo -p /var/ftp/ &
   ```

   This command creates a directory with the name **local.repo** inside **/var/ftp/** and downloads all the packages from **devhub.cisco.com** to the directory.

6. Check the status of the synchronization using the **tail -f nouhup.out** command.
   a) **ls /var/ftp/localrepo**
   b) **cd /var/ftp/localrepo/ && createrepo .**

Configure a switch to reach an FTP server

Complete the following steps to configure a switch to reach an FTP server:

Procedure

1. Log in as a sudo user using the **run bash sudo su** command.
2. Verify that the FTP server can be reached using the **ip netns exec management ping** *ip_address* command.

   This command checks the reachability of the FTP server address from the switch using the **ping** command.

3. Create a repository file using the **touch/etc/yum/repos.d/ftp.repo** command.

   This command creates a repository file under **/etc/yum/repos.d/**, for example, it creates the **ftp.repo** repository.

   Example

   ```
   bash-4.3# touch /etc/yum/repos.d/ftp.repo
   ```

4. Edit the repository file and copy the ftp repo details using the **vim /etc/yum/repos.d/ftp.repo** command.

> **Note**
>
> Modify the base URL to the required ftp server IP.

Example

```
bash-4.3# vim /etc/yum/repos.d/ftp.repo

[ftp]
name=ftp
baseurl=
ftp://198.51.100.1/localrepo/
enabled=1
gpgcheck=0
sslverify=0
```

5. Create a repository file on the switch with the FTP server address as the URL using the **cat /etc/yum/repos.d/ftp.repo** command.

Example

```
bash-4.3# cat /etc/yum/repos.d/ftp.repo
[ftp]
name=ftp
baseurl=ftp://198.51.100.1/localrepo/
enabled=1
gpgcheck=0
sslverify=0
```

6. To use the Bash shell prompt, run the **ip netns exec management bash** command.
7. Check the reachability of the newly created repository using the **dnfrepolist** command.

Example

```
bash-4.3# dnf repolist
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
: protect-packages
groups-repo | 1.1 kB 00:00 ...
localdb | 951 B 00:00 ...
patching | 951 B 00:00 ...
thirdparty | 951 B 00:00 ...
thirdparty/primary | 758 B 00:00 ...
thirdparty 1/1
repo id repo name status
groups-repo Groups-RPM Database 37
localdb Local RPM Database 0
patching Patch-RPM Database 0
thirdparty Thirdparty RPM Database 1
ftp ftp 686
repolist: 724
```

8. List the available packages in the new repository using the **dnflist available** command.

Create user roles for install operation

This section specifies that access to the install command is restricted to the admin role unless granted to other users through Role-Based Access Control (RBAC).

The **install** command is only available to the users of admin role. The **install** command can be available to a user by RBAC. For more information about RBAC configuration guidelines, see Guidelines and Limitations for User Accounts and RBAC.

5 Convert from NX-OS to ACI Boot Mode and from ACI back to NX-OS Boot Mode

Topics:

- Convert a N9000 Series switch from NX-OS to ACI boot mode
- Convert a replacement standby supervisor to ACI boot mode
- Convert a N9000 Series switch back to NX-OS

Informs how to transition N9000 Series switches between NX-OS and ACI boot modes by following specific image installation and configuration procedures.

This chapter describes how to convert a N9000 Series switch from NX-OS to Application Centric Infrastructure (ACI) boot mode.

> **Note**
>
> If you need to convert the mode of operation from NX-OS to ACI and the NX-OS switch model has only 16G memory, then the switch requires an RAM memory upgrade to support the ACI mode. For more information about upgrading the RAM of such switches, refer to Adding an 8, 16, or 32 Gigabyte DIMM to a Cisco Nexus 9000 Series Switch.

The sections of this chapter include:

Convert a N9000 Series switch from NX-OS to ACI boot mode

Use this procedure to convert a N9000 Series switch from NX-OS mode to ACI boot mode.

You can convert any N9000 Series switch from NX-OS to Application Centric Infrastructure (ACI) boot mode.

> **Note**
>
> When converting any switch from NX-OS (standalone) to ACI boot mode, you can delete the NX-OS image from the bootflash to save space.

Before you begin

The prerequisites to convert to ACI boot mode include:

- Verify whether your switch hardware is supported in ACI boot mode by checking the *Supported Hardware* section of the Release Notes for Nexus 9000 Series ACI-Mode Switches. For example, line cards are not compatible between NX-OS and ACI boot mode.
- Remove or turn off any unsupported modules (using the **poweroff module** *module* command). Otherwise, the software uses a recovery and retry mechanism before it powers down the unsupported modules. This can cause delays in the conversion process.
- For dual-supervisor systems, use the **show module** command to make sure that the standby supervisor module is in the ha-standby state.
- Verify that the Application Policy Infrastructure Controller (APIC) is running Release 1.0(2j) or a later release.
- Make sure that the ACI image is 11.0(2*x*) or a later release.
- Use the **show install all impact epld** *epld-image-name* command to verify that the switch does not require any EPLD image upgrades. If any upgrades are required, follow the instructions in the Nexus 9000 Series FPGA/EPLD Upgrade Release Notes.

Procedure

1. Verify that the switch is running the latest release.

   Example

   ```
   switch(config)# show version
   ```

   NX-OS file names begin with `nxos`.

2. Copy the ACI image from the APIC:
   a) Set the IP address on the mgmt0 interface of the switch to allow connectivity between this interface and the APIC.
   b) Enable Secure Copy Protocol (SCP) services on the switch.

   Example

   ```
   switch(config)# feature scp-server
   ```

c) From the APIC command line interface, use SCP to copy the firmware image from the APIC to the active supervisor module on the switch.

Example

```
admin@apic1:aci> scp -r /firmware/fwrepos/fwrepo/switch-image-name
admin@switch-ip-address:switch-image-name
```

d) For dual supervisor systems, copy the ACI image to the standby supervisor module.

Example

```
switch(config)# copy bootflash:aci-image bootflash://sup-standby/
```

3. Follow these steps to boot to the ACI image:

a) Configure the switch to not boot from NX-OS.

Example

```
switch(config)# no boot nxos
```

b) Save the configuration.

Example

```
switch(config)# copy running-config startup-config
```

> **Note**
>
> You must run the **copy running-config startup-config** command prior to booting the ACI image. Do not run it after you enter the **boot aci** command.

c) Boot the active and standby supervisor modules with the ACI image.

Example

```
switch(config)# boot aci bootflash:aci-image-name
```

> **Caution**
>
> Do not enter the **copy running-config startup-config** command after the **boot aci** command. If you do, the switch goes to the loader> prompt.

d) Verify the integrity of the file by displaying the MD5 checksum.

Example

```
switch(config)# show file bootflash:aci-image-name md5sum
```

e) Reload the switch.

Example

```
switch(config)# reload
```

f) Log in to the switch as an administrator.

Example

```
Login: admin
```

4. Verify whether you must install certificates for your device.

Example

```
admin@apic1:aci> openssl asn1parse -in /securedata/ssl/server.crt
```

Look for PRINTABLESTRING in the command output. If `Manufacturing CA` is listed, the correct certificates are installed. If something else is listed, contact TAC to generate and install the correct certificates for your device.

> **Note**
>
> You might need to install certificates for Nexus 9000 Series switches that were shipped prior to May 2014.
>
> To run this command, contact TAC.

What to do next

Refer to the ACI and APIC documentation to configure and operate the switch in ACI mode: https://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html.

Convert a replacement standby supervisor to ACI boot mode

Use this procedure to replace the standby supervisor module in a dual-supervisor system by copying and booting the Application Centric Infrastructure (ACI) image from a USB drive, and properly configuring the new module to ensure seamless switch operation.

If you ever need to replace the standby supervisor module in a dual-supervisor system, you will need to copy and boot the Application Centric Infrastructure (ACI) image for use with the replacement standby supervisor.

Before you begin

Copy the ACI image to a USB drive.

Procedure

1. Reload the switch.

   Example

   ```
   switch# reload
   ```

2. Enter a break sequence (Ctrl-C or Ctrl-]) during the initial boot sequence to access the loader > prompt.

   Example

   ```
   Ctrl-C
   loader>
   ```

3. Plug the USB drive containing the ACI image into the standby supervisor USB slot.
4. Boot the ACI image.

   Example

   ```
   loader> boot usb#:aci-image-name
   ```

   Note

   If you have two USB drives, enter the **dir** command to see which drive contains the ACI image. Then specify either **usb1** or **usb2** in the **boot** command.

5. Log in to the switch as an administrator.

   ```
   Login: admin
   ```

6. Copy the ACI image from the USB drive to the switch.

   Example

   ```
   switch# copy usb#:aci-image-name bootflash:aci-image-name
   ```

Convert a N9000 Series switch back to NX-OS

Learn how to convert a N9000 Series switch from ACI boot mode back to NX-OS standalone mode by reformatting the file system and booting the standalone software image.

You can convert a N9000 Series switch from Application Centric Infrastructure (ACI) boot mode back to NX-OS.

Note

When converting any switch from ACI to NX-OS (standalone) boot mode, you can delete the ACI image from the bootflash to save space.

Procedure

1. Reload the switch.

   Example

   ```
   switch# reload
   ```

2. During the initial boot sequence, enter a break sequence (Ctrl-C or Ctrl-]) to access the loader> prompt.

   Example

   ```
   Ctrl-C
   loader>
   ```

3. Configure the boot process to stop at the boot prompt.

   Example

   ```
   loader> cmdline recoverymode=1
   ```

4. Boot the active supervisor module with the NX-OS image.

   Example

   ```
   loader> boot nxos.9.2.3.bin
   ```

> **Note**
>
> If the NX-OS image that is mentioned in the boot variable is not present in the bootflash, the system falls back to the loader prompt during the boot sequence. To recover the switch from the loader prompt, boot the system through a different image present in the bootflash, perform a **tftpboot** , or boot through a USB device.

> **Note**
>
> For some NX-OS releases and N9000 Series switches, a sample error message that appears is

```
!!Fatal error!!
Can't reserve space for RPM repo
Please free up bootflash space and reboot
```

> If you see this error message, start over from Step 1. After Step 3, enter the **cmdline init_system** command and then go to Step 4. The switch boots into the normal NX-OS prompt and skips the switch(boot)# prompt.

5. Restore the switch file system partitioning to the default settings. The bootflash filesystem is reset to NX-OS partitioning, and the NX-OS image is deleted.

   Example

   ```
   switch(boot)# init system
   ```

6. Complete the upload of the NX-OS image file.

   Example

   ```
   switch(boot)# load-nxos
   ```

Note

For some N9000 Series switches, the device does not load with the normal NX-OS prompt (switch#) and instead comes up as `bash-4.2#`. In this case, you must power cycle the device, jump to loader, and boot the NX-OS image using either TFTP or a USB method.

- For the TFTP method - First assign an IP address and gateway to the device using the **set ip***ip addresssubnet mask* and the **set gw***gateway address* commands. This is required as the **init system** command in the above step erases all available configurations on the device

  Example

  ```
  loader> set ip 192.0.2.10 255.255.255.0
  loader> set gw 192.0.2.1
  ```

  Then use the **tftp** command to load the image.

  ```
  loader> boot tftp://<tftp server ip>/<nxos-image-name>
  ```

- For the USB method - Mount the USB on the switch and execute the **dir** command on the loader to see the contents of the bootflash folder and the USB device.

  Example

  ```
  loader > dir
  usb1::
  lost+found
  /nxos.9.x.y.bin
  ```

  Then boot the NX-OS image using this command:

  ```
  loader> boot usb1:/nxos-image
  Example: boot usb1:/nxos.9.x.y.bin
  ```

After you boot the NX-OS image, the device loads as a NX-OS switch and you can continue with the remaining steps.

7. Copy the NX-OS image into the bootflash: and set the appropriate boot variables to ensure that the system boots the NX-OS image on the next reload.

   Example

   TFTP example:

   ```
   switch# copy tftp://tftp-server-ip/nxos-image-namebootflash:

   switch# configure terminal

   switch(config)# boot nxos bootflash:nxos-image-name

   switch(config)# copy running-config startup-config
   switch(config)# end
   ```

USB example:

```
switch# copy usb1:nxos-image-namebootflash:

switch# configure terminal
switch(config)# boot nxos bootflash:nxos-image-name
switch(config)# copy running-config startup-config
switch(config)# end
```

8. Wait for the system controllers to come up (approximately 15-20 minutes).

   File system differences between ACI and NX-OS require one-time reformatting change during the ACI to NX-OS conversion. Subsequent reloads with the NX-OS image are faster.

9. Verify that the active supervisor module and the system controllers are in the active state.

   Example

```
switch# show module
```

> **Note**
>
> - For dual-supervisor systems, follow Steps 3-6 on the standby supervisor.
> - Log in to the switch and verify that it is running NX-OS software.

Load NX-OS image into bootflash using SCP on the ACI shell

Describes how to use SCP on the ACI Shell to load an NX-OS image when TFTP and USB access are unavailable.

If you have a N9000 Series Switch in ACI mode and must convert it to NX-OS mode but are unable to perform a TFTP boot and physical access to USB is not available, then Secure Copy Protocol (SCP) can be used on the ACI Shell to load a NX-OS Image into the bootflash. Contact TAC for further information about the process, as this requires root access.

6 Migrate Switches in a vPC Topology

Topics:

- Upgrade process for a vPC topology on the primary switch
- Upgrade process for a vPC topology on the secondary switch
- vPC forklift upgrade
- vPC upgrade and downgrade procedure for N9000 -R series switches

Provides instructions about migration from one pair of N9000 Series switches to another within a vPC topology, covering both forklift upgrades and specific procedures for the -R series models.

Upgrade process for a vPC topology on the primary switch

This section summarizes the upgrade process on a switch in a vPC topology that holds either the Primary or Operational Primary vPC roles.

> **Note**
>
> In vPC topologies, the two peer switches must be upgraded individually. An upgrade on one peer switch does not automatically update the vPC peer switch.
>
> In a dual-homed non-vPC access network (either a triangle or Y shaped access network), with or without STP configured, with BFD enabled and with HSRP configured on the SVIs and with HSRP configured as BFD client, transient traffic may drop for both IPv4 native unicast and/or labeled traffic after performing an ND-ISSU n fallback mode.
>
> To counter this, configure in all the HSRP IPv4 groups for all the HSRP enabled SVIs the `timer 2 120` on both HSRP peers prior to performing the ND-ISSU. The configuration of the `timer 3 120` may lead to traffic loss.

Summary

Upgrading a vPC primary switch requires a coordinated process to update its software while minimizing network disruption, ensuring only the primary switch is upgraded at a time for operational stability. The upgrade involves compatibility checks, configuration locks, and a stateful restart. Thus, it involves coordinating several steps to update its software while minimizing disruption. This process ensures that only the primary switch is upgraded at a time, maintaining operational stability.

Workflow

Perform this upgrade procedure on a switch in a vPC topology that holds either the Primary or Operational Primary vPC roles.

Step 1 and Step 4 differ from a switch upgrade in a non-vPC topology.

1. The **install all** command issued on the vPC primary switch triggers the installation upgrade.
2. The compatibility checks display the impact of the upgrade.
3. The installation proceeds or not based on the upgrade impact.
4. The configuration is locked on both vPC peer switches.
5. The current state is saved.
6. The system unloads and runs the new image.
7. The stateful restart of the system software and application occurs.
8. The installer resumes with the new image.
9. The installation is complete.

Result

When the installation is complete, the vPC primary switch is upgraded.

> **Note**
>
> The vPC primary switch is running the upgraded version, and the vPC secondary switch is running the original software version.

Upgrade process for a vPC topology on the secondary switch

Learn how to upgrade a switch in vPC topology that holds either the Secondary or Operational Secondary vPC roles.

Summary

This procedure describes the upgrade process initiated by the install all command on the vPC secondary switch, including compatibility checks, image installation, stateful restart, and configuration unlock on both primary and secondary switches to complete the upgrade.

Step 1 and Step 8 differ from a switch upgrade in a non-vPC topology.

Workflow

1. The **install all** command issued on the vPC secondary switch triggers the installation upgrade.
2. The compatibility checks display the impact of the upgrade.
3. The installation proceeds or not based on the upgrade impact.
4. The current state is saved.
5. The system unloads and runs the new image.
6. The stateful restart of the system software and application occurs.
7. The installer resumes with the new image.
8. The configuration is unlocked on the primary and secondary switches.
9. The installation is complete.

vPC forklift upgrade

This section describes the scenario and considerations for performing a vPC forklift upgrade on Cisco Nexus 9500 switches.

In a vPC topology, you can migrate from a pair of Nexus 9000 Series switches to a different pair of Nexus 9000 Series switches. For example, you can migrate from a pair of Nexus 9508 vPC peer nodes to a pair of Nexus 9516 switches. For more information, see the *vPC Forklift Upgrade Scenario* section in the Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide.

vPC upgrade and downgrade procedure for N9000 -R series switches

Upgrade or downgrade vPC peer switches in N9000 -R series, ensuring compatibility and minimizing traffic impact.

In vPC topologies, the two peer switches usually must be upgraded individually. An upgrade on one peer switch does not automatically update the vPC peer switch.

However, NX-OS Releases 7.0(3)F3(3c) and 7.0(3)F3(4) are not compatible with NX-OS Release 9.2(x) for vPC peer switches. Both vPC peers must be upgraded simultaneously to NX-OS Release 9.2(x) to avoid one switch running a 7.0(3)F3(x) release and the other switch running 9.2(x). Optionally, if the switches are being upgraded from NX-OS Release 7.0(3)F3(4), you can use the following procedure to minimize the traffic impact during upgrade.

Note

This procedure not to be used on Broadcom or Cloudscale-based switches.

Procedure

1. Switch A and B are running a NX-OS release. Switch A is the primary switch, and switch B is the secondary switch. On both the switches, use the **copy r s** command to save the running configuration.

   Example

```
primary_switch# show vpc role
vPC Role status
----------------------------------------------------
vPC role : primary
vPC system-mac : 00:23:04:ee:be:64
vPC system-priority : 32667
vPC local system-mac : 70:df:2f:eb:86:1f
vPC local role-priority : 90
vPC peer system-mac : 70:df:2f:eb:1c:ab
vPC peer role-priority : 100
primary_switch#

secondary_switch# show vpc role
vPC Role status
----------------------------------------------------
vPC role : secondary
vPC system-mac : 00:23:04:ee:be:64
vPC system-priority : 32667
vPC local system-mac : 70:df:2f:eb:1c:ab
vPC local role-priority : 100
vPC peer system-mac : 70:df:2f:eb:86:1f
vPC peer role-priority : 90
secondary_switch#

primary_switch# copy r s v
[########################################] 100%
Copy complete.

secondary_switch# copy r s v
[########################################] 100%
Copy complete.
```

2. Bring down the peer link (PL) on the primary switch. The secondary switch brings down its vPC legs.

   Example

```
primary_switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

```
primary_switch(config)# int port-channel 100
primary_switch(config-if)# shutdown

Reload the secondary switch with Release 9.2.1 image (change bootvar /reload)

secondary_switch(config)# boot nxos nxos.9.2.1.bin
Performing image verification and compatibility check, please wait....
secondary_switch(config)#
secondary_switch(config)# copy r s v
[########################################] 100%
Copy complete.


secondary_switch# reload
This command will reboot the system. (y/n)? [n] y


After reload
--------------------
secondary_switch# show vpc
Legend:
(*) - local vPC is down, forwarding via vPC peer-link
vPC domain id : 100
Peer status : peer link is down
vPC keep-alive status : peer is alive
Configuration consistency status : failed
Per-vlan consistency status : success
Configuration inconsistency reason: Consistency Check Not Performed
Type-2 inconsistency reason : Consistency Check Not Performed
vPC role : none established
Number of vPCs configured : 20
Peer Gateway : Enabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Disabled (due to peer configuration)
Auto-recovery status : Disabled
Delay-restore status : Timer is off.(timeout = 90s)
Delay-restore SVI status : Timer is off.(timeout = 10s)
Operational Layer3 Peer-router : Disabled
vPC Peer-link status
---------------------------------------------------------------------
id Port Status Active vlans
-- ---- ------ --------------------------------------------------
1 Po100 down -

secondary_switch#

primary_switch(config-if)# show vpc
Legend:
(*) - local vPC is down, forwarding via vPC peer-link
vPC domain id : 100
Peer status : peer link is down
vPC keep-alive status : peer is alive
Configuration consistency status : success
Per-vlan consistency status : success
Type-2 consistency status : success
vPC role : primary
Number of vPCs configured : 20
Peer Gateway : Enabled
Peer gateway excluded VLANs : -
Dual-active excluded VLANs and BDs : -
Graceful Consistency Check : Enabled
Auto-recovery status : Enabled, timer is off.(timeout = 240s)
```

```
Operational Layer3 Peer-router : Disabled
vPC Peer-link status
---------------------------------------------------------------------
id Port Status Active vlans
-- ---- ------ --------------------------------------------------
1 Po100 down -
```

3. Configure vPC auto-recovery under the vPC domain on the secondary switch. Enable **vpc upgrade** (exec command).

Example

```
secondary_switch(config)# vpc domain 100
secondary_switch(config-vpc-domain)# auto-recovery
secondary_switch(config-vpc-domain)# end

secondary_switch# show running-config vpc
!Command: show running-config vpc
!Running configuration last done at: Wed May 16 06:34:10 2018
!Time: Wed May 16 06:34:14 2018
version 9.2(1) Bios:version 01.11
feature vpc
vpc domain 100
peer-switch
role priority 100
peer-keepalive destination 10.1.31.30 source 10.1.31.29
delay restore 90
peer-gateway
auto-recovery
ipv6 nd synchronize
ip arp synchronize
interface port-channel100
vpc peer-link
interface port-channel2001
vpc 101


secondary_switch# show vpc upgrade
vPC upgrade : TRUE
SVI Timer : 0
Delay Restore Timer : 0
Delay Orphan Port Timer : 0
secondary_switch#

secondary_switch# show vpc upgrade   >> Hidden command
vPC upgrade : FALSE
SVI Timer : 10
Delay Restore Timer : 90
Delay Orphan Port Timer : 0


secondary_switch# vpc upgrade   >> Hidden command
```

4. After Layer 3 routes are learned on the secondary switch, reload the primary switch with the new release image. The secondary switch takes over the primary role and brings up its vPC legs in approximately 5 seconds.

Example

```
primary_switch(config)# show boot
Current Boot Variables:
```

```
sup-1
NXOS variable = bootflash:/nxos.9.2.1.bin
No module boot variable set
Boot Variables on next reload:
sup-1
NXOS variable = bootflash:/nxos.9.2.1.bin

No module boot variable set
primary_switch(config)# end

primary_switch# show boot
Current Boot Variables:
sup-1
NXOS variable = bootflash:/nxos.9.2.1.bin
No module boot variable set
Boot Variables on next reload:
sup-1
NXOS variable = bootflash:/nxos.9.2.1.bin

No module boot variable set
primary_switch# reload
This command will reboot the system. (y/n)? [n] y

secondary_switch# show vpc
Legend:
(*) - local vPC is down, forwarding via vPC peer-link
vPC domain id : 100
Peer status : peer link is down
vPC keep-alive status : peer is not reachable through peer-keepalive
Configuration consistency status : failed
Per-vlan consistency status : success
Configuration inconsistency reason: Consistency Check Not Performed
Type-2 inconsistency reason : Consistency Check Not Performed
vPC role : primary
Number of vPCs configured : 20
Peer Gateway : Enabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Disabled (due to peer configuration)
Auto-recovery status : Enabled, timer is off.(timeout = 240s)
Delay-restore status : Timer is off.(timeout = 0s)
Delay-restore SVI status : Timer is off.(timeout = 0s)
Operational Layer3 Peer-router : Disabled
vPC Peer-link status
---------------------------------------------------------------------
id Port Status Active vlans
-- ---- ------ --------------------------------------------------
1 Po100 down -
vPC status
```

5. When the primary switch comes back up, the peer link on it is operationally up.

   Example

```
primary_switch# show vpc
Legend:
(*) - local vPC is down, forwarding via vPC peer-link
vPC domain id : 100
Peer status : peer adjacency formed ok
vPC keep-alive status : peer is alive
Configuration consistency status : success
Per-vlan consistency status : success
```

```
Type-2 consistency status : success
vPC role : primary, operational secondary
Number of vPCs configured : 20
Peer Gateway : Enabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Enabled
Auto-recovery status : Disabled
Delay-restore status : Timer is off.(timeout = 90s)
Delay-restore SVI status : Timer is off.(timeout = 10s)
Operational Layer3 Peer-router : Disabled
vPC Peer-link status
---------------------------------------------------------------------
id Port Status Active vlans
-- ---- ------ --------------------------------------------------
1 Po100 up 1,101-400
```

For downgrade, reload both switches at the same time.