



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

First Published: 2025-08-13

Last Modified: 2026-04-16

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

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)

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



CONTENTS

CHAPTER 1	What's Changed in This Book	1
	What's changed in this guide	1

CHAPTER 2	Upgrade or Downgrade the Nexus 9000 Series NX-OS Software	3
	Software image	3
	Prerequisites for NX-OS software upgrade	5
	Nexus health and configuration checks	6
	NX-OS software upgrade guidelines	6
	MACsec non-disruptive ISSU	18
	ISSU platform support	19
	Upgrade paths	28
	Install upgrade patches	29
	Upgrade the NX-OS software	38
	In-Service Software Upgrade	41
	NX-OS upgrade history	43
	Prerequisites for NX-OS software downgrade	44
	NX-OS software downgrade guidelines	44
	Downgrade to an earlier software release	47

CHAPTER 3	Transceiver Firmware	51
	Transceiver firmware upgrades	51
	Firmware upgrade using NX-OS commands	52
	Parallel firmware download queues and port mapping	52
	Firmware upgrade phases	53
	Firmware upgrade procedures	53
	Guidelines and limitations for firmware upgrade using NX-OS commands	55

Upgrade firmware using NX-OS commands	55
Disruptive immediate upgrade of the transceiver	56
Non-disruptive staged upgrade of the transceiver	58
Non-disruptive immediate upgrade of the transceiver	60
Examples for firmware upgrade	62
Sample disruptive immediate upgrade with both banks valid	62
Sample disruptive immediate upgrade with one bank invalid	64
Sample non-disruptive staged upgrade	67
Sample non-disruptive immediate upgrade	70
Firmware upgrade for a range of interfaces or parallel firmware download	72
Parallel firmware download	72
Firmware download guidelines and limitations	72
Copy Transceiver Firmware to Switch and Line Card	73
Download, Run, and Commit Transceiver Firmware	74

CHAPTER 4
Optionality in NX-OS Software 77

Optionality in NX-OS software	77
Modular packages	78
NX-OS image boot modes	79
Red Hat Package Managers	80
Format of the RPM	80
Optional RPMs and their associated features	81
Guidelines for NX-OS feature RPM installation	83
Guidelines for third-party RPM installation	84
Install command options for feature and third-party RPMs	85
Use install commands for digital signature support	87
Query all installed RPMs	88
Install RPMs using the one-step procedure	89
Install RPMs using the two-step procedure	90
Upgrade the RPMs	91
Downgrade RPMs	91
Uninstall the RPMs	92
Remove the RPMs	93
Dandified YUM commands	93

Package operations with DNF commands	94	
Find the base version RPM of the image	94	
Check the list of the installed RPMs	95	
Get details of the installed RPMs	95	
Install RPMs	96	
Upgrade RPMs	98	
Downgrade RPMs	100	
Delete RPMs	101	
Support for DNF groups	102	
Find repositories	105	
Installed DNF version	106	
Mapping of NX-OS commands to the DNF commands	106	
Configure an FTP server and set up a local FTP YUM repository	108	
Create an FTP server on Red Hat Enterprise Linux 7 (RHEL7) Virtual Machine	108	
Create a local FTP YUM repository	109	
Configure a switch to reach an FTP server	110	
Create user roles for install operation	111	
<hr/>		
CHAPTER 5	Convert from NX-OS to ACI boot mode and from ACI boot mode back to NX-OS	113
	Convert a Nexus 9000 Series switch from NX-OS to ACI boot mode	113
	Convert a replacement standby supervisor to ACI boot mode	115
	Convert a Nexus 9000 Series switch back to NX-OS	116
	Load NX-OS image into bootflash using SCP on the ACI shell	119
<hr/>		
CHAPTER 6	Migrate Switches in a vPC Topology	121
	Upgrade process for a vPC topology on the primary switch	121
	Upgrade process for a vPC topology on the secondary switch	122
	vPC forklift upgrade	123
	vPC upgrade and downgrade procedure for Nexus 9000 -R series switches	123



CHAPTER 1

What's Changed in This Book

- [What's changed in this guide, on page 1](#)

What's changed in this guide

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
N9164E-NS4-O software image	Added a new software image for N9164E-NS4-O switch	10.6(2n)	Software image, on page 3 NX-OS software upgrade guidelines, on page 6
MACsec ND ISSU	Added support for ND ISSU on MACsec-enabled interfaces on Nexus 9300-GX2, H2R, and H1 switches	10.6(2)F	NX-OS software upgrade guidelines, on page 6 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, on page 51
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 3
NA	No new features added for this release	10.6(1)F	NA



CHAPTER 2

Upgrade or Downgrade the Nexus 9000 Series NX-OS Software

- [Software image, on page 3](#)
- [Prerequisites for NX-OS software upgrade, on page 5](#)
- [Nexus health and configuration checks, on page 6](#)
- [NX-OS software upgrade guidelines, on page 6](#)
- [ISSU platform support, on page 19](#)
- [Upgrade paths, on page 28](#)
- [Install upgrade patches, on page 29](#)
- [Upgrade the NX-OS software, on page 38](#)
- [In-Service Software Upgrade, on page 41](#)
- [NX-OS upgrade history, on page 43](#)
- [Prerequisites for NX-OS software downgrade, on page 44](#)
- [NX-OS software downgrade guidelines, on page 44](#)
- [Downgrade to an earlier software release, on page 47](#)

Software image

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 with different prefixes for supported Nexus platforms.

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 include:

- The 64-bit NX-OS image file name with `nxos64-cs` as the prefix (for example, `nxos64-cs.10.5.3.F.bin`) is supported on all Nexus 9000 series switches except Nexus 9500 -R and -R2 switches and line cards, Nexus 9800 switches, and N9364E-SG2 switches.
- The 64-bit NX-OS image file name with `nxos64-ms11` as the prefix (for example, `nxos64-ms11.10.5.3.F.bin`) is supported only on Nexus 9500 Series -R and -R2 modular switches.
- The 64-bit NX-OS image file name with `nxos64-s1` as the prefix (for example, `nxos64-s1.10.5.3.bin`) is 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.
- The 64-bit NX-OS image file name with `nxos64-s1-dpu` as the prefix (for example, `nxos64-s1-dpu.10.6.2.F.bin`) is mandatory on N9324C-SE1U and N9348Y2C6D-SE1U Smart switches. This image is supported from NX-OS Release 10.6(2)F. Currently, these switches work in networking mode only.

**Note**

- Only disruptive upgrade is supported on the Smart switches.
- Service-acceleration feature is not supported in 10.6(2)F.

- The 64-bit NX-OS image file name with `nxos64-msx` as the prefix (for example, `nxos64-msx.10.6.2n.F.bin`) is 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 [Cisco Nexus 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](#).
- 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 Nexus 9000 Series NX-OS System Management Configuration Guide](#).

EPLD image

The Nexus 9000 Series NX-OS mode switches contain several programmable logical devices (PLDs) that provide hardware functions in all modules. PLDs include electronic programmable logic devices (EPLDs), field programmable gate arrays (FPGAs), and complex programmable logic devices (CPLDs), but not ASICs.

In this document, the term EPLD is used for FPGA and CPLDs. EPLD upgrades improve hardware functions and resolve known issues.

- For more information about EPLD, refer to [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.

Prerequisites for 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.



Note For more information about configuration sessions, see the [Cisco Nexus 9000 Series NX-OS System Management Configuration Guide](#) specific to your release.

Nexus health and configuration checks

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

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](#)
- [Enhanced mode](#)
- [Feature specific](#)

- [FEX](#)
- [FC/FCoE NPV](#)
- [VXLAN with TRM](#)
- [Unsupported PIDs](#)

Generic

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

- 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 console ports to maintain connectivity when switchovers occur during a software upgrade. For more information about your specific chassis, see the relevant [Hardware Installation Guide](#).
 - Perform the installation on the active supervisor module, not the standby supervisor module.
- 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.
- Beginning with NX-OS Release 10.4(2)F, for Nexus 9300-R platforms, to upgrade bios to the latest version you should first upgrade to `nxos` image. This release onwards, the `install all nxos` command only upgrades the `nxos` sw to the latest version but the bios image will be upgraded to the last bios released prior to 10.4(2)F version.

To upgrade to bios released with 10.4(2)F or higher version, first upgrade the `nxos` image and then use `bios-force` option to upgrade the bios. For example,

1. Install all `nxos` bootflash:`nxos64-msll.10.4.2.F.bin`.

The system reloads and boots up with 10.4(2)F image.

2. Install all nxos bios-force.



Note The switch reloads twice, once for nxos upgrade and then again for bios upgrade.

- For platforms that need to be upgraded from any release to nxos64-cs.10.3(1)F or higher release, use nxos.9.3.10.bin or nxos64-cs.10.2(3)F or higher release as an interim hop. This restriction is applicable to both disruptive and non-disruptive upgrades. The nxos64-msll.10.3(1)F does not have this restriction.
- Loading an unsupported image on Nexus 9800 platform switches cause the switch to be stuck. Only a power cycle can reset it.
- Beginning with NX-OS Release 10.2(2)F, Nexus 9504 and 9508 platform switches, and Nexus 9508-R, R2, and RX line cards support NX-OS 64-bit images. Disruptive upgrade from earlier releases to 10.2(2)F 64-bit NX-OS image is supported. NX-OS 32-bit image is not supported on these platform switches anymore.
- Beginning with NX-OS Release 10.1(x), when an existing SMU is active, if you install a bundle that contains the existing active SMU, the installer installs only the non-existing SMUs.
- 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.



Note For Nexus 9500 platform switches with -R line cards, you must save the configuration and reload the device to upgrade from NX-OS Release 7.0(3)F3(5) to 9.3(1). To upgrade from NX-OS Release 9.2(2) or 9.2(3), we recommend that you use the **install all** command.

- 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, refer to *Table 1* in the [FPGA/EPLD Upgrade Procedure to Address Secure Boot Vulnerability](#) document.
- Beginning with NX-OS Release 10.5(3)F, all NX-OS images are bundled with EPLD image and EPLD upgrade is triggered automatically as part of **install all nxos** command. However, you have the option to skip EPLD image upgrade.
- 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.
- ISSU is not supported if EPLD is not at NX-OS Release 7.0(3)I3(1) or later.

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](#).
- Beginning with NX-OS Release 10.4(2)F, SR ISSU is not supported with underlay ISIS.
- When upgrading from earlier release to NX-OS Release 10.3(3)F and later, if the **hardware rate-limiter span-egress** command is configured then it must be removed and reapplied after the upgrade/ISSU is complete.
- Beginning with NX-OS Release 10.3(2)F, 2xSFP Eth10/1-2 are not supported on N9K-C9400-SW-GX2A. However, from NX-OS Release 10.4(2)F onwards, N9K-C9400-SW-GX2A Sup card ports 2xSFP Eth10/1-2 are supported.
- ASIC SFP+ ports Eth10/1-2 are not supported in NX-OS Releases 10.3(2)F, 10.3(3)F, and 10.4(1)F. Beginning with NX-OS Release 10.4(2)F, these ports are supported. However, note that after reloading the system, these ASIC SFP+ Eth10/1-2 ports can take up to 3 minutes to link up.
- While performing an ISSU on the L2 switch in a vPC complex or a LAN scenario, the IGMP group timeout must be configured with higher value as the L2 switch will not be able to forward the reports/queries during the control plane down time. The L2 snooping querier interval must also be matched to the L3 querier interval.
- While performing an ISSU from NX-OS Release 9.3(5), 9.3(6), 9.3(7), 10.1(1), or 10.1(2) to NX-OS Release 10.2(1) or higher release, ISSU is blocked.
- 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.
- When upgrading directly to NX-OS Release 10.1(x) from any release prior to 7.0(x), the upgrade is disruptive. For a non-disruptive upgrade, an intermediate upgrade to NX-OS Release 9.x is required. We recommend upgrading to the latest release of NX-OS Release 9.3(x) as an intermediate hop for the

upgrade. For information about the supported upgrade paths, see the [Cisco Nexus 9000 and 3000 Upgrade and ISSU Matrix](#).

- When upgrading from NX-OS Release 7.0(3)I6(1) or 7.0(3)I7(1) to NX-OS Release 10.1(x), if the Nexus 9000 Series switches are running vPC and they are connected to an IOS-based switch via Layer 2 vPC, there is a likelihood that the Layer 2 port channel on the IOS side will become error disabled. The workaround is to disable the spanning-tree etherchannel guard misconfig command on the IOS switch before starting the upgrade process.

Once both the Nexus 9000 Series switches are upgraded, you can re-enable the command.

- If you are upgrading from NX-OS Release 7.0(3)I5(2) to NX-OS Release 10.1(x) by using the **install all** command, BIOS will not be upgraded due to CSCve24965. When the upgrade to NX-OS Release 10.1(x) is complete, use the **install all** command again to complete the BIOS upgrade, if applicable.
- When upgrading from NX-OS Release 9.2(4) or earlier releases to NX-OS Release 9.3(4) or later, running configuration contains extra TCAM configuration lines. You can ignore these extra lines as they do not have an effect on the upgrade and configuration.
- When performing an ISSU from NX-OS Release 9.3(1) or 9.3(2) to NX-OS Release 9.3(3) or later, ensure that the features with user-defined ports, such as **<ssh port>**, are within the prescribed port range. If the port range is incorrect, follow the syslog message recommendation. For more information about the port range, see [Cisco Nexus 9000 Series NX-OS IP SLAs Configuration Guide](#).
- For any prior release version upgrading to NX-OS Release 9.3(5) using ISSU, if the following logging level commands are configured, they are missing in the upgraded version and must be reconfigured:
 - **logging level evmc** *value*
 - **logging level mvsh** *value*
 - **logging level fs-daemon** *value*
- For any prior release version upgrading to NX-OS Release 9.3(6) using ISSU, if the following logging level commands are configured, they are missing in the upgraded version and must be reconfigured:
 - **logging level evmc** *value*
 - **logging level mvsh** *value*

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.
- Beginning with NX-OS Release 10.4(3)F, non-disruptive ISSU is not supported on Nexus 92348GC-X.

- Beginning with NX-OS Release 10.4(1)F, only the LXC mode is supported on Nexus 9300-FX and 9300-FX2 switches, in addition to Nexus 9300-FX3 and 9300-GX switches. This allows you to perform enhanced non-disruptive ISSU with minimal downtime. However, on the rest of the Nexus 9000 switches, you have an option to perform a non-disruptive ISSU in the enhanced LXC mode with minimal downtime.
- Beginning with NX-OS Release 10.4(1)F, only the enhanced LXC mode is supported on Nexus N9K-C9332D-H2R, N9K-C9348GC-FX3, and N9K-C9348GC-FX3PH switches by default.
- Non-disruptive ISSU is not supported on interfaces with 2.5G or 5G speed on N9K-C93108TC-FX3P platform. For more information, see [CSCwq38959](#).
- After disruptive upgrade from NX-OS Release 10.3(x) to 10.5(x) and later releases, Nexus 9300-FX switches lose FC ports configuration and the FC ports turn into Ethernet ports. However, if `port x - y mode fc` exists under `slot z` in the running configuration, though such ports are changed to Ethernet ports, after a switch reload, they change back to FC ports.
- Beginning with NX-OS Release 10.2(2)F, FCoE/FC NPV is supported on N9K-C9336C-FX2-E platform switches.

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.

- Before enabling the FHS on the interface, we recommend that you carve the ifacl TCAM region on Nexus 9300 and 9500 platform switches. If you carved the ifacl TCAM region in a previous release, you must reload the system after upgrading to NX-OS Release 10.1(x). Uploading the system creates the required match qualifiers for the FHS TCAM region, ifacl.
- Before enabling the FHS, we recommend that you carve the ing-redirect TCAM region on Nexus 9200 and 9300-EX platform switches. If you carved the ing-redirect TCAM region in a previous release, you must reload the system after upgrading to NX-OS Release 10.1(x). Uploading the system creates the required match qualifiers for the FHS TCAM region, ing-redirect.
- When upgrading from Nexus 94xx, 95xx, and 96xx line cards to Nexus 9732C-EX line cards and their fabric modules, upgrade the NX-OS software before inserting the line cards and fabric modules. Failure to do so can cause a diagnostic failure on the line card and no TCAM space to be allocated. You must use the **write_erase** command followed by the **reload** command.

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.

- While performing multi-hop ND ISSU upgrade to higher releases, use 10.3(5)M or higher release as an intermediate hop.
 - If the switch has been previously upgraded using multi-hop ND ISSU and experiences unexpected CoPP drops, open a TAC case to determine if remediation is required.
- Beginning with NX-OS Release 10.2(3)F, non-disruptive ISSU is supported for VPC fabric peering on all Nexus 9300-X TORs. Both standard and enhanced non-disruptive upgrades are supported. Note that ISSU should be started or triggered when there is no failure. An example for failure would be one of the VPC legs is down.
- The recommended routing protocol graceful restart timer is 600 seconds and nve source-interface hold-down-time is 400 seconds.
- 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.
- Disruptive upgrade from any version before 9.3(10) or 10.2(3)F may fail due to [CSCwb63451](#). You must upgrade to 9.3(10) or 10.2(3)F first, before upgrading to 10.3(1)F or later.
- Beginning from NX-OS Release 10.2(8)M onwards, Nexus 9300-FX3 supports non-disruptive upgrade.
- Beginning with NX-OS Release 10.2(3)F, for switches that are in LXC mode and for non-disruptive upgrade, a new option **skip-kernel-upgrade** is added to **install** command.
- MPLS strip, GRE strip, and any underlying ACL configuration is not ISSU compatible when you perform ND ISSU to NX-OS Release 10.2(2)F from a previous release.

After ND ISSU to NX-OS Release 10.2(2)F or 10.2(3)F from a previous release, post GRE strip dot1q tunnel VLAN_tag might be missing. Workaround for this issue is to remove and add port ACL from L2 interfaces for GRE strip enabled interface.
- For a device that is running on NX-OS Release 10.1(2), 10.2(1)F, and 10.2(2)F, ND-ISSU is not supported if Layer 2 sub-interfaces are configured.
- When performing ND ISSU using BGP non-default hold timers, ensure that the BGP graceful-restart timer is reasonably long enough, for example, 180 seconds.
- 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.
- Beginning with NX-OS Release 9.3(5), standard, nondisruptive ISSU, on switches that are configured with uRPF, is supported on:
 - Nexus 9300-EX platform switches
 - Nexus 9300-FX/FX2 platform switches, and
 - Nexus 9300-GX platform switches



Note Prior to NX-OS Release 9.3(5), if any of the above switches were configured with uRPF, standard, non-disruptive ISSU was not supported.

Enhanced mode

- Beginning with NX-OS Release 10.3(3)F, only the LXC mode is supported on Nexus 9300-FX3 and 9300-GX switches, which allows you to perform enhanced non-disruptive ISSU with minimal downtime.
- ND ISSU can be performed in LXC mode in two methods:
 - ND ISSU in LXC mode - Switchover-based ISSU that is similar to EOR. Second SUP is brought up in new container and switchover is done. The second SUP now becomes the new active. There is no change to the kernel.
 - Fallback ND LXC ISSU - This is only done when the above switchover-based ISSU cannot be done (SRG Kernel incompatible or less memory). The kernel is upgraded.
 - skip-kernel-upgrade option will force ND ISSU in LXC mode - Switchover-based ISSU (even in case when running) and target kernels are incompatible.
- For switches that are in LXC boot mode, the enhanced LXC upgrade will fall back to standard ND ISSU as the target image kernels are likely be different than the current image.

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 bootflash. 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](#).
- When the PTP feature is configured on Nexus 9300-H2R and C93400LD-H1 switches, performing an ND ISSU upgrade from version 10.4(1)F to 10.5(2)F triggers a kernel panic, resulting in an additional switch reload and causing a traffic outage. For more information, refer to [CSCwh34732](#).
- While upgrading from NX-OS releases prior to 10.4(3) to 10.4(3) or later releases with **mode tap-aggregation** command enabled on the Layer 2 interface, make sure to enable the global **hardware acl tap-agg** command and reload.
- 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 on Nexus 9300 and 92348GC-X platform switches; and on 9300-HX platform switches from NX-OS Release 10.4(3)F. However, the following features are not supported on nondisruptive ISSU:
 - SR L2EVPN
 - ISIS and OSPF underlay
 - vPC configuration with segment-routing
 - Egress Peer engineering
 - Segment routing and GRE co existence

- While upgrading from NX-OS releases prior to 10.2(2)F to 10.2(2)F or later releases, configure the **mode tap-aggregation** command before attaching TapAgg ACLs on Layer 2 interface.
- If you upgrade from a NX-OS release that supports the CoPP feature to a NX-OS release that supports the CoPP feature with additional classes for new protocols, you must either run the setup utility using the **setup** command or use the **copp profile** command for the new CoPP classes to be available. For more information on these commands of configuring control plane policing, see the [Cisco Nexus 9000 Series NX-OS Security Configuration Guide](#).
- Beginning with NX-OS Release 10.1(2), CoPP is supported on N9K-X9624D-R2 and N9K-C9508-FM-R2 platform switches.
- Beginning with NX-OS Release 10.1(2), RACL is supported on N9K-X9624D-R2 and N9K-C9508-FM-R2 platform switches.
- Beginning with NX-OS Release 10.1(1), during the disruptive upgrade to the 64-bit image or a downgrade from 64-bit to 32-bit image, if feature ITD is enabled, refer to *Guidelines and Limitations for ITD* in the *Cisco Nexus 9000 Series NX-OS Intelligent Traffic Director Configuration Guide, Release 10.1(x)*, if the upgrade or downgrade proceeds with an ASIC reload.
- Beginning with NX-OS Release 10.1(1), **Fs_daemon** does not support **snmpwalk** on devices with more than 5000 files. When performing snmpwalk on a device with more than 5000 files, the error `resourceUnavailable (This is likely a out-of-memory failure within the agent)` is an expected behavior.
- When you upgrade a Nexus 9000 device to NX-OS Release 10.1(x), if a QSFP port is configured with the manual breakout command and is using a QSA, the configuration of the interface Ethernet 1/50/1 is no longer supported and must be removed. To restore the configuration, you must manually configure the interface Ethernet 1/50 on the device.
- When upgrading from NX-OS Release 9.3(3) to NX-OS Release 9.3(6) or later, if you do not retain configurations of the TRM enabled VRFs from NX-OS Release 9.3(3), or if you create new VRFs after the upgrade, the auto-generation of **ip multicast multipath s-g-hash next-hop-based** command, when feature **ngmvpn** is enabled, will not happen. You must enable the command manually for each TRM enabled VRF. For the configuration instructions, see the [Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide](#).
- Upgrading from NX-OS Release 9.3(1), 9.3(2) or 9.3(3) to a higher release, with Embedded Event Manager (EEM) configurations that are saved to the running configuration, may cause a DME error to be presented. The error is in the output of the **show consistency-checker dme running-config enhanced** command, specifically, the event manager commands. If this error occurs, delete all EEM applet configurations after completing the ISSU, then reapply the EEM configurations.
- When redistributing static routes, NX-OS requires the **default-information originate** command to successfully redistribute the default static route starting in 7.0(3)I7(6).

FEX

- When upgrading from NX-OS Release 9.3x to NX-OS Release 10.4x using the **install all** command, ensure that storm control is disabled on all attached Fabric Extenders (FEXs). If you do not disable storm control, the switch fails to boot up after the upgrade. (See [CSCws43646](#).)
- Beginning with NX-OS Release 10.2(2)F, ND ISSU is supported for FEX and you need to re-adjust the BGP **graceful-restart restart time** command for the upgrade to work non-disruptively. This must be done for each FEX upgrade one-by-one.

The following example shows the time taken to re-adjust `bgp-graceful restart-time` for each non-disruptive FEX upgrade.

In the Non-disruptive upgrade with FEX, each FEX will upgrade taking about 90 seconds (1.5 minutes) sequentially (one-by-one and not a parallel upgrade).

Total non-disruptive upgrade time for all FEX = No. of fex * time taken per fex
 For 10 FEX = 10 * 90
 = 900 seconds or 15 minutes

- When you upgrade a Nexus 9000 switch from NX-OS Release 7.x with an attached FEX in straight-through mode to 9.x and then to 10.x, the FEX Layer 2 Host Interface (HIF) configuration can be lost after upgrading to a 10.x Release. This occurs due to a design change in handling Layer 2 FEX HIF ports at boot time from Release 9.x to 10.x.



Note The issue occurs only for FEX connected in a straight-through mode and not for dual-homed (A-A) mode.

To resolve this, run the following non-intrusive commands before upgrading the switch from 9.x to 10.x:

1. Apply **no switchport** only on all Layer 3 (L3) physical and Layer 3 (L3) port-channel interfaces. For example,

```
switch(config)# interface e1/1
switch(config-if)# no switchport
```

2. Configure **system default switchport** globally and save the configuration. For example,

```
switch(config)# system default switchport
switch(config)# copy r s
```



Note The issue does not occur if:

- the switch was originally booted in 9.x with an attached FEX and then upgraded to 10.x.
 - the switch was upgraded from 7.x to 9.x without an attached FEX, and the FEX was added later in 9.x before upgrading to 10.x.
-

- An upgrade that is performed via the **install all** command for NX-OS Release 7.0(3)I2(2b) to Release 10.1(x) might result in the VLANs being unable to be added to the existing FEX HIF trunk ports. To recover from this, the following steps should be performed after all FEXs have come online and the HIFs are operationally up:

1. Enter the copy run bootflash:fex_config_restore.cfg command at the prompt.
2. Enter the copy bootflash:fex_config_restore.cfg running-config echo-commands command at the prompt.

- In NX-OS Release 7.0(3)I6(1) and earlier, performing an ASCII replay or running the copy file run command on a FEX HIF configuration requires manually reapplying the FEX configuration after the FEX comes back up.

FC/FCoE NPV

- Beginning with NX-OS Release 10.1(1), ISSU is supported on FC/FCoE switch mode on Nexus 93360YC-FX2. For more information about the FC/FCoE switch mode and supported hardware, see the [Cisco Nexus 9000 Series NX-OS SAN Switching Configuration Guide](#).
- Beginning with NX-OS Release 10.1(1), enhanced ISSU is supported on FC/FCoE switch mode for Nexus 93180YC-FX and 93360YC-FX2 switches. For more information about the FC/FCoE switch mode and supported hardware, see the [Cisco Nexus 9000 Series NX-OS SAN Switching Configuration Guide](#).
- Beginning with NX-OS Release 10.1(1), Enhanced ISSU is supported on FC/FCoE NPV mode for Nexus 93180YC-FX and 93360YC-FX2 switches. For more information about the FC/FCoE NPV mode and supported hardware, see the [Cisco Nexus 9000 Series NX-OS FC-NPV and FCoE NPV Configuration Guide](#).

VXLAN with TRM



Caution Following changes must be done during a maintenance window.

After upgrading a NX-OS 9000 Series switches configured with VXLAN (specifically VRF-related configurations) from NX-OS Release 7.x through 9.3 to 10.3(6) or earlier, two issues arise:

- The startup-config displays both legacy and new Layer 3 VNID configuration modes
- TRM traffic's RPF changes to the new mode for S,Gs, causing multicast traffic forwarding problems.

To avoid these issues, follow these steps:

- Enable the REST configuration input using the following commands:

```
feature nxapi
  nxapi http port 80
```

- Open a browser and enter the management IP address of the switch. This will open the Sandbox page. Use the same credentials as the switch admin login to sign in.
- In the top input textbox, enter the following command for each VRF that has an issue with the VNI ID:


```
vrf context tenant-1
  no vni 50000 13
```
- On the right side of the page, set the Method to **NXAPI-REST (DME)** and keep the Input Type as **cli**.
- Click the **Convert (with DN)** button in the middle of the page. This will generate the XML equivalent of the configuration change.
- When the XML appears in the second textbox, click **Send** to apply the changes and remove the VNI ID configuration from the switch.
- To ensure the changes are applied, run the command:

```
copy running-config startup-config
```

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 Note N9K-X97160YC-EX line card is supported.	10.6(1)F
N9K-C93180YC-EX and N9K-C93108TC-EX	10.4(x)
N9K-X9732C-EXM line card	10.3(x)
N9K-C9364C-GX	9.3(16)
N9K-C93600CD-GX	9.3(16)
N9K-C9316D-GX	9.3(16)
N9K-C93180LC-EX	9.3(x)

MACsec non-disruptive ISSU

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

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. 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.`

ISSU for Nexus 9200 platform switches

ISSU Type	Release and Supported Platforms	Features Not Supported with Non-disruptive ISSU
Standard	Beginning with NX-OS Release 7.0(3)I6(1): Nexus 92300YC Beginning with NX-OS Release 9.3(3): Nexus 92348GC-X	Both ISSU types are disruptive for Nexus 9200 platform switches configured with features such as <ul style="list-style-type: none"> • Segment routing • Tetration
Enhanced	Nexus 92300YC	<p>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 Nexus 92348GC-X platform switches. For more information about the features that are not supported, see NX-OS software upgrade guidelines, on page 6.</p> <p>Note Beginning with NX-OS Release 10.4(3)F, non-disruptive ISSU is not supported on Nexus 92348GC-X. For more information about the features that are not supported, see NX-OS software upgrade guidelines, on page 6.</p>

ISSU for Nexus 9300 platform switches

ISSU Type	Release and Supported Platforms	Features Not Supported with Non-disruptive ISSU
Standard	Beginning with NX-OS Release 9.3(3): Nexus 9332C Nexus 9364C Note ISSU on Nexus 9300 platform switches is supported when the switch is the spanning tree root. You can use the show spanning-tree issu-impact command to verify if the switch meets this criteria.	Both ISSU types are disruptive for Nexus 9300 platform switches configured with features such as <ul style="list-style-type: none"> • Dual-homed FEX • Segment routing • MACsec 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 Nexus 9300 platform switches. For more information about the features that are not supported, see NX-OS software upgrade guidelines, on page 6 .
Enhanced	Beginning with NX-OS Release 9.3(5): Nexus 9332C Nexus 9364C Note ISSU on Nexus 9300 platform switches is supported when the switch is the spanning tree root. You can use the show spanning-tree issu-impact command to verify if the switch meets this criteria.	

ISSU for Nexus 9300-X platform switches

ISSU Type	Release and Supported Platforms	Features Not Supported with Non-disruptive ISSU
Standard	Beginning with NX-OS Release 10.2(3)F, VPC Fabric peering is supported on Nexus 9300-X TORs.	Beginning with NX-OS Release 10.2(3)F, the VXLAN and VPC features that are not supported during non- disruptive ISSU for VPC Fabric Peering include <ul style="list-style-type: none"> • TRM • VXLAN IPv6 underlay • Proportional Multipath for VNF • VXLAN Flood-and-learn • HSRP and VRRP • VXLAN Cloudsec • VXLAN to SR Handoff and all Handoff features • Multi-Site, and • MACsec <p>Note Beginning with NX-OS Release 10.4(2)F, non-disruptive ISSU is supported for VXLAN to SR Handoff with BGP as underlay on Nexus 9300-X platform switches.</p>
Enhanced	Beginning with NX-OS Release 10.2(3)F, VPC Fabric peering is supported on Nexus 9300-X TORs.	

ISSU for Nexus 9300-EX platform switches

ISSU Type	Release and Supported Platforms	Features Not Supported with Non-disruptive ISSU
Standard	Beginning with NX-OS Release 7.0(3)I6(1): Nexus 93108TC-EX Nexus 93180YC-EX	Both ISSU types are disruptive for Nexus 9300-EX platform switches configured with features such as <ul style="list-style-type: none"> • Segment routing • Tetration, and • MACsec Note Beginning with NX-OS Release 10.2(1), both ISSU types are non-disruptive for Nexus 9300-EX platform switches configured with Straight-Through FEX and Dual-Homed FEX.
Enhanced	Beginning with NX-OS Release 7.0(3)I7(3): Nexus 93108TC-EX Nexus 93180YC-EX	

ISSU for Nexus 9300-FX platform switches

ISSU Type	Release and Supported Platforms	Features Not Supported with Non-disruptive ISSU
Standard	<p>NX-OS Release 9.3(1) and 9.3(2): None</p> <p>Beginning with NX-OS Release 9.3(3):</p> <p>Nexus 9336C-FX2</p> <p>Nexus 93240YC-FX2</p> <p>Nexus 93240YC-FX2Z</p> <p>Nexus 9348GC-FXP</p> <p>Nexus 93108TC-FX</p> <p>Nexus 93180YC-FX</p> <p>Beginning with NX-OS Release 10.2(8)M:</p> <p>Nexus 93180YC-FX3</p> <p>Nexus 93180YC-FX3S</p> <p>Beginning with NX-OS Release 10.4(1)F:</p> <p>9348GC-FX3</p> <p>9348GC-FX3PH</p> <p>Beginning with NX-OS Release 10.4(2)F:</p> <p>93108TC-FX3</p>	<p>Standard ISSU is disruptive for Nexus 9300-FX platform switches configured with features such as</p> <ul style="list-style-type: none"> • Segment Routing • TRM Feature, and • MACsec <p>Note Beginning with NX-OS Release 10.2(1), Standard ISSU is non-disruptive for Nexus 9300-FX platform switches configured with Straight-Through FEX and Dual-Homed FEX.</p> <p>Beginning with NX-OS Release 10.3(3)F, standard ISSU is not supported on Nexus 93180YC-FX3 and FX3S platform switches.</p> <p>Beginning with NX-OS Release 10.4(1)F, standard ISSU is not supported on Nexus 9300-FX and 9300-FX2 platform switches.</p> <p>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 Nexus 9300 platform switches. For more information about the features that are not supported, see NX-OS software upgrade guidelines, on page 6.</p>

ISSU Type	Release and Supported Platforms	Features Not Supported with Non-disruptive ISSU
Enhanced	<p>NX-OS Release 9.3(1), 9.3(2), and 9.3(3): None</p> <p>Beginning with NX-OS Release 9.3(5):</p> <p>Nexus 9336C-FX2</p> <p>Nexus 93240YC-FX2</p> <p>Nexus 93216TC-FX2</p> <p>Nexus 93360YC-FX2</p> <p>Nexus 93240YC-FX2Z</p> <p>Nexus 9348GC-FXP</p> <p>Nexus 93108TC-FX</p> <p>Nexus 93180YC-FX</p> <p>Beginning with NX-OS Release 10.1(1), Enhanced ISSU is supported on the following platforms with FC/FCoE features:</p> <p>Nexus 93360YC-FX2</p> <p>Nexus 93180YC-FX</p> <p>Beginning with NX-OS Release 10.2(8)M, Enhanced ISSU is supported on the following platforms:</p> <p>Nexus 93180YC-FX3</p> <p>Nexus 93180YC-FX3S</p> <p>Beginning with NX-OS Release 10.4(1)F:</p> <p>9348GC-FX3</p> <p>9348GC-FX3PH</p> <p>Beginning with NX-OS Release 10.4(2)F:</p> <p>93108TC-FX3</p> <p>Beginning with NX-OS Release 10.2(2)F, Enhanced ISSU is supported on the following platform with FC/FCoE features:</p> <p>N9K-C9336C-FX2-E</p>	

ISSU Type	Release and Supported Platforms	Features Not Supported with Non-disruptive ISSU
		<p>Enhanced ISSU is disruptive for Nexus 9300-FX platform switches configured with features such as</p> <ul style="list-style-type: none"> • Segment Routing • TRM Feature, and • MACsec <p>Note In NX-OS Releases 9.3(x), Enhanced ISSU on Nexus 93360YC-FX2 and Nexus 93180YC-FX with FC/FCoE features will be disruptive.</p> <p>Note Beginning with NX-OS Release 10.2(1), Enhanced ISSU is non-disruptive for Nexus 9300-FX platform switches configured with Straight-Through FEX and Dual-Homed FEX.</p> <p>Beginning with NX-OS Release 10.3(3)F, only the LXC mode is supported on Nexus 9300-FX3 and 9300-FX3S switches, which allows you to perform enhanced non-disruptive ISSU with minimal downtime.</p> <p>Beginning with NX-OS Release 10.4(1)F, only the LXC mode is supported on Nexus 9300-FX and 9300-FX2 switches, which allows you to perform enhanced non-disruptive ISSU with minimal downtime.</p> <p>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 Nexus 9300 platform switches. For more information about the features that are not supported, see NX-OS software</p>

ISSU Type	Release and Supported Platforms	Features Not Supported with Non-disruptive ISSU
		upgrade guidelines, on page 6.

ISSU for Nexus 9300-GX platform switches

ISSU Type	Release and Supported Platforms	Features Not Supported with Non-disruptive ISSU
Standard	<p>Beginning with NX-OS Release 10.2(8)M:</p> <ul style="list-style-type: none"> • Nexus 9364C-GX • Nexus 9316D-GX • Nexus 93600CD-GX <p>Note Beginning with NX-OS Release 10.3(3)F, standard ISSU is not supported on Nexus 9300-GX platform switches.</p>	<ul style="list-style-type: none"> • TRM Feature • Segment Routing <p>Note Beginning with NX-OS Release 10.3(3)F, standard ISSU is not supported on Nexus 9300-GX platform switches.</p> <p>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 Nexus 9300 platform switches. For more information about the features that are not supported, see NX-OS software upgrade guidelines, on page 6.</p>
Enhanced	<p>Beginning with NX-OS Release 10.2(8)M:</p> <ul style="list-style-type: none"> • Nexus 9364C-GX • Nexus 9316D-GX • Nexus 93600CD-GX <p>Beginning with NX-OS Release 10.2(2)F, Enhanced ISSU is supported on Nexus 9300-GX2B platform switches.</p> <p>Beginning with NX-OS Release 10.2(3)F, Enhanced ISSU is supported on Nexus 9300-GX2A platform switches.</p> <p>Beginning with NX-OS Release 10.3(3)F, only the LXC mode is supported on Nexus 9300-GX switches, which allows you to perform enhanced non-disruptive ISSU with minimal downtime.</p>	<ul style="list-style-type: none"> • TRM Feature • Segment Routing <p>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 Nexus 9300 platform switches. For more information about the features that are not supported, see NX-OS software upgrade guidelines, on page 6.</p>

ISSU for Nexus 9300-HX platform switches

ISSU Type	Release and Supported Platforms	Features Not Supported with Non-disruptive ISSU
Enhanced	<p>Boot mode LXC ISSU is supported by default on the following Nexus C9300-HX platform switches:</p> <ul style="list-style-type: none"> • 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 <p>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.</p>	<ul style="list-style-type: none"> • SR L2EVPN • ISIS and OSPF underlay • vPC configuration with segment-routing • Egress Peer engineering • Segment routing and GRE co-existence • MACsec

ISSU for Nexus 9400 platform switches

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.	<ul style="list-style-type: none"> • TRM Feature • Segment Routing • MACsec

Upgrade paths

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.

Install upgrade patches

On Nexus 9500 series switches only, a software upgrade from NX-OS Release 7.0(3)I1(2), 7.0(3)I1(3), or 7.0(3)I1(3a) to any other NX-OS release requires installing two patches prior to upgrading using the **install all** command. These patches are available for each respective release and can be downloaded using the links below.



Caution Failing to follow this procedure could require console access in order to recover the switch after the upgrade.



Note These patches are only for upgrading. After the upgrade, the patch is automatically removed. If you decide not to upgrade after installing the patches, do not deactivate it. Deactivating the patch may cause a bios_daemon crash.

[Cisco NX-OS Release 7.0\(3\)I1\(2\) Upgrade Patch](#)

[Cisco NX-OS Release 7.0\(3\)I1\(3\) Upgrade Patch](#)

[Cisco NX-OS Release 7.0\(3\)I1\(3a\) Upgrade Patch](#)

To install these patches prior to upgrading using the **install all** command, follow the instructions shown below. An example is demonstrated below with an NX-OS software patch and upgrade from 7.0(3)I1(2) to 7.0(3)I7(1):

Procedure

Step 1 Add both patches with the **install add bootflash: {patch-file.bin}** command.

Example:

```
switch(config)# install add bootflash:n9000-dk9.7.0.3.I1.2.CSCuy16604.bin
Install operation 16 completed successfully at Thu Mar 3 04:24:13 2016
switch(config)# install add bootflash:n9000-dk9.7.0.3.I1.2.CSCuy16606.bin
Install operation 17 completed successfully at Thu Mar 3 04:24:43 2016
```

Step 2 Activate both patches with the **install activate {patch-file.bin}** command.

Example:

```
switch(config)# install activate n9000-dk9.7.0.3.I1.2.CSCuy16604.bin
Install operation 18 completed successfully at Thu Mar 3 04:28:38 2016
switch (config)# install activate n9000-dk9.7.0.3.I1.2.CSCuy16606.bin
Install operation 19 completed successfully at Thu Mar 3 04:29:08 2016
```

Step 3 Commit both patches with the **install commit {patch-file.bin}** command.

Example:

```
switch(config)# install commit n9000-dk9.7.0.3.I1.2.CSCuy16604.bin
Install operation 20 completed successfully at Thu Mar 3 04:30:38 2016
switch (config)# install commit n9000-dk9.7.0.3.I1.2.CSCuy16606.bin
Install operation 21 completed successfully at Thu Mar 3 04:31:16 2016
```

Step 4 Proceed with a software upgrade to the chosen target release with the **install all** command.

Example:

```

switch (config)# install all nxos bootflash:nxos.7.0.3.I7.1.bin
Installer will perform compatibility check first. Please wait.
uri is: /nxos.7.0.3.I7.1.bin
Installer is forced disruptive

Verifying image bootflash:/nxos.7.0.3.I7.1.bin for boot variable "nxos".
[#####] 100% -- SUCCESS

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

Preparing "lcn9k" version info using image bootflash:/nxos.7.0.3.I7.1.bin.
[#####] 100% -- SUCCESS

Preparing "bios" version info using image bootflash:/nxos.7.0.3.I7.1.bin.
[#####] 100% -- SUCCESS

Preparing "lcn9k" version info using image bootflash:/nxos.7.0.3.I7.1.bin.
[#####] 100% -- SUCCESS

Preparing "lcn9k" version info using image bootflash:/nxos.7.0.3.I7.1.bin.
[#####] 100% -- SUCCESS

Preparing "lcn9k" version info using image bootflash:/nxos.7.0.3.I7.1.bin.
[#####] 100% -- SUCCESS

Preparing "lcn9k" version info using image bootflash:/nxos.7.0.3.I7.1.bin.
[#####] 100% -- SUCCESS

Preparing "lcn9k" version info using image bootflash:/nxos.7.0.3.I7.1.bin.
[#####] 100% -- SUCCESS

Preparing "lcn9k" version info using image bootflash:/nxos.7.0.3.I7.1.bin.
[#####] 100% -- SUCCESS

Preparing "lcn9k" version info using image bootflash:/nxos.7.0.3.I7.1.bin.
[#####] 100% -- SUCCESS

Preparing "lcn9k" version info using image bootflash:/nxos.7.0.3.I7.1.bin.
[#####] 100% -- SUCCESS

Preparing "lcn9k" version info using image bootflash:/nxos.7.0.3.I7.1.bin.
[#####] 100% -- SUCCESS

Preparing "lcn9k" version info using image bootflash:/nxos.7.0.3.I7.1.bin.
[#####] 100% -- SUCCESS

Preparing "lcn9k" version info using image bootflash:/nxos.7.0.3.I7.1.bin.
[#####] 100% -- SUCCESS

Preparing "lcn9k" version info using image bootflash:/nxos.7.0.3.I7.1.bin.
[#####] 100% -- SUCCESS

Preparing "lcn9k" version info using image bootflash:/nxos.7.0.3.I7.1.bin.
[#####] 100% -- SUCCESS

Preparing "nxos" version info using image bootflash:/nxos.7.0.3.I7.1.bin.
[#####] 100% -- SUCCESS

Preparing "lcn9k" version info using image bootflash:/nxos.7.0.3.I7.1.bin.

```

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

```
Preparing "lcn9k" version info using image bootflash:/nxos.7.0.3.I7.1.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	Incompatible image
6	yes	disruptive	reset	Incompatible image
8	yes	disruptive	reset	Incompatible image
9	yes	disruptive	reset	Incompatible image
10	yes	disruptive	reset	Incompatible image
11	yes	disruptive	reset	Incompatible image
14	yes	disruptive	reset	Incompatible image
15	yes	disruptive	reset	Incompatible image
16	yes	disruptive	reset	Incompatible image
21	yes	disruptive	reset	Incompatible image
22	yes	disruptive	reset	Incompatible image
23	yes	disruptive	reset	Incompatible image
24	yes	disruptive	reset	Incompatible image
25	yes	disruptive	reset	Incompatible image
26	yes	disruptive	reset	Incompatible image
27	yes	disruptive	reset	Incompatible image
28	yes	disruptive	reset	Incompatible image
29	yes	disruptive	reset	Incompatible image
30	yes	disruptive	reset	Incompatible image

```
Images will be upgraded according to following table:
```

Module	Image	Running-Version(pri:alt)	New-Version	Upg-Required
1	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)	yes
1	bios	v01.42(00:v01.42(00	v01.48(00	yes
6	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)	yes
6	bios	v01.48(00:v01.48(00	v01.48(00	no
8	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)	yes
8	bios	v01.48(00:v01.29(00	v01.48(00	no
9	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)	yes
9	bios	v01.48(00:v01.35(00	v01.48(00	no
10	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)	yes
10	bios	v01.48(00:v01.42(00	v01.48(00	no
11	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)	yes
11	bios	v01.48(00:v01.52(00	v01.48(00	no
14	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)	yes
14	bios	v01.48(00:v01.48(00	v01.48(00	no
15	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)	yes
15	bios	v01.48(00:v01.40(00	v01.48(00	no
16	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)	yes
16	bios	v01.48(00:v01.42(00	v01.48(00	no
21	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)	yes
21	bios	v01.48(00:v01.42(00	v01.48(00	no
22	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)	yes
22	bios	v01.48(00:v01.40(00	v01.48(00	no
23	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)	yes
23	bios	v01.48(00:v01.40(00	v01.48(00	no
24	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)	yes
24	bios	v01.48(00:v01.40(00	v01.48(00	no
25	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)	yes

Install upgrade patches

25	bios	v01.48(00:v01.40(00	v01.48(00	no
26	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)	yes
26	bios	v01.48(00:v01.40(00	v01.48(00	no
27	nxos	7.0(3)I1(2)	7.0(3)I7(1)	yes
27	bios	v08.06(09/10/2014):v08.18(08/11/2015)	v08.26(01/12/2016)	yes
28	nxos	7.0(3)I1(2)	7.0(3)I7(1)	yes
28	bios	v08.06(09/10/2014):v08.26(01/12/2016)	v08.26(01/12/2016)	yes
29	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)	yes
29	bios	v01.48(00:v01.35(00	v01.48(00	no
30	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)	yes
30	bios	v01.48(00:v01.35(00	v01.48(00	no

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.

Performing runtime checks.

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

Syncing image bootflash:/nxos.7.0.3.I7.1.bin to standby.

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

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 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 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 9: Refreshing compact flash and upgrading bios/loader/bootrom.

Warning: please do not remove or power off the module at this time.

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

Module 10: Refreshing compact flash and upgrading bios/loader/bootrom.

Warning: please do not remove or power off the module at this time.

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

Module 11: Refreshing compact flash and upgrading bios/loader/bootrom.

Warning: please do not remove or power off the module at this time.

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

Module 14: Refreshing compact flash and upgrading bios/loader/bootrom.

Warning: please do not remove or power off the module at this time.

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

Module 15: Refreshing compact flash and upgrading bios/loader/bootrom.

Warning: please do not remove or power off the module at this time.

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

Module 16: 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.
[#####] 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.
[#####] 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.
[#####] 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.
[#####] 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.
[#####] 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.
[#####] 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

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

Module 29: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#####] 100% -- SUCCESS

Module 30: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#####] 100% -- SUCCESS
Finishing the upgrade, switch will reboot in 10 seconds.
switch(config)#
User Access Verification

switch login:
[ 2644.917727] [1456980048] writing reset reason 88,

CISCO SWITCH Ver 8.26

CISCO SWITCH Ver 8.26
Memory Size (Bytes): 0x0000000080000000 + 0x0000000380000000
Relocated to memory
Time: 6/3/2016 4:41:8
Detected CISCO IOFPGA
Booting from Primary Bios
Code Signing Results: 0x0
Using Upgrade FPGA
FPGA Revision      : 0x27
FPGA ID           : 0x1168153
FPGA Date         : 0x20160111
Reset Cause Register: 0x22
Boot Ctrl Register : 0x60ff
EventLog Register1 : 0x2000000
EventLog Register2 : 0xfbe77fff
```

```

Version 2.16.1240. Copyright (C) 2013 American Megatrends, Inc.
Board type 1
IOFPGA @ 0xe8000000
SLOT_ID @ 0x1b
Standalone chassis
check_bootmode: grub: Continue grub
Trying to read config file /boot/grub/menu.lst.local from (hd0,4)
  Filesystem type is ext2fs, partition type 0x83

Booting bootflash:/nxos.7.0.3.I7.1.bin ...
Booting bootflash:/nxos.7.0.3.I7.1.bin
Trying diskboot
  Filesystem type is ext2fs, partition type 0x83
IOFPGA ID: 1168153
Image valid

Image Signature verification was Successful.

Boot Time: 3/3/2016 4:41:44
INIT: version 2.88 booting
Unsquashing rootfs ...

Loading IGB driver ...
Installing SSE module ... done
Creating the sse device node ... done
Loading I2C driver ...
Installing CTRL driver for card_type 3 ...
CTRL driver for card_index 21000 ...
old data: 4000004 new data: 1
Not Micron SSD...

Checking all filesystems.....
Installing default sptom values ...
  done.Configuring network ...
Installing LC netdev ...
Installing psdev ...
Installing veobc ...
Installing OBFL driver ...
mounting plog for N9k!
tune2fs 1.42.1 (17-Feb-2012)
Setting reserved blocks percentage to 0% (0 blocks)
Starting portmap daemon...
creating NFS state directory: done
starting 8 nfsd kernel threads: done
starting mountd: done
starting statd: done
Saving image for img-sync ...
Loading system software
Installing local RPMS
Patch Repository Setup completed successfully
dealing with default shell..
file /proc/cmdline found, look for shell
unset shelltype, nothing to do..
user add file found..edit it
Uncompressing system image: Thu Jun 3 04:42:11 UTC 2016
blogger: nothing to do.

..done Thu Mar 3 04:42:11 UTC 2016
Creating /dev/mcelog
Starting mcelog daemon
Overwriting dme stub lib
Replaced dme stub lib
INIT: Entering runlevel: 3

```

Running S93thirdparty-script...

```

2016 Mar  3 04:42:37 switch%$ VDC-1 %$ %USER-2-SYSTEM_MSG: <<%USBHSD-2-MOUNT>> logflash: online -
usbhsd
2016 Mar  3 04:42:37 switch%$ VDC-1 %$ Mar  3 04:42:37 %KERN-2-SYSTEM_MSG: [ 12.509615] hwport
mode=6 - kernel
2016 Mar  3 04:42:40 switch%$ VDC-1 %$ %VMAN-2-INSTALL_STATE: Installing virtual service 'guestshell+'
2016 Mar  3 04:42:40 switch%$ VDC-1 %$ %DAEMON-2-SYSTEM_MSG: <<%ASCII-CFG-2-CONF_CONTROL>> Binary
restore - ascii-cfg[13904]
2016 Mar  3 04:42:40 switch%$ VDC-1 %$ %DAEMON-2-SYSTEM_MSG: <<%ASCII-CFG-2-CONF_CONTROL>> Restore
DME database - ascii-cfg[13904]
2016 Mar  3 04:42:42 switch%$ VDC-1 %$ netstack: Registration with cli server complete
2016 Mar  3 04:43:00 switch%$ VDC-1 %$ %USER-2-SYSTEM_MSG: ssnmgr_app_init called on ssnmgr up -
aclmgr
2016 Mar  3 04:43:09 switch%$ VDC-1 %$ %USER-0-SYSTEM_MSG: end of default policer - copp
2016 Mar  3 04:43:10 switch%$ VDC-1 %$ %VMAN-2-INSTALL_STATE: Install success virtual service
'guestshell+'; Activating
2016 Mar  3 04:43:10 switch%$ VDC-1 %$ %VMAN-2-ACTIVATION_STATE: Activating virtual service
'guestshell+'
2016 Mar  3 04:43:13 switch%$ VDC-1 %$ %CARDCLIENT-2-FPGA_BOOT_PRIMARY: IOFPGA booted from Primary
2016 Mar  3 04:43:18 switch%$ VDC-1 %$ %USER-2-SYSTEM_MSG: IPV6 Netlink thread init successful -
icmpv6
2016 Mar  3 04:43:19 switch%$ VDC-1 %$ %VDC_MGR-2-VDC_ONLINE: vdc 1 has come online

```

User Access Verification

switchlogin:

```

2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence of Module 1
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence of Module 6
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence of Module 8
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence of Module 9
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence of Module 10
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence of Module 11
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence of Module 14
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence of Module 15
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence of Module 16
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence of Module 21
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence of Module 22
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence of Module 23
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence of Module 24
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence of Module 25
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence of Module 26
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence of Module 28
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence of Module 29
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence of Module 30
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-PS_OK: Power supply 1 ok (Serial number XYZ284014RR)
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-PS_FANOK: Fan in Power supply 1 ok
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-PS_OK: Power supply 2 ok (Serial number XYZ285111TC)
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-PS_FANOK: Fan in Power supply 2 ok
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-PS_OK: Power supply 3 ok (Serial number XYZ285111QQ)
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-PS_FANOK: Fan in Power supply 3 ok
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-PS_OK: Power supply 4 ok (Serial number XYZ284014TI)
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-PS_FANOK: Fan in Power supply 4 ok
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-PS_OK: Power supply 5 ok (Serial number XYZ284014TS)
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-PS_FANOK: Fan in Power supply 5 ok
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-FANMOD_FAN_OK: Fan module 1 (Fan1(sys_fan1) fan)
ok
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-FANMOD_FAN_OK: Fan module 2 (Fan2(sys_fan2) fan)
ok
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-FANMOD_FAN_OK: Fan module 3 (Fan3(sys_fan3) fan)
ok
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 30 detected (Serial number
ABC1234DE56) Module-Type System Controller Model N9K-SC-A
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 30 powered up (Serial number

```

```

ABC1234DE56)
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 28 detected (Serial number
:unavailable) Module-Type Supervisor Module Model :unavailable
2016 Mar 3 04:43:58 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 29 detected (Serial number
ABC1234DEFG) Module-Type System Controller Model N9K-SC-A
2016 Mar 3 04:43:58 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 29 powered up (Serial number
ABC1234DEFG)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 21 detected (Serial number
ABC1213DEFG) Module-Type Fabric Module Model N9K-C9516-FM
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 22 detected (Serial number
ABC1211DEFG) Module-Type Fabric Module Model N9K-C9516-FM
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 21 powered up (Serial number
ABC1213DEFG)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 22 powered up (Serial number
ABC1211DEFG)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 23 detected (Serial number
ABC1234D5EF) Module-Type Fabric Module Model N9K-C9516-FM
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 23 powered up (Serial number
ABC1234D5EF)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 24 detected (Serial number
ABC1211DE3F) Module-Type Fabric Module Model N9K-C9516-FM
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 24 powered up (Serial number
ABC1211DE3F)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 25 detected (Serial number
ABC1213DEFG) Module-Type Fabric Module Model N9K-C9516-FM
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 25 powered up (Serial number
ABC1213DEFG)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 26 detected (Serial number
ABC1211DE34) Module-Type Fabric Module Model N9K-C9516-FM
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 26 powered up (Serial number
ABC1211DE34)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED: All Ejectors closed
for module 1. Ejector based shutdown enabled
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 1 detected (Serial number
ABC1217DEFG) Module-Type 32p 40G Ethernet Module Model N9K-X9432PQ
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 1 powered up (Serial number
ABC1217DEFG)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED: All Ejectors closed
for module 9. Ejector based shutdown enabled
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 9 detected (Serial number
ABC1236D4E5) Module-Type 48x1/10G-T 4x40G Ethernet Module Model N9K-X9564TX
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 9 powered up (Serial number
ABC1236D4E5)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED: All Ejectors closed
for module 10. Ejector based shutdown enabled
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 10 detected (Serial number
ABC1217EFGH) Module-Type 32p 40G Ethernet Module Model N9K-X9432PQ
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 10 powered up (Serial number
ABC1217EFGH)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED: All Ejectors closed
for module 11. Ejector based shutdown enabled
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 11 detected (Serial number
ABC123DEF4) Module-Type 36p 40G Ethernet Module Model N9K-X9536PQ
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 11 powered up (Serial number
ABC123DEF4)
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED: All Ejectors closed
for module 15. Ejector based shutdown enabled
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 15 detected (Serial number
ABC1212DEFG) Module-Type 36p 40G Ethernet Module Model N9K-X9536PQ
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 15 powered up (Serial number
ABC1212DEFG)
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED: All Ejectors closed
for module 16. Ejector based shutdown enabled
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 16 detected (Serial number

```

```

ABCD1235DEFG) Module-Type 48x1/10G SFP+ 4x40G Ethernet Module Model N9K-X9464PX
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 16 powered up (Serial number
ABCD1235DEFG)
2016 Mar 3 04:44:08 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED: All Ejectors closed
for module 14. Ejector based shutdown enabled
2016 Mar 3 04:44:08 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 14 detected (Serial number
ABC9876DE5F) Module-Type 8p 100G Ethernet Module Model N9K-X9408PC-CFP2
2016 Mar 3 04:44:08 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 14 powered up (Serial number
ABC9876DE5F)
2016 Mar 3 04:44:09 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED: All Ejectors closed
for module 6. Ejector based shutdown enabled
2016 Mar 3 04:44:09 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 6 detected (Serial number
ABC9876DE3F) Module-Type 8p 100G Ethernet Module Model N9K-X9408PC-CFP2
2016 Mar 3 04:44:09 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 6 powered up (Serial number
ABC9876DE3F)
2016 Mar 3 04:44:10 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED: All Ejectors closed
for module 8. Ejector based shutdown enabled
2016 Mar 3 04:44:10 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 8 detected (Serial number
ABC3456D7E8) Module-Type 48x1/10G-T 4x40G Ethernet Module Model N9K-X9564TX
2016 Mar 3 04:44:10 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 8 powered up (Serial number
ABC3456D7E8)
2016 Mar 3 04:44:56 switch%$ VDC-1 %$ %USBHSD-STANDBY-2-MOUNT: logflash: online
2016 Mar 3 04:47:31 switch%$ VDC-1 %$ %ASCII-CFG-2-CONF_CONTROL: System ready
2016 Mar 3 04:47:51 switch%$ VDC-1 %$ %VMAN-2-ACTIVATION_STATE: Successfully activated virtual
service 'guestshell+'
2016 Mar 3 04:47:51 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.

```

User Access Verification

```

switch# show version
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright (C) 2002-2016, Cisco and/or its affiliates.
All rights reserved.
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under their own
licenses, such as open source. This software is provided "as is," and unless
otherwise stated, there is no warranty, express or implied, including but not
limited to warranties of merchantability and fitness for a particular purpose.
Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or
GNU General Public License (GPL) version 3.0 or the GNU
Lesser General Public License (LGPL) Version 2.1 or
Lesser General Public License (LGPL) Version 2.0.
A copy of each such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://opensource.org/licenses/gpl-3.0.html and
http://www.opensource.org/licenses/lgpl-2.1.php and
http://www.gnu.org/licenses/old-licenses/library.txt.

```

Software

```

BIOS: version 08.26
NXOS: version 7.0(3)I7(1)
BIOS compile time: 06/12/2016
NXOS image file is: bootflash:///nxos.7.0.3.I7.1.bin
NXOS compile time: 2/8/2016 20:00:00 [02/09/2016 05:18:17]

```

Hardware

```

cisco Nexus9000 C9516 (16 Slot) Chassis ("Supervisor Module")
Intel(R) Xeon(R) CPU E5-2403 0 @ 1.80GHz with 16401664 kB of memory.
Processor Board ID SAL1745FTPW

```

```

Device name: switch
bootflash: 20971520 kB
Kernel uptime is 0 day(s), 0 hour(s), 8 minute(s), 13 second(s)

Last reset at 235176 usecs after Thu Mar 3 04:40:48 2016

Reason: Reset due to upgrade
System version: 7.0(3)I1(2)
Service:

plugin
Core Plugin, Ethernet Plugin

Active Package(s):
switch#

```

Upgrade the NX-OS software

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

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

Step 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
```

Step 3 Log in and choose the software image file for your device from the [Software Download](#) website, and download it to a file server.

Step 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
3bf6a771aa4a192a8e1383e348b26bb483356a9774d74ba39ecbf7718248483b3391942d8103de8104deea8fda212266e70bd736220cff34943bd8e359432975
```

- Step 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.

- Step 6** Save the running configuration to the startup configuration.

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

- Step 7** Upgrade the NX-OS software using the **install all nxos bootflash:filename [no-reload | non-disruptive | non-interruptive | 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.
- **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
```

Step 8 (Optional) Post upgrade actions.

- Display the entire upgrade process.

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

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

```
switch# show version
```

- 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 Guide](#) and [Cisco Nexus 9000 and 3000 Series NX-OS Smart Licensing Using Policy User Guide](#).

In-Service Software Upgrade

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 19](#) 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 Nexus 9000 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 Nexus 9300 platform switches are the NX-OS switches with single supervisors. Performing ISSU on the Nexus 9000 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 Nexus 9000 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.

To enable booting in the enhanced ISSU (LXC) mode, use the **[no] boot mode lxc** command. This command is executed in the configuration mode. Here is a sample configuration for your reference.

```
switch(config)# boot mode lxc
Using LXC boot mode
Please save the configuration and reload system to switch into the LXC mode.
switch(config)# copy r s
[#####] 100%
Copy complete.
```



Note Reload the switch first, when enabling enhanced ISSU for the first time.

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 Nexus 9000 Series NX-OS High Availability and Redundancy Guide](#).

NX-OS upgrade history

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

```

Prerequisites for NX-OS software downgrade

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

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 not 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

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.



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

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

Step 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
```

Step 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.

- b) Check for any hardware incompatibilities.

Example:

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

Step 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
```

Step 5 Save the running configuration to the startup configuration.

Example:

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

Step 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.

Step 7 (Optional) Display the entire downgrade process.

Example:

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

Step 8 (Optional) Log in and verify that the device is running the required software version.

Example:

```
switch# show version
```



CHAPTER 3

Transceiver Firmware

A transceiver firmware in Nexus 9000 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 upgrades, on page 51](#)
- [Firmware upgrade using NX-OS commands, on page 52](#)
- [Parallel firmware download, on page 72](#)

Transceiver firmware upgrades

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 52](#).

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 72](#).

Firmware upgrade using NX-OS commands

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

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

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:

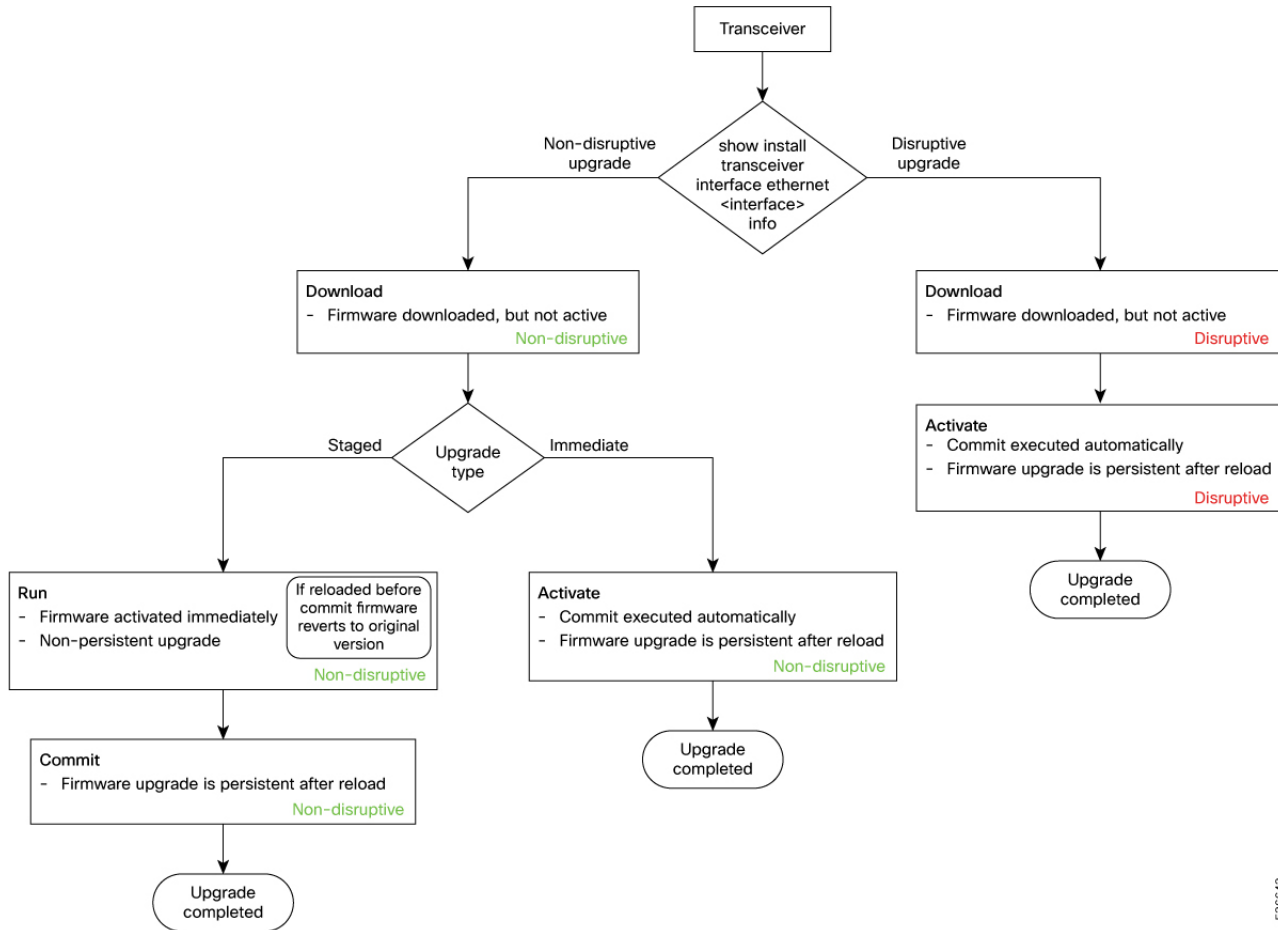
- **Download:** Transfers the new firmware image from the source (such as the switch bootflash) to the transceiver's memory. The download can be disruptive or non-disruptive.
- **Activate:** Switches the transceiver from the current firmware to the newly downloaded version immediately, with no warning. 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

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	<ol style="list-style-type: none"> 1. Download 2. Activate
Non-disruptive	Staged	<ol style="list-style-type: none"> 1. Download 2. Run 3. Commit
	Immediate	<ol style="list-style-type: none"> 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

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.

Upgrade firmware using NX-OS commands

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

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 64.

Procedure

Step 1 Get the firmware file, copy it on to the switch bootflash, and check the information of the firmware.

- 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
```

- 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.

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.

Step 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.

- Step 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

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 67](#).

Procedure

- Step 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.

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.

Step 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.

- Step 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

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 70](#).

Procedure

- Step 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.

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.

Step 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.

Step 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.

Step 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

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

Sample disruptive immediate upgrade with both banks 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 56](#).

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

Step 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.

Step 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.
```

Step 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
```

Step 4 Enter the **info** command.

Example:

Sample disruptive immediate upgrade with one bank invalid

```

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).

Step 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

```

Step 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

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 56](#).

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

Step 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.

Step 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
```

Step 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
```

Sample disruptive immediate upgrade with one bank invalid

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

Step 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.

Step 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
```

Step 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

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 58](#).

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

Step 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.

Step 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.
```

Step 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
```

Step 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.

Step 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
```

Step 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.

Step 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

```

Step 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

```

Step 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

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 60](#).

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

Step 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.

Step 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.
```

Step 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
```

Step 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.

Step 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
```

Step 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

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

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

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

Step 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.

Step 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.

Step 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 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

Step 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.

Step 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.

Step 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
```

Step 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.

Step 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.



CHAPTER 4

Optionality in NX-OS Software

- [Optionality in NX-OS software, on page 77](#)
- [Modular packages, on page 78](#)
- [NX-OS image boot modes, on page 79](#)
- [Red Hat Package Managers, on page 80](#)
- [Dandified YUM commands, on page 93](#)
- [Configure an FTP server and set up a local FTP YUM repository, on page 108](#)
- [Create user roles for install operation, on page 111](#)

Optionality in NX-OS software

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:

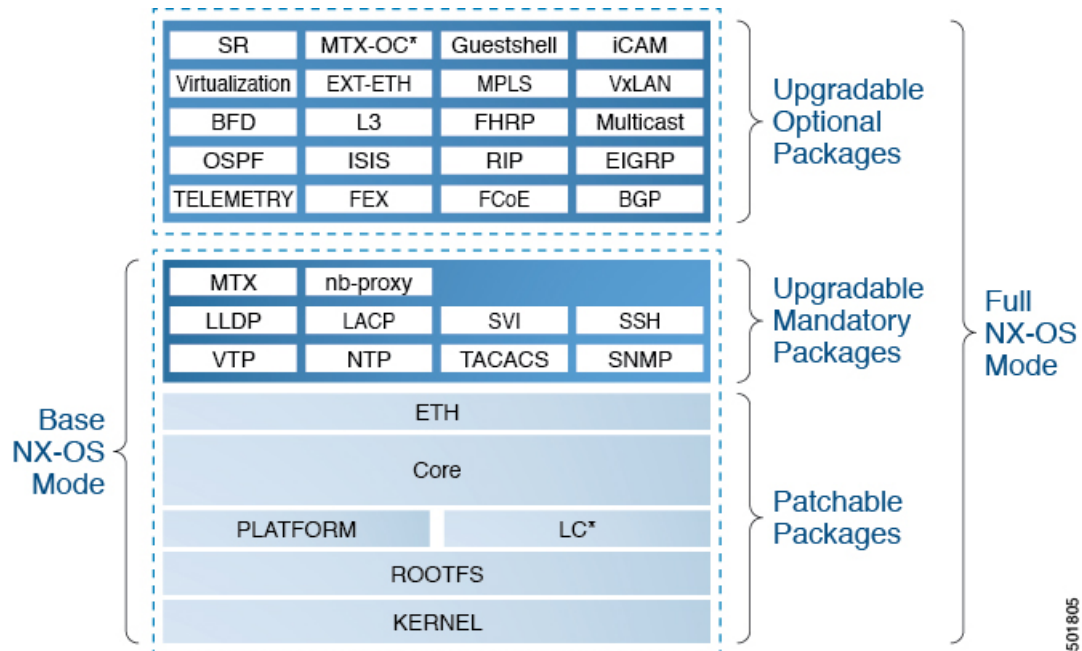
- Leaner NX-OS software
- Asynchronous delivery of the features and the fixes—provide quick fixes that are independent of the releases, including new features
- Reduced footprint of binaries and libraries at run time

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

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

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

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

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 2: RPM naming convention

RPM Naming Convention	Description
Example: fex-2.0.0.0-9.2.1.lib32_n9000.rpm	
fex	Indicates the name of the component.
2	Indicates that the RPM is not backward compatible. Configuration loss takes place during an upgrade.

RPM Naming Convention Example: fex-2.0.0.0-9.2.1.lib32_n9000.rpm	Description
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.
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 3: 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

Package Name	Associated Features
Ext-Eth	<ul style="list-style-type: none"> • feature openflow • feature evb • feature imp • feature netflow • feature sla_sender • feature sla_responder • feature sla_twamp-server • feature sflow
EXT_ETH_LOWMEM	<ul style="list-style-type: none"> • feature evb • feature netflow
FCoE	<ul style="list-style-type: none"> • feature-set fcoe • feature-set fcoe-npv
FEX	feature-set fex
FHRP	<ul style="list-style-type: none"> • feature hsrp • feature vrrpv3
HW TELEMETRY	feature hw telemetry
iCAM	feature icam
ISIS	feature isis
MPLS	<ul style="list-style-type: none"> • feature mpls segment-routing • feature mpls evpn
Multicast	<ul style="list-style-type: none"> • feature pim • feature pim6 • feature msdp • feature ngmvpn
NIA	NA
NXSDK	NA

Package Name	Associated Features
OSPF	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • feature nv overlay • feature fabric forwarding

Guidelines for NX-OS feature RPM installation

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 4: 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.

Repository Name	Repository Path	Description
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

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 5: Reference for install commands for the feature RPM operations

Command	Description
install reset	<p>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.</p> <p>The install reset command also performs write erase operation. The following message is displayed at the prompt:</p> <pre>switch(config)# install reset</pre> <hr/> <pre>WARNING!!This operation will remove all patches, upgraded packages, persisted etc configs, third party packages installed, startup configuration(write erase) and reload the switch with default packages.</pre> <hr/> <pre>Do you want to proceed with reset operation? (y/n)? [n]</pre>
install reset nxos base	<p>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.</p>
install reset nxos full	<p>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).</p>
install add <>	<p>Adds an RPM file to the respective repository and updates the repository (patch/feature/third-party).</p>
install activate <rpm name>	<p>Installs an RPM that is present in the repository.</p>
install commit <rpm name>	<p>Used for the patch RPMs. Makes the patch persist during the reload.</p>

Command	Description
install deactivate <i><rpm name></i>	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 <i><rpm name></i>	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
<p>[no] system software allow third-party</p>	<p>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.</p> <p>The following command shows the error message when you activate third-party RPM without applying the third-party configuration:</p> <pre>switch(config)# install activate pbwMonitor-1.0-1.5.0.x86_64.rpm</pre> <p>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</p> <p>The following command shows activating third-party RPM installations after applying the configuration:</p> <pre>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</pre> <pre>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</pre> <p>The following command shows disabling the third-party configuration:</p> <pre>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</pre>



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.

Procedure

Step 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.

Step 2 Use one of the two steps to verify whether the RPM file is a signed or non-signed file.

- Verify that the package is a signed file using the **install verify package <package-name>** command.
- 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

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-udhcp.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.

Procedure

Step 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

```

Step 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.e15.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.

Procedure

Step 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

```

Step 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

```

Step 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

```

Step 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

The commands for both install and upgrade RPMs are the same. Perform this procedure to upgrade the RPMs:

Procedure

Step 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
```

Step 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.e15.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

The downgrade procedure needs a special command attribute. Downgrade the RPMs using the one-step procedure.

Procedure

Step 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
```

Step 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.e15.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.

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.

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

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:



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

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                mppls-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.

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

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
```

```
localdb                | 1.1 kB    00:00 ...
patching                | 951 B     00:00 ...
thirdparty              | 951 B     00:00 ...
                        | 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. 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

```

```

localdb                               | 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	Repository	Arch	Version	Size
Installing:				
vxlan		lib32_n9000	2.0.1.0-9.2.1	6.4 M
/vxlan-2.0.1.0-9.2.1.lib32_n9000				

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
```

```

localdb                | 1.1 kB    00:00 ...
patching               |  951 B    00:00 ...
thirdparty             |  951 B    00:00 ...
                       |  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 Version	Arch Size	Repository
Updating: vxlan 2.0.1.0-9.2.1	lib32_n9000 6.4 M	/vxlan-2.0.1.0-9.2.1.lib32_n9000

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

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

```

```

localdb                | 1.1 kB    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	Repository	Arch	Version	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

```

```

localdb           | 1.1 kB    00:00 ...
patching          | 951 B     00:00 ...
thirdparty       | 951 B     00:00 ...
                  | 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	Repository
Version	Size	
Updating:		
vxlan	lib32_n9000	
2.0.1.0-9.2.1	6.4 M	/vxlan-2.0.1.0-9.2.1.lib32_n9000

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).

```

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

```

```

localdb          | 1.1 kB    00:00 ...
localdb/primary  | 951 B     00:00 ...
localdb          | 1.3 kB    00:00 ...

                               2/2
patching

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	Version	Size	Arch	Repository
Downgrading:				
vxlan	2.0.0.0-9.2.1	1.6 M	lib32_n9000	groups-repo
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. 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
=====
Remove 1 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 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 DNF supports 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, 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

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. 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

Update any existing installed group packages using the **dnf groupupdate** command.

```
bash-4.4# dnf groupupdate 13
```

```
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	Repository	Size	Version
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

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      Repository      Arch      Size      Version
=====
Removing:
bfd          @System        lib32_64_n9000  2.3 M    2.0.0.0-10.4.3

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

localdb          | 1.1 kB    00:00 ...
patching         | 951 B     00:00 ...
thirdparty      | 951 B     00:00 ...
repo id         | 951 B     00:00 ...
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:

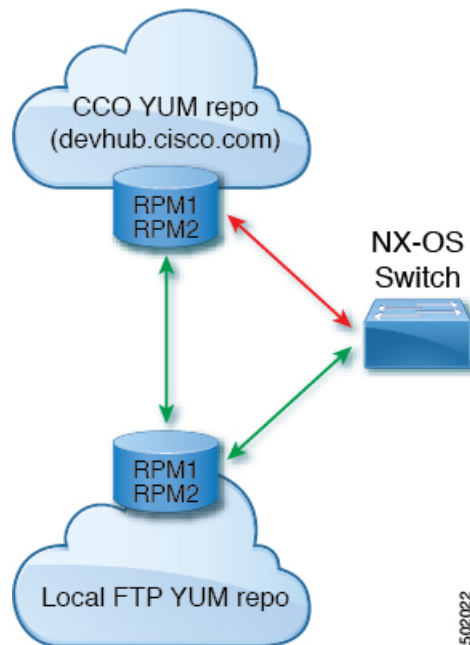
Table 6: Patching command reference

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: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • 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
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

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

- Step 1** Install vsftpd, an FTP server, using the **dnfinstall vsftpd** command.
- Step 2** Start the FTP server using the **systemctl start vsftpd** command.
- Step 3** Check the status of the FTP server using the **systemctl status vsftpd** command.
- Step 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.
- Step 5** Add the FTP service using the **firewall-cmd --zone=public --permanent --add-service=ftp** command.

Step 6 Reload the server using the **firewall-cmd --reload** command.

Step 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

Step 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
```

Step 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
```

Step 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
```

Step 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
```

Step 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.

Step 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

Step 1 Log in as a sudo user using the **run bash sudo su** command.

Step 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.

Step 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
```

Step 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
```

Step 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
```

Step 6 To use the Bash shell prompt, run the `ip netns exec management bash` command.

Step 7 Check the reachability of the newly created repository using the `dnf repolist` 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
```

Step 8 List the available packages in the new repository using the `dnf list available` command.

Create user roles for install operation

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](#).



CHAPTER 5

Convert from NX-OS to ACI boot mode and from ACI boot mode back to NX-OS

This chapter describes how to convert a Nexus 9000 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 Nexus 9000 Series switch from NX-OS to ACI boot mode, on page 113](#)
- [Convert a replacement standby supervisor to ACI boot mode, on page 115](#)
- [Convert a Nexus 9000 Series switch back to NX-OS, on page 116](#)

Convert a Nexus 9000 Series switch from NX-OS to ACI boot mode

You can convert any Nexus 9000 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(2x) 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

Step 1 Verify that the switch is running the latest release.

Example:

```
switch(config)# show version
```

NX-OS file names begin with `nxos`.

Step 2 Copy the ACI image from the APIC:

- Set the IP address on the `mgmt0` interface of the switch to allow connectivity between this interface and the APIC.
- Enable Secure Copy Protocol (SCP) services on the switch.

Example:

```
switch(config)# feature scp-server
```

- 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
```

- For dual supervisor systems, copy the ACI image to the standby supervisor module.

Example:

```
switch(config)# copy bootflash:aci-image bootflash://sup-standby/
```

Step 3 Follow these steps to boot to the ACI image:

- Configure the switch to not boot from NX-OS.

Example:

```
switch(config)# no boot nxos
```

- 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
```

Step 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

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

Step 1 Reload the switch.

Example:

```
switch# reload
```

Step 2 Enter a break sequence (Ctrl-C or Ctrl-]) during the initial boot sequence to access the loader> prompt.

Example:

```
ctrl-c
loader>
```

Step 3 Plug the USB drive containing the ACI image into the standby supervisor USB slot.

Step 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.

Step 5 Log in to the switch as an administrator.

```
Login: admin
```

Step 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 Nexus 9000 Series switch back to NX-OS

You can convert a Nexus 9000 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

Step 1 Reload the switch.

Example:

```
switch# reload
```

Step 2 During the initial boot sequence, enter a break sequence (Ctrl-C or Ctrl-]) to access the loader> prompt.

Example:

```
Ctrl-C  
loader>
```

Step 3 Configure the boot process to stop at the boot prompt.

Example:

```
loader> cmdline recoverymode=1
```

Step 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 Nexus 9000 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.

Step 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
```

Step 6 Complete the upload of the NX-OS image file.

Example:

```
switch(boot)# load-nxos
```

Note

For some Nexus 9000 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** *address subnet 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 the following 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.

- Step 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-name bootflash:
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-name bootflash:
switch# configure terminal
switch(config)# boot nxos bootflash:nxos-image-name
switch(config)# copy running-config startup-config
switch(config)# end
```

- Step 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.

- Step 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

If you have a Nexus 9000 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.

Load NX-OS image into bootflash using SCP on the ACI shell



CHAPTER 6

Migrate Switches in a vPC Topology

- Upgrade process for a vPC topology on the primary switch, on page 121
- Upgrade process for a vPC topology on the secondary switch, on page 122
- vPC forklift upgrade, on page 123
- vPC upgrade and downgrade procedure for Nexus 9000 -R series switches, on page 123

Upgrade process for a vPC topology on the primary switch



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 in 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

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

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 Nexus 9000 -R series switches

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

Step 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.

```

Step 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 -

```

Step 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
```

Step 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
```

Step 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.
