Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide, Release 9.3(x)

First Published: 2021-01-18

Last Modified: 2021-09-23

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

CONTENTS

Preface

This preface includes the following sections:

Audience

This publication is for network administrators who install, configure, and maintain Cisco Nexus switches.

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which you supply the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments that are separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments that are separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.

Convention	Description
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string includes the quotation marks.

Examples use the following conventions:

Convention	Description
`screen font`	Terminal sessions and information the switch displays are in screen font.
`boldface screen font`	Information that you must enter is in boldface screen font.
italic screen font	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Related Documentation for Cisco Nexus 9000 Series Switches

The entire Cisco Nexus 9000 Series switch documentation set is available at the following URL:

http://www.cisco.com/en/US/products/ps13386/tsd_products_support_series_home.html

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus9k-docfeedback@cisco.com. We appreciate your feedback.

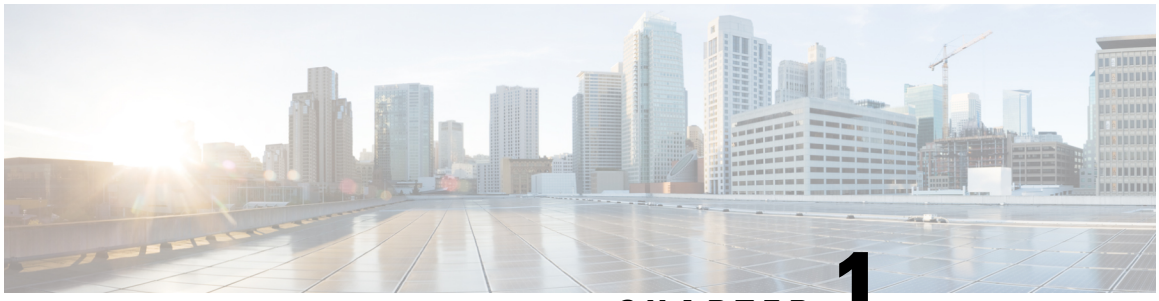
Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

Preface



CHAPTER 1

New and Changed Information

This chapter provides release-specific information for each new and changed feature in the *Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide, Release 9.3(x)*.

- New and Changed Information, on page 1

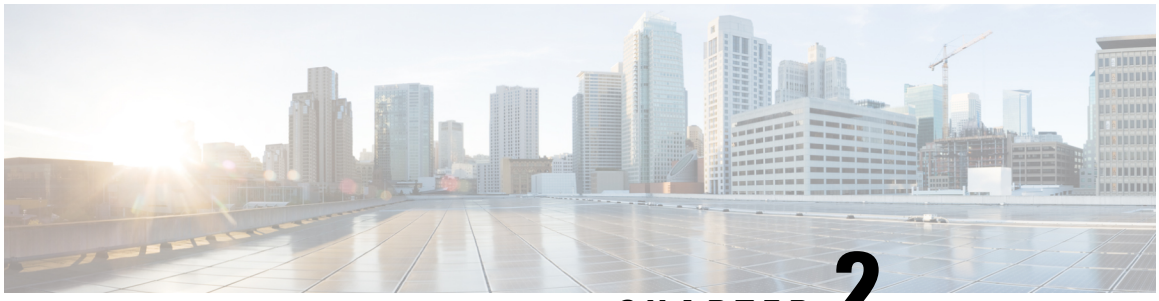
New and Changed Information

This table summarizes the new and changed features for the *Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide, Release 9.3(x)* and tells you where they are documented.

Table 1: New and Changed Features for Cisco NX-OS Release 9.3(x)

Feature	Description	Changed in Release	Where Documented
NX-OS Upgrade History	Support for maintaining the software upgrade history across upgrades.	9.3(5)	Cisco NX-OS Upgrade History, on page 39
ISSU Support for uRPF	Support for standard ISSU on Cisco Nexus 9300-EX/FX/FX2/GX platform switches configured with uRPF.	9.3(5)	Cisco NX-OS Software Upgrade Guidelines, on page 8
Cisco NX-OS Software Image Compaction	Support for compacting the software image during copy operations.	9.3(5)	Upgrading the Cisco NX-OS Software, on page 33 Optionality in Cisco NX-OS Software, on page 41
Standard and enhanced ISSU support for Cisco nexus 9300-FX3 platform switches	Introduced standard and enhanced ISSU support for Cisco Nexus 93180YC-FX3S switches.	9.3(5)	Cisco NX-OS Software Upgrade Guidelines, on page 8
Standard and enhanced ISSU support for Cisco Nexus 9300-GX platform switches	Introduced standard and enhanced ISSU support for Cisco Nexus 9300-GX platform switches.	9.3(5)	Cisco NX-OS Software Upgrade Guidelines, on page 8

Feature	Description	Changed in Release	Where Documented
Standard ISSU support for Cisco Nexus 9300-FX platform switches	Introduced standard ISSU support for Cisco Nexus 9300-FX Series platform switches.	9.3(3)	Cisco NX-OS Software Upgrade Guidelines, on page 8
Enhanced ISSU	Added support from Cisco NX-OS Release 9.3(1) to later releases, even in cases of kernel incompatibility.	9.3(1)	Upgrading or Downgrading the Cisco Nexus 9000 Series NX-OS Software, on page 3



CHAPTER 2

Upgrading or Downgrading the Cisco Nexus 9000 Series NX-OS Software

This chapter describes how to upgrade or downgrade the Cisco NX-OS software. It contains the following sections:

About the Software Image

Each device is shipped with the Cisco NX-OS software preinstalled. The Cisco NX-OS software consists of one NX-OS software image. The image filename begins with "nxos" (for example, nxos.9.3.1.bin). Only this image is required to load the Cisco NX-OS operating system.

The Cisco Nexus 9000 Series switches and the Cisco Nexus 3132C-Z, 3132Q-V, 3164Q, 3232C, 3264C-E, 3264Q, 31108PC-V, 31108TC-V, 31128PQ, and 34180YC switches support disruptive software upgrades and downgrades by default.

Note 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 from Cisco.com and generally include the ID number of the resolved defect in the filename (for example, n9000-dk9.3.1.**CSCab00001**.gbin). For more information on SMUs, see the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide.

Note Cisco also provides electronic programmable logic device (EPLD) image upgrades to enhance hardware functionality or to resolve known hardware issues. The EPLD image upgrades are independent from the Cisco NX-OS software upgrades. For more information on EPLD images and the upgrade process, see the Cisco Nexus 9000 Series FPGA/EPLD Upgrade Release Notes.

About ISSU

An in-service software upgrade (ISSU) allows you to upgrade the device software while the switch continues to forward traffic. ISSU reduces or eliminates the downtime typically caused by software upgrades. You can perform an ISSU, also known as a nondisruptive upgrade, for some switches. (See the Cisco NX-OS Software Upgrade Guidelines 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 nondisruptive option helps ensure a nondisruptive upgrade. The guest shell is disabled during the ISSU process and it is later reactivated after the upgrade.

Enhanced ISSUs are supported for some Cisco Nexus 9000 Series switches and the Cisco Nexus 3164Q, 31128PQ, 3132Q-V, 31108PC-V, and 31108TC-V switches.

The following ISSU scenarios are supported:

- Performing standard ISSU on Top-of-Rack (ToR) switches with a single supervisor
- Performing standard ISSU on End-of-Row (EoR) switches with two supervisors
- Performing enhanced ISSU on Top-of-Rack (ToR) switches with a single supervisor

Performing Standard ISSU on Top-of-Rack (ToR) Switches with a Single Supervisor

The ToR Cisco Nexus 9300 platform switches and Cisco Nexus 3100 Series switches are the standalone switches with single supervisors. Performing ISSU on the Cisco Nexus 9000 and 3100 Series switches causes the supervisor CPU to reset and to load the new software version. After the CPU loads the updated version of the Cisco 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.

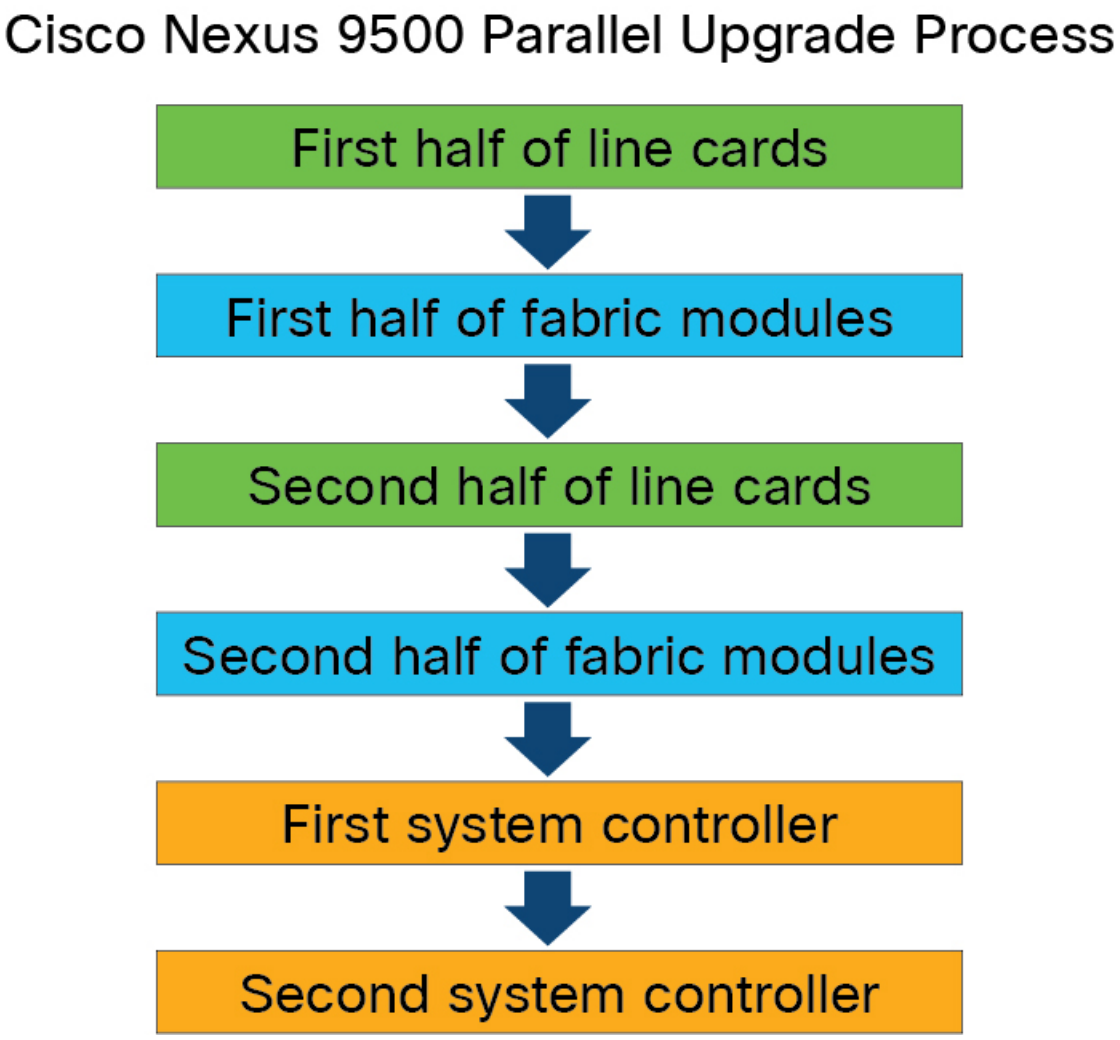
The data plane traffic is not disrupted during the ISSU process. In other words, the data plane forwards the packets while the control plane is being upgraded, any servers that are connected to the Cisco Nexus 9000 and 3100 Series switches do not see any traffic disruption. The control plane downtime during the ISSU process is approximately less than 120 seconds.

Performing Standard ISSU on End-of-Row (EoR) Switches with Two Supervisors

Cisco Nexus 9500 Series switches are the modular EoR switches that require two supervisors for ISSU. The minimum configuration required is two system controllers and two fabric modules.

Cisco Nexus 9500 Series switches support parallel upgrade as the default method. The parallel method upgrades the modules in the batches (as outlined in the following illustration) instead of upgrading the modules one after the other.

Figure 1: Parallel Upgrade Process for Cisco Nexus 9500 Series Switches



The steps for the parallel upgrade process on Cisco Nexus 9500 Series switches are:

- First the supervisors are upgraded (This procedure requires a switchover). Then the line cards, the fabric modules, the system controllers, and the FEX are upgraded.
- After the switchover is performed in a parallel upgrade, the secondary supervisor takes over. The installer determines the current line cards and the fabric modules.
- The installer then divides the components into the buckets. It places the first half of the line cards in the first bucket, the first half of the fabric modules in the second bucket, the second half of line cards in the

third bucket, the second half of the fabric modules in the fourth bucket, the first system controller in the fifth bucket, and the second system controller in the sixth bucket.

- Each bucket is upgraded successfully before an upgrade process starts for the next bucket.
- The console displays the modules with the bucket assignments and the status of the upgrade.

The user also has the option to choose a serial upgrade using the CLI.

While performing standard ISSU for the modular switches, the data plane traffic is not disrupted. The control plane downtime is approximately less than 6 Seconds.

Note The minimum requirement for a modular Cisco Nexus 9000 Series switch undergoing ISSU is two supervisors, two system controllers, and two fabric modules. The Cisco Nexus 9400 line cards can have a partially connected fabric module. In this case, if only two fabric modules are used with the Cisco Nexus 9400 line cards, the fabric modules should not be in slots 21 and 25. They should be in slots 22, 23, 24, or 26. This allows for the system to maintain high availability during ISSU.

Performing Enhanced ISSU on Top-of-Rack (ToR) Switches with a Single Supervisor

Note Enhanced ISSU to Cisco NX-OS Release 9.3(1) is not supported as there are kernel fixes that cannot take effect without reloading the underlying kernel. Enhanced ISSU from Cisco NX-OS Release 9.3(1) to later releases is supported, even in cases of kernel incompatibility.

The Cisco 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 config mode. See the following sample configuration for more information:

```
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 When you are enabling enhanced ISSU for the first time, you have to reload the switch first.

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 nondisruptive downgrades, are not supported.

For information on ISSU and high availability, see the Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide.

Prerequisites for Upgrading the Cisco NX-OS Software

Upgrading the Cisco NX-OS software has the following prerequisites:

- For ISSU compatibility for all releases, see the Cisco NX-OS ISSU Support Matrix.
- Ensure that everyone who has access to the device or the network is not configuring the device or the network during this time. You cannot configure a device during an upgrade. Use the **show configuration session summary** command to verify that you have no active configuration sessions.
- Save, commit, or discard any active configuration sessions before upgrading or downgrading the Cisco NX-OS software image on your device. On a device with dual supervisors, the active supervisor module cannot switch over to the standby supervisor module during the Cisco NX-OS software upgrade if you have an active configuration session.
- To transfer NX-OS software images to the Nexus switch through a file transfer protocol (such as TFTP, FTP, SFTP, SCP, etc.), 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 subnets, ensure that the Nexus switch and the remote file server are on the same subnetwork. 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. An example of using the **ping** command to verify connectivity to a remote file server 192.0.2.100 is shown below:

```
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 more information on configuration sessions, see the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide* specific to your release.

Prerequisites for Downgrading the Cisco NX-OS Software

Downgrading the Cisco NX-OS software has the following prerequisites:

- Before you downgrade from a Cisco NX-OS release that supports the Control Plane Policing (CoPP) feature to an earlier Cisco NX-OS release that does not support the CoPP feature, you should verify compatibility using the **show incompatibility nxos bootflash:***filename* command. If an incompatibility exists, disable any features that are incompatible with the downgrade image before downgrading the software.

Cisco NX-OS Software Upgrade Guidelines

Before attempting to upgrade to any software image, follow these guidelines:

- When upgrading from Cisco NX-OS Release 9.3(3) to Cisco NX-OS Release 9.3(6), if you do not retain configurations of the TRM enabled VRFs from Cisco 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** CLI, when feature **ngmvpn** is enabled, will not happen. You must enable the CLI manually for each TRM enabled VRF. For the configuration instructions, see Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide, Release 9.3(x).
- 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.
- When you upgrade a Cisco Nexus 9000 device to Cisco NX-OS Release 9.3(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.
- Enhanced ISSU: Non-disruptive enhanced ISSU to Cisco NX-OS Release 9.3(x) is not supported as there are kernel fixes that cannot take effect without reloading the underlying kernel. The upgrade will be disruptive. However, a non-disruptive enhanced ISSU from Cisco NX-OS Release 9.3(x) to later releases is supported in fallback mode only, even in cases of kernel incompatibility.
- When upgrading from Cisco NX-OS Release 9.2(2) or earlier releases to Cisco NX-OS Release 9.3(x), you need to make sure that ingress RACL TCAM region is not more than 50% full. Otherwise, the atomic update feature will be enabled after the upgrade and interfaces with RACLs that exceed 50% of TCAM allocation will remain down.
- When upgrading from Cisco NX-OS Release 9.2(4) or earlier releases to Cisco 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 Cisco NX-OS Release 9.3(1) or 9.3(2) to Cisco 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, Release 9.3(x).
- Before upgrading from Cisco NX-OS Release 7.0(3)I7(5) to Cisco NX-OS Release 9.3(5), make sure that you configure TCAM region Egress Layer3/VLAN QOS [egr-l3-vlan-qos].

- Beginning with Cisco NX-OS Release 9.3(5), ISSU is supported on FC/FCoE switch mode on N9K-C93180YC-FX. For more information about the FC/FCoE switch mode and supported hardware, see Cisco Nexus 9000 Series NX-OS SAN Switching Configuration Guide, Release 9.3(x)
- Beginning with Cisco NX-OS Release 9.3(5), ISSU is supported with FC/FCoE NPV mode on N9K-C93180YC-FX and N9K-C93360YC-FX2. For more information about the FC/FCoE NPV mode and supported hardware, see Cisco Nexus 9000 Series NX-OS FC-NPV and FCoE-NPV Configuration Guide
- Software image compaction is only supported on Cisco Nexus 9300-series platform switches.
- The compressed image of Cisco Nexus 3000-series is hardware dependent and can only be used on the same device that it got compressed or downloaded from CCO. Do not use the Nexus 3000-series compressed image on Nexus 9000-series
- The following limitation applies to software upgrades from 7.0(3)I5 to 9.3(x) or 9.2(3) to 9.3(x):

  If you have the same NetFlow configuration in both VLAN and SVI, you must remove the NetFlow flow monitor from the VLAN configuration prior to the upgrade. Once upgraded, reconfigure NetFlow by creating a new flow monitor and adding it to the VLAN configuration. Failure to perform these steps results in error messages and the inability to modify the VLAN NetFlow configuration in the upgraded software.
- When upgrading from Cisco NX-OS Releases 7.0(3)I4(8), 7.0(3)I5(3), and 7.0(3)I6(1) to Cisco NX-OS Release 9.3(x) results in a disruptive upgrade. If syncing images to standby SUP failed during the disruptive upgrade from Cisco NX-OS Releases 7.0(3)I4(8), 7.0(3)I5(3,) or 7.0(3)I6(1) to 9.3(x), you should manually copy the image to the standby SUP and perform the disruptive upgrade.
- When upgrading to Cisco NX-OS Release to 9.3(x) from any release prior to 7.0(3)I2(3) an intermediate upgrade to 7.0(3)I4(x), 7.0(3)I5(x), 7.0(3)I6(x), or 7.0(3)I7(x) is required. We recommend using Cisco NX-OS Release 7.0(3)I4(8) or 7.0(3)I7(4) as the interim release to aid in a smooth migration.
- When upgrading from Cisco NX-OS Release 7.0(3)I6(1) or 7.0(3)I7(1) to Cisco NX-OS Release 9.3(x), if the Cisco 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 Cisco Nexus 9000 Series switches are upgraded, you can re-enable the command.
- If you are upgrading from Cisco NX-OS Release 7.0(3)I5(2) to Cisco NX-OS Release 9.3(x) using the install all command, BIOS will not be upgraded due to CSCve24965. When the upgrade to Cisco NX-OS Release 9.3(x) is complete, use the install all command again to complete the BIOS upgrade, if applicable.
- An upgrade that is performed via the install all command for Cisco NX-OS Release 7.0(3)I2(2b) to Release 9.3(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 Cisco 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.
- When upgrading to Cisco NX-OS Release 9.3(x) from 7.0(3)I2(x) or before and running EVPN VXLAN configuration, an intermediate upgrade to 7.0(3)I4(x) or 7.0(3)I5(x) or 7.0(3)I6(x) is required.
- Before enabling the FHS on the interface, we recommend that you carve the ifacl TCAM region on Cisco 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 Cisco NX-OS Release 9.3(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 Cisco 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 Cisco NX-OS Release 9.3(x). Uploading the system creates the required match qualifiers for the FHS TCAM region, ing-redirect.
- Upgrading from Cisco 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.
- For any prior release version upgrading to Cisco 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 Cisco 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*
- 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.
- During an ISSU, there is a drop for all traffic to and from 100-Mb ports 65–66 on the Cisco Nexus 92304QC switch.
- The install all command is the recommended method for software upgrades and downgrades 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.
- Upgrading from Cisco NX-OS Release 7.0(3)I1(2), Release 7.0(3)I1(3), or Release 7.0(3)I1(3a) requires installing a patch for Cisco Nexus 9500 platform switches only. For more information on the upgrade patch, see Patch Upgrade Instructions.

- When upgrading to Cisco NX-OS Release 9.3(x), Guest Shell automatically upgrades from 1.0 to 2.0. In the process, the contents of the guest shell 1.0 root filesystem are lost. To keep from losing important content, copy any needed files to /bootflash or an off-box location before upgrading to Cisco NX-OS Release 9.3(x).
- An ISSU can be performed only from a Cisco NX-OS Release 7.0(3)I4(1) to a later image.
- While performing an ISSU, VRRP and VRRPv3 displays the following 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
```

- Guest Shell is disabled during an ISSU and reactivated after the upgrade. Any application running in the Guest Shell is affected.
- If you have ITD probes configured, you must disable the ITD service (using the shutdown command) before upgrading to Cisco NX-OS Release 9.3(x). After the upgrade, enter the **feature sla sender** command to enable IP SLA for ITD probes and then the no shutdown command to re-enable the ITD service. (If you upgrade without shutting down the service, you can enter the feature sla sender command after the upgrade.)
- 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. See the Hardware Installation Guide for your specific chassis.
- Perform the installation on the active supervisor module, not the standby supervisor module.
- 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 Cisco Nexus 9500 platform switches with -R line cards, you must save the configuration and reload the device to upgrade from Cisco NX-OS Release 7.0(3)F3(5) to 9.3(1). To upgrade from Cisco 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 or SHA256 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 SHA256 checksum of the software image, run the **show file bootflash:**<*IMAGE-NAME*> **sha256sum** command and compare the resulting value to the published SHA256 checksum for the software image on Cisco's Software Download website.

- When upgrading from Cisco Nexus 94xx, 95xx, and 96xx line cards to Cisco Nexus 9732C-EX line cards and their fabric modules, upgrade the Cisco 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.
- If you upgrade from a Cisco NX-OS release that supports the CoPP feature to a Cisco 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, see the "Configuring Control Plane Policing" chapter in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide, Release 9.3(x).
- 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, Release 9.3(x).
- When you upgrade from an earlier release to a Cisco 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, see the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide, Release 9.3(x).
- By default, the software upgrade process is disruptive.
- OpenFlow and LACP fast timer rate configurations are not supported for ISSU.
- Guest Shell is disabled during an ISSU and reactivated after the upgrade.
- ISSU supports only default hold timers for BGP peers.
- During an ISSU on a Cisco Nexus 3164Q, 31128PQ, or 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.
- Make sure that both vPC peers are in the same mode (regular mode or enhanced mode) before performing a nondisruptive 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 Cisco 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.
- In general, ISSUs are supported for the following:
  - From a major release to any associated maintenance release.
  - From the last two maintenance releases to the next two major releases.
  - From an earlier maintenance release to the next two major releases.

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

- After performing ISSU on Cisco Nexus 9300 platform switches and the Cisco Nexus 3164Q switches, you may see the MTS_OPC_CLISH message on the vPC peers. MTS_OPC_CLISH is the last MTS code that is sent from the back-end component to the VSH to specify the end of the show command output.

  If the user executes a show command that produces more output and keeps the session on for more than 3 minutes, the following warning message may be displayed on the console. As a workaround, you can set the terminal length as 0 using the **terminal length 0** command or the **show <command> | no-more** option.

```
--More--2018 Jun 5 19:11:21 Th-agg1 %$ VDC-1 %$ Jun 5 19:11:20 %KERN-2-SYSTEM_MSG:
[12633.219113]
App vsh.bin on slot 1 vdc 1 SUP sap 64098(cli_api queue) did not drop MTS_OPC_CLISH
with
msg_id 0x675ecf from sender sap 64132(NULL) in 180 sec, contact app owner - kernel


(config)# show ip mroute detail
IP Multicast Routing Table for VRF "default"

Total number of routes: 4801
Total number of (*,G) routes: 2400
Total number of (S,G) routes: 2400
Total number of (*,G-prefix) routes: 1

(*, 225.0.0.1/32), uptime: 00:09:32, igmp(1) pim(0) ip(0)
  RPF-Source: 10.10.10.3 [11/110]
  Data Created: No
  VPC Flags
    RPF-Source Forwarder
  Stats: 15/720 [Packets/Bytes], 0.000   bps
  Stats: Inactive Flow
  Incoming interface: Ethernet1/1, RPF nbr: 12.0.0.2
  LISP dest context id: 0  Outgoing interface list: (count: 1) (bridge-only: 0)
    Vlan2001, uptime: 00:09:32, igmp (vpc-svi)


(60.60.60.2/32, 225.0.0.1/32), uptime: 00:09:31, ip(0) mrib(1) pim(0)
  RPF-Source: 60.60.60.2 [20/110]
  Data Created: Yes
  VPC Flags
--More--2018 Jun  5 19:11:21 Th-agg1 %$ VDC-1 %$ Jun  5 19:11:20 %KERN-2-SYSTEM_MSG:
[12633.219113] App vsh.bin on slot 1 vdc 1 SUP
sap 64098(cli_api queue) did not drop MTS_OPC_CLISH with msg_id 0x675ecf from sender
```

```
sap 64132(NULL) in 180 sec,
contact app owner - kernel
```

There is no functionality impact or traffic loss due to this issue. All the MTS messages are drained once the show command displays the complete output, the user enters CTRL+c, or the session gets closed.

- 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.
- ISSU is not supported if EPLD is not at Cisco NX-OS Release 7.0(3)I3(1) or later.
- A simplified NX-OS numbering format is used for platforms that are supported in Cisco NX-OS 9.3(x) releases. In order to support a software upgrade from releases prior to Cisco NX-OS Release 7.0(3)I7(4) that have the old release format, an installer feature supplies an I9(x) label as a suffix to the actual release during the **install all** operation. This label is printed as part of the image during the install operation from any release prior to Cisco NX-OS Release 7.0(3)I7(4) to 9.3(x), and it can be ignored. See the following example.

```
switch# install all nxos bootflash:nxos.9.3.1.bin
Installer will perform compatibility check first. Please wait.
Installer is forced disruptive

Verifying image bootflash:/nxos.9.3.1.bin for boot variable "nxos".
[####################] 100% -- SUCCESS

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

Preparing "nxos" version info using image bootflash:/nxos.9.3.1.bin.
[####################] 100% -- SUCCESS

Preparing "bios" version info using image bootflash:/nxos.9.3.1.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 for ISSU

Images will be upgraded according to following table:
Module   Image               Running-Version(pri:alt)  New-Version         Upg-Required
------  -------  ------------------------------------ -------------------  ------------

  1      nxos                                 7.0(3)I7(3)          9.3(1)I9(1)
 yes
  1      bios     v07.61(04/06/2017):v07.61(04/06/2017)   v05.33(09/08/2018)
 yes


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

- A nondisruptive standard ISSU is supported from Cisco NX-OS Release 7.0(3)I7(4), 7.0(3)I7(5), 7.0(3)I7(6), or 9.2(x) to Cisco NX-OS Release 9.3(1). For more information, see the ISSU Support Matrix.
- Beginning with Cisco NX-OS Release 9.3(5), standard, nondisruptive ISSU, **on switches that are configured with uRPF**, is supported on the following:
  - Cisco Nexus 9300-EX platform switches
  - Cisco Nexus 9300-FX/FX2 platform switches
  - Cisco Nexus 9300-GX platform switches

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

- ISSU is blocked if **boot poap enable** is configured.
- On performing a non-disruptive ISSU from Cisco NX-OS Release 7.0(3)I6(1) to any higher version, a traffic loss might occur based on the number of VLANs configured. To avoid traffic loss, it is recommended to increase the routing protocol's graceful restart timer to higher value. The recommended value of the graceful restart timer is 600 seconds. You can further increase or decrease this value based on the scale of the configuration.

ISSU Platform Support

The following tables identify the platforms supporting standard and enhanced ISSU, and the release when the support was introduced.

Note An enhanced ISSU can be performed only from a Cisco NX-OS Release 7.0(3)I5(1) to a later image. The upgrade will be disruptive.

Non-disruptive enhanced ISSU to Cisco NX-OS Release 9.3(1) is not supported as there are kernel fixes that cannot take effect without reloading the underlying kernel. The upgrade will be disruptive.

A non-disruptive enhanced ISSU from Cisco NX-OS Release 9.3(1) to later releases is supported in fallback mode only, even in cases of kernel incompatibility.

ISSU for Cisco Nexus 9200 Platform Switches

ISSU Type	Release/Supported Platforms	Features Not Supported with Non-disruptive ISSU
Standard	Beginning with Cisco NX-OS Release 7.0(3)I6(1): Cisco Nexus 9236C Cisco Nexus 9272Q Cisco Nexus 92160YC-X Cisco Nexus 92300YC Cisco Nexus 92304QC Beginning with Cisco NX-OS Release 9.3(3): Cisco Nexus 92348GC-X	Both ISSU types are disruptive for Cisco Nexus 9200 platform switches configured with the following features: • Segment routing • Tetration
Enhanced	Beginning with Cisco NX-OS Release 7.0(3)I7(3): Cisco Nexus 9236C Cisco Nexus 9272Q Cisco Nexus 92160YC-X Cisco Nexus 92300YC Cisco Nexus 92304QC Beginning with Cisco NX-OS Release 9.3(5): Cisco Nexus 92348GC-X	

ISSU for Cisco Nexus 9300 Platform Switches

ISSU Type	Release/Supported Platforms	Features Not Supported with Non-disruptive ISSU
Standard	Beginning with Cisco NX-OS Release 7.0(3)I3(1): Cisco Nexus 9332PQ Cisco Nexus 9372PX Cisco Nexus 9372PX-E Cisco Nexus 9372TX Cisco Nexus 9372TX-E Cisco Nexus 9396PX Cisco Nexus 9396TX Cisco Nexus 93120TX Cisco Nexus 93128TX Beginning with Cisco NX-OS Release 9.3(3): Cisco Nexus 9332C Cisco Nexus 9364C **Note** ISSU on Cisco 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 Cisco Nexus 9300 platform switches configured with the following features: • Dual-homed FEX • Segment routing • VXLAN **Note** Straight-through FEX is supported on Cisco Nexus 9372PX and 9396PX switches starting with Cisco NX-OS Release 7.0(3)I4(1).
Enhanced		

ISSU Type	Release/Supported Platforms	Features Not Supported with Non-disruptive ISSU
	Beginning with Cisco NX-OS Release 7.0(3)I5(1): Cisco Nexus 9332PQ Cisco Nexus 9372PX Cisco Nexus 9372PX-E Cisco Nexus 9372TX Cisco Nexus 9372TX-E Cisco Nexus 9396PX Cisco Nexus 9396TX Cisco Nexus 93120TX Cisco Nexus 93128TX Beginning with Cisco NX-OS Release 9.3(5): Cisco Nexus 9332C Cisco Nexus 9364C **Note** ISSU on Cisco 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 Cisco Nexus 9300-EX Platform Switches

ISSU Type	Release/Supported Platforms	Features Not Supported with Non-disruptive ISSU
Standard	Beginning with Cisco NX-OS Release 7.0(3)I6(1): Cisco Nexus 93108TC-EX Cisco Nexus 93180YC-EX Beginning with Cisco NX-OS Release 7.0(3)I7(1): Cisco Nexus 93180LC-EX	Both ISSU types are disruptive for Cisco Nexus 9300-EX platform switches configured with the following features: • Straight-through FEX • Dual-homed FEX • Segment routing • Tetration
Enhanced	Beginning with Cisco NX-OS Release 7.0(3)I7(3): Cisco Nexus 93108TC-EX Cisco Nexus 93180YC-EX Cisco Nexus 93180LC-EX	

ISSU for Cisco Nexus 9300-FX Platform Switches

ISSU Type	Release/Supported Platforms	Features Not Supported with Non-disruptive ISSU
Standard	Cisco NX-OS Release 9.3(1) and 9.3(2): None Beginning with Cisco NX-OS Release 9.3(3): Cisco Nexus 9336C-FX2 Cisco Nexus 93240YC-FX2 Cisco Nexus 93240YC-FX2Z Cisco Nexus 9348GC-FXP Cisco Nexus 93108TC-FX Cisco Nexus 93180YC-FX	Standard ISSU is disruptive for Cisco Nexus 9300-FX platform switches configured with the following features: • Straight-through FEX • Dual-homed FEX
Enhanced	Cisco NX-OS Release 9.3(1), 9.3(2), and 9.3(3): None Beginning with Cisco NX-OS Release 9.3(5): Cisco Nexus 9336C-FX2 Cisco Nexus 93240YC-FX2 Cisco Nexus 93216TC-FX2 Cisco Nexus 93360YC-FX2 Cisco Nexus 93240YC-FX2Z Cisco Nexus 9348GC-FXP Cisco Nexus 93108TC-FX Cisco Nexus 93180YC-FX	Enhanced ISSU is disruptive for Cisco Nexus 9300-FX platform switches configured with the following features: • Straight-through FEX • Dual-homed FEX Enhanced ISSU is not supported for Cisco Nexus 93180YC-FX and 93360YC-FX2 with FCoE features.

ISSU for Cisco Nexus 9500 Platform Switches

ISSU Type	Release/Supported Platforms	Features Not Supported with Non-disruptive ISSU
Standard	Beginning with Cisco NX-OS Release 7.0(3)I3(1): Cisco Nexus 9504, 9508, and 9516 with dual supervisor modules, a minimum of two system controllers, two fabric modules, and any of the following line cards: • Cisco Nexus 9432PQ • Cisco Nexus 9464PX • Cisco Nexus 9464TX • Cisco Nexus 9536PQ • Cisco Nexus 9564PX • Cisco Nexus 9564TX • Cisco Nexus 9636PQ **Note** Cisco Nexus 9500 platform switches with -R, -EX, or -FX line cards do not support ISSU.	Standard ISSU is disruptive for Cisco Nexus 9500 platform switches configured with the following features: • Dual-homed FEX • Segment routing • VXLAN **Note** Straight-through FEX is supported on Cisco Nexus 9500 platform switches with a Cisco Nexus 9464PX or 9564PX line card starting with Cisco NX-OS Release 7.0(3)I4(1).
Enhanced	None	

ISSU for Cisco Nexus 3000 Platform Switches Running Cisco Nexus 9000 Series NX-OS Software

ISSU Type	Release/Supported Platforms	Features Not Supported with Non-disruptive ISSU
Standard	Beginning with Cisco NX-OS Release 7.0(3)I3(1): Cisco Nexus 3164Q Cisco Nexus 31128PQ Beginning with Cisco NX-OS Release 7.0(3)I6(1): Cisco Nexus 3132Q-V Cisco Nexus 31108PC-V Cisco Nexus 31108TC-V Cisco Nexus 3232C Cisco Nexus 3264Q	Standard ISSU is disruptive for Cisco Nexus 3000 patform switches running Cisco Nexus 9000 Series NX-OS Software configured with the following features: • Segment routing on Cisco Nexus 3164Q, 31128PQ, 3232C, and 3264Q • VXLAN on Cisco Nexus 3164Q and 31128PQ

ISSU Type	Release/Supported Platforms	Features Not Supported with Non-disruptive ISSU
Enhanced	Beginning with Cisco NX-OS Release 7.0(3)I5(1): Cisco Nexus 3164Q Cisco Nexus 31128PQ Cisco Nexus 3132Q-V Cisco Nexus 31108PC-V Cisco Nexus 31108TC-V	Enhanced ISSU is disruptive for Cisco Nexus 3000 platform switches running Cisco Nexus 9000 Series NX-OS Software configured with the following features: • Segment routing on Cisco Nexus 3164Q and 31128PQ • VXLAN on Cisco Nexus 3164Q and 31128PQ

Cisco NX-OS Software Downgrade Guidelines

Before attempting to downgrade to an earlier software release, follow these guidelines:

- The only supported method of downgrading a Cisco 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 Cisco NX-OS Release 9.3(x) to an earlier release.

  - Performing an ISSU downgrade from Cisco 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
  - Downgrading with PVLANs (Private VLANs) configured is only supported with Cisco NX-OS 6.1(2)I3(4x) releases.
  - For a boot-variable change and reload to Cisco NX-OS Release 7.0(3)I1(1x), the PVLAN process is not brought up, and the PVLAN ports are kept down. For a boot-variable change to the Cisco NX-OS Release 6.1(2)I3(3) and earlier, an ASCII replay will be tried, but feature PVLANs and other PVLAN configurations will fail.

- When downgrading from the Cisco 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 Cisco Nexus 9500-series switch, with -FX or -FX+EX line cards, from Cisco NX-OS Release 9.3(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 Cisco NX-OS Release 9.3(x) to earlier releases, make sure that the ACL TCAM usage for ingress features does exceed the allocated TCAM space in the absence of the label sharing feature. Label sharing is a new feature in Cisco 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.
- The following limitation applies to Cisco Nexus platform switches that support Trust Anchor Module (TAM):

  The TACACS global key cannot be restored when downgrading from Cisco 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 Cisco 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. See the Hardware Installation Guide for your specific chassis.
- Cisco NX-OS automatically installs and enables the guest shell by default. However, if the device is reloaded with a Cisco 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 Cisco NX-OS image.
- You must delete the switch profile (if configured) when downgrading from a Cisco 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 nondisruptive downgrades, are not supported.

ISSU Upgrade Compatibility

For ISSU compatibility for all releases, see the Cisco NX-OS ISSU Support Matrix.

Upgrade Paths

Upgrading from a 7.x release to a 9.3(x) release may require more than a single hop. The following section describe the upgrade paths required.

Upgrade Paths to Release 9.3(x) from 7.0(3)F3(x) Releases

The following are the upgrade paths from previous 7.0(3)F3(x) releases:

- Release 7.0(3)F3(x) -> Release 7.0(3)F3(4) -> Release 9.3(x)

Note This upgrade is disruptive.

Upgrade Patch Instructions

On Cisco Nexus 9500 series switches only, a software upgrade from Cisco NX-OS Release 7.0(3)I1(2), 7.0(3)I1(3), or 7.0(3)I1(3a) to any other Cisco 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):

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

```
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
```

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

```
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
```

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

```
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
```

4. Proceed with an NX-OS software upgrade to the desired target release with the **install all** command.

```
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 "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
    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 CCTRL driver for card_type 3 ...
CCTRL driver for card_index 21000 ...
old data: 4000004 new data: 1
Not Micron SSD...

Checking all filesystems.......
Installing default sprom 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#
```

Configuring Enhanced ISSU

You can enable or disable enhanced (LXC) ISSU.

Note Enhanced ISSU to Cisco NX-OS Release 9.3(1) is not supported as there are kernel fixes that cannot take effect without reloading the underlying kernel. Enhanced ISSU from Cisco NX-OS Release 9.3(1) to later releases is supported, even in cases of kernel incompatibility.

Before you begin

Before you enable the LXC mode, ensure that the installed licenses do not include the 27000 string in the license file.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config#)`
		Enters global configuration mode.

Step 2	[**no**] **boot mode lxc**
	Example:
	`switch(config)# boot mode lxc`
	`Using LXC boot mode`
	Example:
	`switch(config)# no boot mode lxc`
	`Using normal native boot mode`
		Enables or disables enhanced (LXC) ISSU.
		Note In order to perform a nondisruptive enhanced ISSU, you must first boot the switch in LXC mode.

Step 3	(Optional) **show boot mode**
	Example:
	`switch(config)# show boot mode`
	`LXC boot mode is enabled`
	Example:
	`switch(config)# show boot mode`
	`LXC boot mode is disabled`
		Shows whether enhanced (LXC) ISSU is enabled or disabled.

Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the running configuration to the startup configuration.

Step 5	**reload**
	Example:
	`switch(config)# reload`
	`This command will reboot the system. (y/n)?  [n] Y`
	`loader>`
		Reloads the device. When prompted, press **Y** to confirm the reboot.

What to do next

Follow the instructions in Upgrading the Cisco NX-OS Software section. Make sure to choose the **non-disruptive** option if you want to perform an enhanced or regular ISSU.

Upgrading the Cisco NX-OS Software

Use this procedure to upgrade to a Cisco NX-OS 9.3(x) release.

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

Note If an error message appears during the upgrade, the upgrade will fail because of the reason indicated. See the Cisco Nexus 9000 Series NX-OS Troubleshooting Guide for a list of possible causes and solutions.

Before you begin

Before performing a nondisruptive ISSU to Cisco NX-OS Release 9.3(1), you must configure the BGP graceful restart timer to 180 seconds for Cisco Nexus 3132Q-V platform switches.

Procedure

Step 1 **Read the release notes for the software image file for any exceptions to this upgrade procedure.** See the Cisco Nexus 9000 Series NX-OS Release Notes.

Step 2 Log in to the device on the console port connection.

Step 3 Ensure that the required space is available for the image file to be copied.

```
switch# dir bootflash:
49152    Dec 10 14:43:39 2018 lost+found/
80850712 Dec 10 15:57:44 2018 n9000-dk9.9.2.1.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 Cisco NX-OS software on the device to use if the new image file does not load successfully.

Step 4 If you need more space on the active supervisor module, delete unnecessary files to make space available.

```
switch# delete bootflash:n9000-dk9.9.2.1.bin
```

Step 5 Verify that there is space available on the standby supervisor module.

```
switch# dir bootflash://sup-standby/
49152    Dec 10 14:43:39 2018 lost+found/
80850712 Dec 10 15:57:44 2018 n9000-dk9.9.2.1.bin
...

Usage for bootflash://sup-standby
 4825743360 bytes used
16312102912 bytes free
21137846272 bytes total
```

Step 6 If you need more space on the standby supervisor module, delete any unnecessary files to make space available.

```
switch# delete bootflash://sup-standby/n9000-dk9.9.2.1.bin
```

Step 7 Log in to Cisco.com, choose the software image file for your device from the following URL, and download it to a file server: http://software.cisco.com/download/navigator.html.

Step 8 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/nxos.9.3.1.bin bootflash:nxos.9.3.1.bin
```

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

```
switch# copy scp://user@scpserver.cisco.com//download/nxos.9.3.5.bin
bootflash:nxos.9.3.5.bin compact vrf management use-kstack

user1@10.65.42.196's password:
nxos.9.3.5.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 Cisco Nexus 9300-series platform switches.

Step 9 You can detect an incomplete or corrupt NX-OS software image prior to performing an upgrade by verifying the MD5 or SHA256 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 SHA256 checksum of the software image, run the **show file bootflash:**<*IMAGE-NAME*> **sha256sum** command and compare the resulting value to the published SHA256 checksum for the software image on Cisco's Software Download website.

```
switch# show file bootflash:nxos.9.3.1.bin sha256sum
5214d563b7985ddad67d52658af573d6c64e5a9792b35c458f5296f954bc53be

switch# show file bootflash:nxos.9.3.1.bin md5sum
e55f6496a0b445e2adf58fdfd856b5ec
```

Step 10 Check the impact of upgrading the software before actually performing the upgrade.

```
switch# show install all impact nxos bootflash:nxos.9.3.1.bin
```

During the compatibility check, the following ISSU-related messages may appear in the Reason field:

Reason Field Message	Description
Incompatible image for ISSU	The Cisco NX-OS image to which you are attempting to upgrade does not support ISSU.

Reason Field Message	Description
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 11 Save the running configuration to the startup configuration.

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

Step 12 Upgrade the Cisco NX-OS software using the **install all nxos bootflash:***filename* [**no-reload** | **non-disruptive** | **non-interruptive** | **serial**] command.

```
switch# install all nxos bootflash:nxos.9.3.1.bin
```

The following options are available:

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

  Note 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.

- **non-disruptive**—Performs an in-service software upgrade (ISSU) to prevent the disruption of data traffic. (By default, the software upgrade process is disruptive.)
- **non-interruptive**—Upgrades the software without any prompts. This option skips all error and sanity checks.
- **serial**—Upgrades the I/O modules in Cisco 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.)

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.

Step 13 (Optional) Display the entire upgrade process.

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

Step 14 (Optional) Log in and verify that the device is running the required software version.

```
switch# show version
```

Step 15 (Optional) If necessary, install any licenses to ensure that the required features are available on the device. See the Cisco NX-OS Licensing Guide.

Upgrade Process for vPCs

Upgrade Process for a vPC Topology on the Primary Switch

The following list summarizes the upgrade process on a switch in a vPC topology that holds either the Primary or Operational Primary vPC roles. Steps that differ from a switch upgrade in a non-vPC topology are in bold.

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.

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.

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

The following list summarizes the upgrade process on a switch in a vPC topology that holds either the Secondary or Operational Secondary vPC roles. Steps that differ from a switch upgrade in a non-vPC topology are in bold.

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.

Downgrading to an Earlier Software Release

Note If an error message appears during the downgrade, the downgrade will fail because of the reason indicated. See the Cisco Nexus 9000 Series NX-OS Troubleshooting Guide for a list of possible causes and solutions.

Procedure

Step 1 **Read the release notes for the software image file for any exceptions to this downgrade procedure.** See the Cisco Nexus 9000 Series NX-OS Release Notes.

Step 2 Log in to the device on the console port connection.

Step 3 Verify that the image file for the downgrade is present on the active supervisor module bootflash:.

```
switch# dir bootflash:
```

Step 4 If the software image file is not present, log in to Cisco.com, choose the software image file for your device from the following URL, and download it to a file server: http://software.cisco.com/download/navigator.html.

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

Step 5 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/n9000-dk9.9.2.1.bin
bootflash:n9000-dk9.9.2.1.bin
```

Step 6 Check for any software incompatibilities.

```
switch# show incompatibility-all nxos bootflash:n9000-dk9.9.2.1.bin
Checking incompatible configuration(s)
No incompatible configurations
```

The resulting output displays any incompatibilities and remedies.

Step 7 Disable any features that are incompatible with the downgrade image.

Step 8 Check for any hardware incompatibilities.

```
switch# show install all impact nxos bootflash:n9000-dk9.9.2.1.bin
```

Step 9 Power off any unsupported modules.

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

Step 10 Save the running configuration to the startup configuration.

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

Step 11 Downgrade the Cisco NX-OS software.

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 12 (Optional) Display the entire downgrade process.

Example:

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

Step 13 (Optional) Log in and verify that the device is running the required software version.

```
switch# show version
```

Cisco NX-OS Upgrade History

During the life of a Cisco 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 may 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.

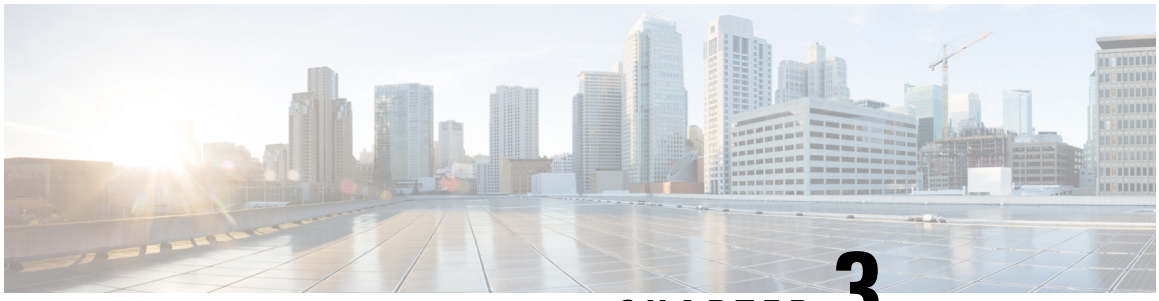
Beginning with Cisco NX-OS Release 9.3(5), Cisco Nexus 9000 switches log all upgrade activity performed over time providing a comprehensive history of these events. The stored upgrade history types are:

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

View the Cisco 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 EPLD          n9000-    26 Apr 2020 11:37:16    EPLD Upgrade completed
                   epld.9.3.4.img
NXOS EPLD          n9000-    26 Apr 2020 11:32:41    EPLD Upgrade started
                   epld.9.3.4.img
NXOS system image  9.3(5)    24 Mar 2020 20:09:10    Installation End
NXOS system image  9.3(5)    24 Mar 2020 20:05:29    Installation started
NXOS SMU           9.3(5)    03 Mar 2020 23:34:15    Patch activation ended for
```

```
                                                    nxos.libnbproxycli_patch-n9k_
                                                    ALL-1.0.0-9.3.5.lib32_n9000.rpm
NXOS SMU            9.3(5)    03 Mar 2020 23:34:03    Patch activation started for
                                                    nxos.libnbproxycli_patch-n9k_
                                                    ALL-1.0.0-9.3.5.lib32_n9000.rpm
```



CHAPTER 3

Optionality in Cisco NX-OS Software

This chapter describes optionality in Cisco NX-OS software.

Optionality in Cisco NX-OS Software

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

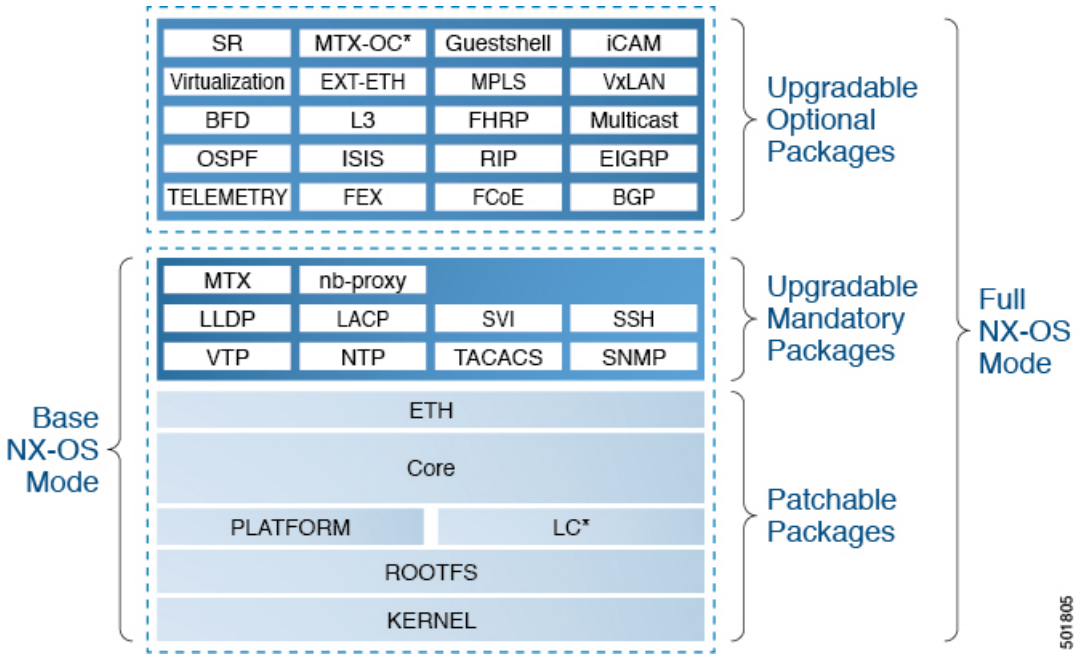
The advantages for using modular Cisco NX-OS software are:

- Lean NX-OS software
- Asynchronous delivery of the features and the fixes: Quick fixes are provided that are independent of the releases, including new features.
- Reduced footprint of binaries and libraries at run time

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

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

Figure 2: Optionality in Cisco 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 plugin manager. There is no change in the user behavior as compared to the previous releases.

Using Modular Packages

The Cisco NX-OS software image is traditionally constructed with the packaging that forms a Cisco Linux distribution. It makes upgrading certain packages difficult as each package is large in size.

This section describes a new package management for the Cisco NX-OS software image. Beginning with Cisco 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 YUM

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.

YUM (Yellowdog Updater, Modified) 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. YUM 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, YUM works with the repositories that are collections of the packages and they are typically accessible over a network connection.

Booting the NX-OS Image in Base or Full Mode

You can now 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, see the following steps:

- 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 CLIs for Feature RPM Operation section.

Information About RPMs

RPMs can be upgraded or downgraded to a new software version using NXOS install commands or by using YUM commands. An upgradable RPM can be optional or mandatory.

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

Format of the RPM

The general format of a RPM is <name>-<version>-<release>.<arch>.rpm. The same format is followed for NXOS 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

See the following table for 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 **Example: fex-2.0.0.0-9.2.1.lib32_n9000.rpm**	**Description**
fex	Indicates the name of the component.
2	Indicates that the RPM is not backward compatible. Configuration loss takes place during an upgrade.
0	Indicates the incremental API changes/CLI 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 NXOS 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 **yum** or **install** CLI commands from the switch.

See the following list of the optional RPMs and their associated features:

Table 3: List of Optional RPMs and Their Associated Features

Package Name	Associated Features
BGP	feature bgp
BFD	feature bfd
Container-tracker	feature container-tracker
EIGRP	feature eigrp
Ext-Eth	• feature openflow • feature evb • feature imp • feature netflow • feature sla_sender • feature sla_responder • feature sla twamp-server • feature sflow
FCoE	• feature-set fcoe • feature-set fcoe-npv
FEX	feature-set fex
FHRP	• feature hsrp • feature vrrpv3
iCAM	feature icam
ISIS	feature isis
MPLS	• feature mpls segment-routing • feature mpls evpn

Package Name	Associated Features
Multicast	• feature pim • feature pim6 • feature msdp • feature ngmvpn
OSPF	• feature ospf • feature ospfv3
RIP	feature rip
Services	feature catena
SR	feature mpls segment-routing traffic-engineering
TELEMETRY	feature telemetry
Virtualization	NA
VXLAN	• feature nv overlay • feature fabric forwarding

Guidelines for NX-OS Feature RPM Installation

See the following NX-OS system RPM repositories that are present in the Cisco NX-OS Series switches for the RPM management.

Note Avoid manually copying the RPMs to system repositories. Instead use the install or YUM 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. YUM commands can be used to populate the repository too. The maximum space for the repository is 200Mb along with the patching repository for Cisco Nexus 9000 Series switches except Cisco Nexus 3000 Series switches. For Cisco 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. YUM commands or **install** command can be used for the installation or the removal of these RPMs.

The following 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.

Using Install CLIs for Feature RPM Operation

See the following reference table for using install CLIs for the feature RPM operations:

Table 5: Reference for Install CLIs for the Feature RPM Operations

CLI	Description

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

install reset nxos base	This operation installs NXOS in base mode by removing all patches, upgraded packages, persisted etc configurations, third party packages installed, startup configuration (write erase), and reloads the switch with the default packages.
install reset nxos full	This operation installs NXOS 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).
install add <>	Adds an RPM file to respective repository and updates the repository (**patch/feature/third-party**).
install activate <*rpm name*>	Installs an RPM that is present in the repository.
install commit <*rpm name*>	Used for the patch RPMs. Makes the patch persist during reload.
install deactivate <*rpm name*>	Un-installs an RPM.
install remove <*rpm name*>	Removes an RPM file from the repository and updates the repository.
sh install active	Displays the list of the installed RPMs in the system apart from base rootfs RPMs. (features/patch/third-party).

CLI	Description
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.

Using Install CLIs for Digital Signature Support

Use the following CLI commands to install CLIs for digital signature support:

Procedure

	Command or Action	Purpose

Step 1	switch#**install add bootflash**:<*keyfile*> **gpg-key**
	Example:
	`install add bootflash:RPM-GPG-KEY-puppetlabs gpg-key`
	`[####################] 100%`
	`Install operation 304 completed successfully at Thu Jun 19 16:40:28 2018`
		Cisco release RPMs are signed with Cisco 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	switch#**install verify package** <*package-name*>	Verifies the package.

Step 3	OR switch#**install verify bootflash**:<*RPM file*>
	Example:
	`switch# install verify bootflash:vxlan-2.0.0.0-9.2.1.lib32_n9000.rpm`
	`RSA signed`
	`switch#`
		Use step 2 or 3 to verify whether the RPM file is a signed or non-signed file.

Querying All Installed RPMs

Complete the following step to query all the installed RPMs:

Procedure

	Command or Action	Purpose

Step 1	**show install packages**
	Example:
	`switch# show install packages`
	`Boot Image:`
		Queries all the installed RPMs.

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

	Command or Action	Purpose
	Unsigned switch#	

Installing the RPMs Using One Step Procedure

The CLIs for both install and upgrade RPMs are the same. See the following step to install the RPMs using one step procedure:

Procedure

	Command or Action	Purpose
Step 1	**install add** *<rpm>* **activate** **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	Installs and activates the RPM.

Example

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

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

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

```
        tacacs-2.0.0.0-9.2.1.lib32_n9000
        vtp-2.0.0.0-9.2.1.lib32_n9000
switch(config)#
```

Installing the RPMs Using Two Steps Procedure

The CLIs for both install and upgrade RPMs are the same. See the following steps to install the RPMs using two steps procedure:

Procedure

	Command or Action	Purpose

Step 1	**install add** <*rpm*>
	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`
	`switch(config)# sh install inactive | grep vxlan`
	`vxlan-2.0.1.0-9.2.1.lib32_n9000`
		Installs the RPM.

Step 2	**install activate** <*rpm*> **Example:**	Activates the RPM.

Example

```
switch#install activate vxlan

[####################] 100%
Install operation 891 completed successfully at Thu Jun  7 13:53:07 2018

switch# show install active | grep vxlan

vxlan-2.0.0.0-9.2.1.lib32_n9000

switch# sh install inactive | grep vxlan

switch#
```

Upgrading the RPMs Using One Step

The CLIs for both install and upgrade RPMs are the same. See the following steps to upgrade the RPMs:

Procedure

	Command or Action	Purpose

Step 1	**install add** *<rpm>***activate upgrade**
	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`
		Installs the RPM.

Example

```
switch(config)# show install active

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

Active Packages:
bgp-2.0.2.0-9.2.1.lib32_n9000
chef-12.0.0alpha.2+20150319234423.git.1608.b6eb10f-1.el5.x86_64

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

Downgrading the RPMs

The downgrade procedure needs a special CLI attribute. See the following step to downgrade the RPMs using the one step procedure:

Procedure

	Command or Action	Purpose

Step 1	**install add** *<rpm>***activate downgrade**
	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`
		Downgrades the RPM.

Example

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

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

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

Removing the RPMs

See the following steps to remove the RPMs:

Procedure

	Command or Action	Purpose

Step 1	**install remove** *<rpm>*
	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`
		Removes the RPM from the repository.

Information About YUM Commands

See the following sections for more information about YUM commands.

Note YUM commands do not support ctrl+c. Install commands do support ctrl+c. If YUM commands are aborted using ctrl+c, manual cleanup must be performed using "/isan/bin/patching_utils.py --unlock".

Performing Package Operations Using the YUM Commands

See the following sections for performing package operations using the YUM commands:

Note YUM commands are accessed only from the BASH shell on the box and they are not allowed from the NXOS VSH terminal.

Note Make sure that as a sudo user, you have access to the super user privileges.

Finding the Base Version RPM of the Image

Use the **ls /rpms** command to find the base version RPM of the image. The base RPM version is the pre-installed RPM that is archived in the system image.

```
#ls /rpms

bfd-2.0.0.0-9.2.1.lib32_n9000.rpm
ins_tor_sdk_t2-1.0.0.0-9.2.0.77.lib32_n9000.rpm
mtx-netconf-agent-2.0.0.0-9.2.1.lib32_n9000.rpm   snmp-2.0.0.0-9.2.1.lib32_n9000.rpm
bgp-2.0.0.0-9.2.1.lib32_n9000.rpm
ins_tor_sdk_t3-1.0.0.0-9.2.0.77.lib32_n9000.rpm
mtx-restconf-agent-2.0.0.0-9.2.1.lib32_n9000.rpm  sr-2.0.0.0-9.2.1.lib32_n9000.rpm
container-tracker-2.0.0.0-9.2.1.lib32_n9000.rpm  isis-2.0.0.0-9.2.1.lib32_n9000.rpm
         mtx-telemetry-2.0.0.0-9.2.1.lib32_n9000.rpm      svi-2.0.0.0-9.2.1.lib32_n9000.rpm
eigrp-2.0.0.0-9.2.1.lib32_n9000.rpm               lacp-2.0.0.0-9.2.1.lib32_n9000.rpm
          nbproxy-2.0.0.0-9.2.1.lib32_n9000.rpm
tacacs-2.0.0.0-9.2.1.lib32_n9000.rpm
ext-eth-2.0.0.0-9.2.1.lib32_n9000.rpm             lldp-2.0.0.0-9.2.1.lib32_n9000.rpm
          ntp-2.0.0.0-9.2.1.lib32_n9000.rpm
telemetry-2.3.4.0-9.2.1.lib32_n9000.rpm
fcoe-2.0.0.0-9.2.1.lib32_n9000.rpm                mcast-2.0.0.0-9.2.1.lib32_n9000.rpm
          nxos-ssh-2.0.0.0-9.2.1.lib32_n9000.rpm
virtualization-2.0.0.0-9.2.1.lib32_n9000.rpm
fex-2.0.0.0-9.2.1.lib32_n9000.rpm                 mpls-2.0.0.0-9.2.1.lib32_n9000.rpm
         ospf-2.0.0.0-9.2.1.lib32_n9000.rpm               vtp-2.0.0.0-9.2.1.lib32_n9000.rpm
fhrp-2.0.0.0-9.2.1.lib32_n9000.rpm                mtx-device-2.0.0.0-9.2.1.lib32_n9000.rpm
          repodata
vxlan-2.0.0.0-9.2.1.lib32_n9000.rpm
guestshell-2.0.0.0-9.2.1.lib32_n9000.rpm        mtx-grpc-agent-2.0.0.0-9.2.1.lib32_n9000.rpm
       rip-2.0.0.0-9.2.1.lib32_n9000.rpm
icam-2.0.0.0-9.2.1.lib32_n9000.rpm                mtx-infra-2.0.0.0-9.2.1.lib32_n9000.rpm
          services-2.0.0.0-9.2.1.lib32_n9000.rpm
```

Checking the List of the Installed RPMs

Use the **yum list installed** command to query the feature and third party RPMs and grep a specific RPM. See the following example for feature RPMs:

```
bash-4.2# yum 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
...
```

Getting Details of the Installed RPMs

The **yum info** <*rpmname*> command lists out the detailed info of the installed RPM.

```
yum info vxlan

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo

                      | 1.1 kB     00:00 ...
localdb

                      |  951 B     00:00 ...
patching

                      |  951 B     00:00 ...
thirdparty

                      |  951 B     00:00 ...
Installed Packages
Name        : vxlan
Arch        : lib32_n9000
Version     : 2.0.0.0
Release     : 9.2.1
Size        : 6.4 M
Repo        : installed
From repo   : groups-repo
Summary     : Cisco NXOS VxLAN
URL         : http://cisco.com/
License     : Proprietary
Description : Provides VxLAN support
```

Installing the RPMs

Installing the RPMs downloads the RPMs and copies the respective program to the switches. See the following example for installing the RPMs from a remote server (that is reachable in the network):

```
bash-4.3# yum install
http://10.0.0.2/modularity/rpms/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo
                                                      | 1.1 kB     00:00 ...
localdb
                                                      |  951 B     00:00 ...
localdb/primary
                                                      |  886 B     00:00 ...
localdb
```

```
                                                                              1/1
patching
                                                    |  951 B     00:00 ...
thirdparty
                                                    |  951 B     00:00 ...
Setting up Install Process
vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm
                                                  | 1.6 MB     00:00
Examining /var/tmp/yum-root-RaANgb/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm:
vxlan-2.0.1.0-9.2.1.lib32_n9000
Marking /var/tmp/yum-root-RaANgb/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

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

 Package                  Arch                          Version
           Repository                                  Size
================================================================================
Installing:
vxlan                     lib32_n9000                   2.0.1.0-9.2.1
     /vxlan-2.0.1.0-9.2.1.lib32_n9000                   6.4 M
Transaction Summary
================================================================================
Install       1 Package

Total size: 6.4 M
Installed size: 6.4 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : vxlan-2.0.1.0-9.2.1.lib32_n9000
                                                                              1/1
starting pre-install package version mgmt for vxlan
pre-install for vxlan complete
starting post-install package version mgmt for vxlan
post-install for vxlan complete

Installed:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1


Complete!
```

See the following example for installing the RPMs from local bootflash:

```
sudo yum install /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo

                      | 1.1 kB     00:00 ...
localdb

                      |  951 B     00:00 ...
patching
```

```
                    |  951 B     00:00 ...
thirdparty

                    |  951 B     00:00 ...
Setting up Install Process
Examining /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm: vxlan-2.0.1.0-9.2.1.lib32_n9000
Marking /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm as an update to
vxlan-2.0.0.0-9.2.1.lib32_n9000
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

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

 Package                                  Arch
Version                                                  Repository
                                       Size
================================================================================
Updating:
 vxlan                                    lib32_n9000
2.0.1.0-9.2.1                                    /vxlan-2.0.1.0-9.2.1.lib32_n9000
                            6.4 M

Transaction Summary
================================================================================
Upgrade       1 Package

Total size: 6.4 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating   : vxlan-2.0.1.0-9.2.1.lib32_n9000

                                    1/2
starting pre-install package version mgmt for vxlan
pre-install for vxlan complete
starting post-install package version mgmt for vxlan
post-install for vxlan complete
  Cleanup    : vxlan-2.0.0.0-9.2.1.lib32_n9000

                                    2/2

Updated:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1


Complete!
```

See the following example for installing the RPM if it is available in a repository:

yum install eigrp

Upgrading the RPMs

See the following example for upgrading the RPMs from a remote server (that is reachable in the network):

```
bash-4.3# yum upgrade
http://10.0.0.2/modularity/rpms/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo
                                                    | 1.1 kB     00:00 ...
localdb
                                                    |  951 B     00:00 ...
patching
                                                    |  951 B     00:00 ...
thirdparty
                                                    |  951 B     00:00 ...
Setting up Upgrade Process
vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm
                                                  | 1.6 MB     00:00
Examining /var/tmp/yum-root-RaANgb/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm:
vxlan-2.0.1.0-9.2.1.lib32_n9000
Marking /var/tmp/yum-root-RaANgb/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm as an update to
vxlan-2.0.0.0-9.2.1.lib32_n9000
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

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

 Package                    Arch                            Version
           Repository                                     Size
================================================================================
Updating:
vxlan                      lib32_n9000                     2.0.1.0-9.2.1
     /vxlan-2.0.1.0-9.2.1.lib32_n9000                     6.4 M
Transaction Summary
================================================================================
Upgrade       1 Package

Total size: 6.4 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
** Found 1 pre-existing rpmdb problem(s), 'yum check' output follows:
busybox-1.23.2-r0.0.x86_64 has missing requires of busybox-syslog
  Updating   : vxlan-2.0.1.0-9.2.1.lib32_n9000
                                                                        1/2
starting pre-install package version mgmt for vxlan
pre-install for vxlan complete
starting post-install package version mgmt for vxlan
post-install for vxlan complete
  Cleanup    : vxlan-2.0.0.0-9.2.1.lib32_n9000
                                                                        2/2

Updated:
```

```
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1

Complete!
```

See the following example for upgrading the RPMs from local bootflash:

sudo yum upgrade /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm

```
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo

                      | 1.1 kB     00:00 ...
localdb

                      |  951 B     00:00 ...
patching

                      |  951 B     00:00 ...
thirdparty

                      |  951 B     00:00 ...
Setting up Upgrade Process
Examining /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm: vxlan-2.0.1.0-9.2.1.lib32_n9000
Marking /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm as an update to
vxlan-2.0.0.0-9.2.1.lib32_n9000
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

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

 Package                                  Arch
Version                                          Repository
                                  Size
================================================================================
Updating:
vxlan                                    lib32_n9000
2.0.1.0-9.2.1                                     /vxlan-2.0.1.0-9.2.1.lib32_n9000
                             6.4 M

Transaction Summary
================================================================================
Upgrade       1 Package

Total size: 6.4 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating   : vxlan-2.0.1.0-9.2.1.lib32_n9000

                                    1/2
starting pre-install package version mgmt for vxlan
pre-install for vxlan complete
```

```
starting post-install package version mgmt for vxlan
post-install for vxlan complete
  Cleanup    : vxlan-2.0.0.0-9.2.1.lib32_n9000

                                  2/2

Updated:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1



Complete!
```

See the following example for upgrading the RPMs if it is available in any repository:

```
yum upgrade eigrp
```

Downgrading the RPMs

See the following example for downgrading the RPMs from a remote server (that is reachable in the network):

```
sudo yum downgrade vxlan-2.0.0.0-9.2.1.lib32_n9000
```

```
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
Setting up Downgrade Process
groups-repo

                     | 1.1 kB     00:00 ...
localdb

                     |  951 B     00:00 ...
localdb/primary

                     | 1.3 kB     00:00 ...
localdb

                                       2/2
patching

                     |  951 B     00:00 ...
thirdparty

                     |  951 B     00:00 ...
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.0.0-9.2.1 will be a downgrade
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be erased
--> Finished Dependency Resolution

Dependencies Resolved

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

 Package                                    Arch
            Version                                                 Repository
                              Size
================================================================================
Downgrading:
```

```
vxlan                                              lib32_n9000
           2.0.0.0-9.2.1                                          groups-repo
                                  1.6 M
Transaction Summary
================================================================================
Downgrade     1 Package

Total download size: 1.6 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : vxlan-2.0.0.0-9.2.1.lib32_n9000

                                     1/2
starting pre-install package version mgmt for vxlan
pre-install for vxlan complete
starting post-install package version mgmt for vxlan
post-install for vxlan complete
  Cleanup    : vxlan-2.0.1.0-9.2.1.lib32_n9000

                                     2/2

Removed:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1



Installed:
  vxlan.lib32_n9000 0:2.0.0.0-9.2.1



Complete!
```

See the following example for downgrading the RPMs from local bootflash:

```
yum downgrade /bootflash/eigrp-2.0.0-9.2.1.lib32_n9000.rpm
```

See the following example for downgrading the RPMs if it is available in any repository:

```
yum downgrade eigrp
```

Deleting the RPMs

Deleting the RPMs de-installs the RPMs and removes any configuration CLI of the feature. Use the **yum erase** <*rpm*> command to delete the RPMs.

```
bash-4.2# sudo yum 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
                                Repository                                 Size
================================================================================
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 YUM Groups

The support for YUM groups is part of the package management. It simplifies the management of the packages for the administrators and it provides greater flexibility.

The administrators can group a list of packages (RPMs) into a logical group and they can perform various operations. YUM supports the following group commands:

- grouplist
- groupinfo
- groupinstall
- groupremove
- groupupdate

YUM groups can be broadly classified as L2, L3, routing, and management.

Using the grouplist Command

In Linux, number of packages are bundled to particular group. Instead of installing individual packages with yum, 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 **yum grouplist** command:

```
bash-4.2# sudo yum grouplist

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
Setting up Group Process
groups-repo

                      | 1.1 kB     00:00 ...
localdb

                      |  951 B     00:00 ...
patching

                      |  951 B     00:00 ...
thirdparty

                      |  951 B     00:00 ...
groups-repo/group

                      | 1.6 kB     00:00 ...
Installed Groups:
   L2
   L3
   management
Available Groups:
   routing
Done

bash-4.3$
```

Using the groupmembers Command

Use **yum groupinfo** command to display the description and the contents of a package group. The command lists out the feature members of the group.

```
bash-4.2# sudo yum groupinfo l2


Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
Setting up Group Process
groups-repo

                      | 1.1 kB     00:00 ...
localdb

                      |  951 B     00:00 ...
patching

                      |  951 B     00:00 ...
thirdparty

                      |  951 B     00:00 ...

Group: L2
 Mandatory Packages:
   lacp
   lldp
   svi
```

```
    vtp
```

Using the groupinstall Command

This command is for both install & upgrade of the members RPM. If the member is not installed, it will install the highest version available. If the member is already installed and higher RPM is available, it will upgrade that member.

```
bash-4.2# sudo yum groupinstall routing

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo

                      | 1.1 kB     00:00 ...
localdb

                      |  951 B     00:00 ...
patching

                      |  951 B     00:00 ...
thirdparty

                      |  951 B     00:00 ...
Setting up Group Process
Package ospf-2.0.0.0-9.2.1.lib32_n9000 already installed and latest version
Resolving Dependencies
--> Running transaction check
---> Package bgp.lib32_n9000 0:2.0.0.0-9.2.1 will be installed
---> Package eigrp.lib32_n9000 0:2.0.0.0-9.2.1 will be installed
---> Package isis.lib32_n9000 0:2.0.0.0-9.2.1 will be installed
---> Package rip.lib32_n9000 0:2.0.0.0-9.2.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

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

 Package                Arch                                        Version
                                          Repository                    Size
================================================================================
Installing:
 bgp                    lib32_n9000                                  2.0.0.0-9.2.1
                                          groups-repo                  2.4 M
 eigrp                  lib32_n9000                                  2.0.0.0-9.2.1
                                          groups-repo                  428 k
 isis                   lib32_n9000                                  2.0.0.0-9.2.1
                                          groups-repo                  1.2 M
 rip                    lib32_n9000                                  2.0.0.0-9.2.1
                                          groups-repo                  214 k
Transaction Summary
================================================================================
Install       4 Packages

Total download size: 4.2 M
Installed size: 19 M
Is this ok [y/N]: y
Downloading Packages:
--------------------------------------------------------------------------------
Total
```

```
            132 MB/s | 4.2 MB     00:00
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : rip-2.0.0.0-9.2.1.lib32_n9000

                                    1/4
starting pre-install package version mgmt for rip
pre-install for rip complete
starting post-install package version mgmt for rip
post-install for rip complete
  Installing : isis-2.0.0.0-9.2.1.lib32_n9000

                                    2/4
starting pre-install package version mgmt for isis
pre-install for isis complete
starting post-install package version mgmt for isis
post-install for isis complete
  Installing : eigrp-2.0.0.0-9.2.1.lib32_n9000

                                    3/4
starting pre-install package version mgmt for eigrp
pre-install for eigrp complete
starting post-install package version mgmt for eigrp
post-install for eigrp complete
  Installing : bgp-2.0.0.0-9.2.1.lib32_n9000

                                    4/4
starting pre-install package version mgmt for bgp
pre-install for bgp complete
starting post-install package version mgmt for bgp
post-install for bgp complete

Installed:
  bgp.lib32_n9000 0:2.0.0.0-9.2.1                   eigrp.lib32_n9000 0:2.0.0.0-9.2.1
             isis.lib32_n9000 0:2.0.0.0-9.2.1                   rip.lib32_n9000
0:2.0.0.0-9.2.1

Complete!
```

Using the groupupdate Command

Use the **yum groupupdate** command to update any existing installed group packages.

```
bash-4.3# yum groupupdate routing

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo

                     | 1.1 kB     00:00 ...
localdb

                     |  951 B     00:00 ...
localdb/primary

                     | 1.9 kB     00:00 ...
localdb
```

```
                                        6/6
patching

                     |  951 B     00:00 ...
thirdparty

                     |  951 B     00:00 ...
Setting up Group Process
Resolving Dependencies
--> Running transaction check
---> Package bgp.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package bgp.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
---> Package eigrp.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package eigrp.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
---> Package isis.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package isis.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
---> Package ospf.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package ospf.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
---> Package rip.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package rip.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

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

 Package                    Arch                                        Version
                                                Repository              Size
================================================================================
Updating:
 bgp                        lib32_n9000                                 2.0.1.0-9.2.1
                                                localdb                 2.4 M
 eigrp                      lib32_n9000                                 2.0.1.0-9.2.1
                                                locald                  428 k
 isis                       lib32_n9000                                 2.0.1.0-9.2.1
                                                local                   1.2 M
 ospf                       lib32_n9000                                 2.0.1.0-9.2.1
                                                localdb                 2.8 M
 rip                        lib32_n9000                                 2.0.1.0-9.2.1
                                                localdb                 214 k
Transaction Summary
================================================================================
Upgrade       5 Packages

Total download size: 7.0 M
Is this ok [y/N]: y
Downloading Packages:
--------------------------------------------------------------------------------
Total

            269 MB/s | 7.0 MB     00:00
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating   : eigrp-2.0.1.0-9.2.1.lib32_n9000

                                  1/10
starting pre-install package version mgmt for eigrp
pre-install for eigrp complete
starting post-install package version mgmt for eigrp
post-install for eigrp complete
  Updating   : ospf-2.0.1.0-9.2.1.lib32_n9000
```

```
                                  2/10
starting pre-install package version mgmt for ospf
pre-install for ospf complete
starting post-install package version mgmt for ospf
post-install for ospf complete
  Updating   : rip-2.0.1.0-9.2.1.lib32_n9000

                                  3/10
starting pre-install package version mgmt for rip
pre-install for rip complete
starting post-install package version mgmt for rip
post-install for rip complete
  Updating   : isis-2.0.1.0-9.2.1.lib32_n9000

                                  4/10
starting pre-install package version mgmt for isis
pre-install for isis complete
starting post-install package version mgmt for isis
post-install for isis complete
  Updating   : bgp-2.0.1.0-9.2.1.lib32_n9000

                                  5/10
starting pre-install package version mgmt for bgp
pre-install for bgp complete
starting post-install package version mgmt for bgp
post-install for bgp complete
  Cleanup    : bgp-2.0.0.0-9.2.1.lib32_n9000

                                  6/10
  Cleanup    : isis-2.0.0.0-9.2.1.lib32_n9000

                                  7/10
  Cleanup    : rip-2.0.0.0-9.2.1.lib32_n9000

                                  8/10
  Cleanup    : ospf-2.0.0.0-9.2.1.lib32_n9000

                                  9/10
  Cleanup    : eigrp-2.0.0.0-9.2.1.lib32_n9000

                                 10/10

Updated:
  bgp.lib32_n9000 0:2.0.1.0-9.2.1       eigrp.lib32_n9000 0:2.0.1.0-9.2.1
isis.lib32_n9000 0:2.0.1.0-9.2.1      ospf.lib32_n9000 0:2.0.1.0-9.2.1      rip.lib32_n9000
 0:2.0.1.0-9.2.1

Complete!
```

Using the grouperase Command

Use the **yum grouperase** command to delete the groups or all the RPM members of the group.

```
bash-4.3$ sudo yum grouperase routing

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
Setting up Group Process
groups-repo

                     | 1.1 kB     00:00 ...
localdb
```

```
                    |  951 B     00:00 ...
patching

                    |  951 B     00:00 ...
thirdparty

                    |  951 B     00:00 ...
Resolving Dependencies
--> Running transaction check
---> Package bgp.lib32_n9000 0:2.0.0.0-9.2.1 will be erased
---> Package eigrp.lib32_n9000 0:2.0.0.0-9.2.1 will be erased
---> Package isis.lib32_n9000 0:2.0.0.0-9.2.1 will be erased
---> Package ospf.lib32_n9000 0:2.0.0.0-9.2.1 will be erased
---> Package rip.lib32_n9000 0:2.0.0.0-9.2.1 will be erased
--> Finished Dependency Resolution

Dependencies Resolved

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

 Package              Arch                                          Version
                                        Repository                  Size
================================================================================
Removing:
 bgp                  lib32_n9000                                   2.0.0.0-9.2.1
                                        @groups-repo                11 M
 eigrp                lib32_n9000                                   2.0.0.0-9.2.1
                                        @groups-repo                2.0 M
 isis                 lib32_n9000                                   2.0.0.0-9.2.1
                                        @groups-repo                5.7 M
 ospf                 lib32_n9000                                   2.0.0.0-9.2.1
                                        @groups-repo                15 M
 rip                  lib32_n9000                                   2.0.0.0-9.2.1
                                        @groups-repo                1.0 M

Transaction Summary
================================================================================
Remove        5 Packages

Installed size: 34 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Erasing    : isis-2.0.0.0-9.2.1.lib32_n9000

                                    1/5
starting pre-remove package version mgmt for isis
pre-remove for isis complete
  Erasing    : ospf-2.0.0.0-9.2.1.lib32_n9000

                                    2/5
starting post-remove package version mgmt for isis
post-remove for isis complete
starting pre-remove package version mgmt for ospf
pre-remove for ospf complete
  Erasing    : eigrp-2.0.0.0-9.2.1.lib32_n9000

                                    3/5
starting post-remove package version mgmt for ospf
post-remove for ospf complete
starting pre-remove package version mgmt for eigrp
```

```
pre-remove for eigrp complete
  Erasing    : rip-2.0.0.0-9.2.1.lib32_n9000

                                    4/5
starting post-remove package version mgmt for eigrp
post-remove for eigrp complete
starting pre-remove package version mgmt for rip
pre-remove for rip complete
  Erasing    : bgp-2.0.0.0-9.2.1.lib32_n9000

                                    5/5
starting post-remove package version mgmt for rip
post-remove for rip complete
starting pre-remove package version mgmt for bgp
pre-remove for bgp complete

Removed:
  bgp.lib32_n9000 0:2.0.0.0-9.2.1       eigrp.lib32_n9000 0:2.0.0.0-9.2.1
isis.lib32_n9000 0:2.0.0.0-9.2.1      ospf.lib32_n9000 0:2.0.0.0-9.2.1      rip.lib32_n9000
 0:2.0.0.0-9.2.1

Complete!
```

Finding Repositories

This command lists the repositories that the switch has along with the number of RPMs it has to those repositories.

```
bash-4.3# yum repolist all

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo

                     | 1.1 kB     00:00 ...
localdb

                     |  951 B     00:00 ...
patching

                     |  951 B     00:00 ...
thirdparty

                     |  951 B     00:00 ...
repo id
          repo name
                                  status
groups-repo
          Groups-RPM Database
                                  enabled: 37
localdb
          Local RPM Database
                                  enabled:  6
patching
          Patch-RPM Database
                                  enabled:  0
thirdparty
          Thirdparty RPM Database
                                  enabled:  0
open-nxos
          open-nxos
```

```
                                    disabled
repolist: 43
```

Finding the Installed YUM Version

See the following example for listing the installed YUM version:

```
yum --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 the NX-OS CLI to the YUM Commands

See the following table for mapping the NX-OS CLI to the YUM commands:

Table 6: Patching Command Reference

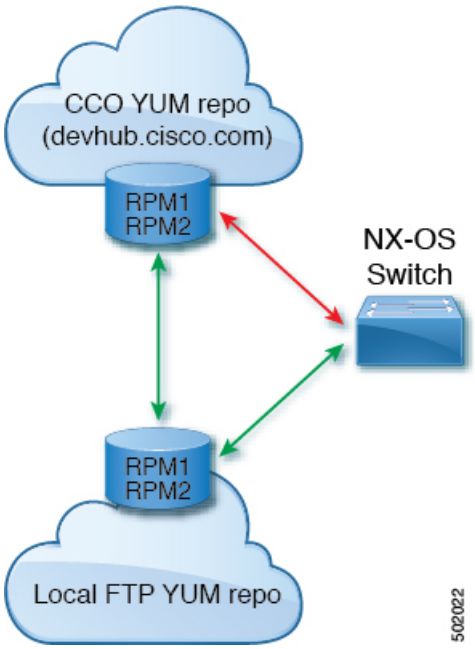
NX-OS CLI Commands	YUM Commands
show install inactive	**yum list --patch-only available**
show install active	**yum list --patch-only installed**
show install committed	**yum list --patch-only committed**
show install packages	**yum list --patch-only**
show install pkg-info	**yum info --patch-only**
show install log	**yum history --show-patch-log** where log_cmd: • opid= - Log that is specific to an operation ID. • last - Shows the latest operation log. • reverse – Shows the log in reverse order. • detail – Show detailed log. • from= - Shows logging from a specific operation ID.
clear install log	**yum history --clear-patch-log=** where clear_log_cmd: • all - Clears the complete log. • - Clears the logs above this operation ID.

NX-OS CLI Commands	YUM Commands
install add	**yum install --add bootflash:/**
install remove	**yum install --remove**
install remove inactive	**yum install --remove all**
install activate	**yum install --no-persist --nocommit** **Note** By default, all packages are activated and committed.
install deactivate	**yum erase --nocommit** **Note** By default, all packages are de-activated and committed.
install commit	**yum install --commit**
Install commit	**yum install --commit all**

Configuring an FTP server and Setting 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 Cisco NX-OS switch to reach the FTP server as outlined in the following illustration.

Figure 3: Configuring an FTP server and Setting up a Local FTP YUM Repository



Note For Cisco NX-OS Release 9.2(1), visit https://devhub.cisco.com/artifactory/open-nxos/9.2.1/ for Cisco **open-nxos** repository.

Creating 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

	Command or Action	Purpose
Step 1	**yum install vsftpd**	Installs vsftpd, an FTP server.
Step 2	**systemctl start vsftpd**	Starts the FTP Server.
Step 3	**systemctl status vsftpd**	Checks the status of the FTP Server.
Step 4	**firewall-cmd --zone=public --permanent --add-port=21/tcp**	Allows access to the FTP services from the external systems and opens port 21.
Step 5	**firewall-cmd --zone=public --permanent --add-service=ftp**	Adds the FTP service.
Step 6	**firewall-cmd --reload**	Reloads the server.

	Command or Action	Purpose
Step 7	**wget ftp://** *<ip of FTP server> /test.txt*	Hosts a file in the FTP server (for example, test.txt) and attempts Wget of that file. **Note** Note that **/var/ftp/** is the default home directory of the FTP server.

Creating a Local FTP YUM Repository

Complete the following steps to synchronize the external repository RPMs to the FTP server and create a local FTP YUM repository:

Procedure

	Command or Action	Purpose

Step 1	**cat /etc/yum.repos.d/local.repo**
	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`
		Creates a repository file under **/etc/yum.repos.d/**, for example, creates **local.repo** repository and adds the base URL.

Step 2	bash-4.3#**yum repolist**
	Example:
	`bash-4.3# yum 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`
		Checks the reachability of the repository.

Step 3	**nohup reposync -r** *<repo-name mentioned in the local.repo>* **-p** *<directory path to sync>***&** **Example:** `nohup reposync -r localrepo -p /var/ftp/ &` This command creates a directory with the name **local.repo** inside **/var/ftp/** and downloads all	Synchronizes all the packages from the external repository to the FTP server home directory.

	Command or Action	Purpose
	the packages from **devhub.cisco.com** to the directory.	
Step 4	**tail -f nouhup.out**	Checks the status of the synchronization.

Configuring a Switch to Reach an FTP Server

Complete the following steps to configure a switch to reach an FTP server:

Procedure

	Command or Action	Purpose
Step 1	**run bash sudo su**	Logs in as a sudo user.
Step 2	**ip netns exec management ping** <*ip_address*>	Checks the reachability of the FTP server address from the switch using the **ping** command.

Step 3	**cat /etc/yum/repos.d/ftp.repo**
	Example:
	`bash-4.3# cat /etc/yum/repos.d/ftp.repo`
	`[ftp]`
	`name=ftp`
	`baseurl=ftp://10.232.44.34/localrepo/`
	`enabled=1`
	`gpgcheck=0`
	`sslverify=0`
		Creates a repository file on the switch with the FTP server address as the URL.

Step 4	**ip netns exec management bash**	Uses the Bash shell prompt.

Step 5	**yum repolist**
	Example:
	`bash-4.3# yum 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`
		Checks the reachability of newly created repository.

Step 6	**yum list available**	Lists the available packages in the new repository.

Creating 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. See RBAC configuration guidelines for the same.

Compacting Cisco NX-OS Software Images

Cisco NX-OS software image compaction reduces the size of the image file before completing a copy request. Use SCP, HTTP, or HTTPS as the source and bootflash or USB as the destination. The following example uses SCP and bootflash:

```
switch# copy scp://user@scpserver.cisco.com//download/nxos.9.3.5.bin
bootflash:nxos.9.3.5.bin compact vrf management use-kstack

user1@10.65.42.196's password:
nxos.9.3.5.bin 100% 1887MB 6.6MB/s 04:47
Copy complete, now saving to disk (please wait)...
Copy complete.
```

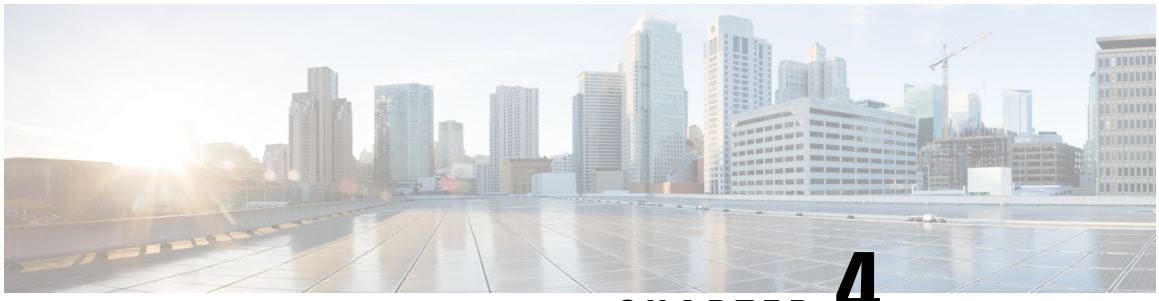
The **compact** keyword compacts the NX-OS image before copying the file to the supervisor module.

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 Cisco Nexus 9300-series platform switches.



CHAPTER 4

Upgrading the Cisco NX-OS Software Using Fast Reload

This chapter describes how to upgrade the Cisco NX-OS software on a switch using fast reload. It contains the following sections:

About Fast Reload

The fast reload feature enables you to reboot the switch faster than with the **reload** command. You can also use fast reload to upgrade the software on the switch.

During a fast reload, the NXOS software image that runs on the CPU reloads the new image and runs it without a CPU or firmware reset. Although traffic is briefly disrupted during a fast reload, this feature enables the switch to reload faster than during a cold reboot.

You can use fast reload in a non-interruptive mode, which runs the installation process without any prompts, or with BGP graceful restart for BGP-compatible peers.

Fast Reload Sequence of Events

The following sequence of events occurs when you perform a fast reload using the **fast-reload** command:

1. The switch loads the NXOS software image and upgrades the kernel. All applications undergo a stateless cold reboot and are restarted through the startup configuration.
2. The control plane is disrupted. During this disruption, all control protocol communication stops. The control plane disruption is less than 90 seconds.
3. After the control plane disruption, all control plane applications undergo a stateless cold reboot and do not retain their state. The new configuration is applied when the switch reloads.

4. The data plane is disrupted. The data plane disruption is less than 30 seconds.
5. On the forwarding plane, all links become unavailable, and the data plane does not retain its state after reload. Traffic forwarding is resumed within 30 seconds.

Prerequisites for Fast Reload

Fast reload has the following prerequisites:

- Verify that sufficient space is available in the bootflash.
- To allow a fast reload, make sure that Link Aggregation Control Protocol (LACP) fast timers are not configured.

Guidelines and Limitations for Fast Reload

Fast reload has the following guidelines and limitations:

- Only the Cisco Nexus 3164Q, 3264C-E, and 92304QC switches support fast reload.
- Fast reload is supported from Cisco NX-OS Release 7.0(3)I7(4), 7.0(3)I7(5), or 9.2(x) to a Cisco NX-OS 9.2(x) release.
- Using fast reload to downgrade the Cisco NX-OS software is not supported. To downgrade the software, use the **install all** command.
- Ensure that everyone who has access to the switch or the network is not configuring the switch or the network during this time. You cannot configure the switch during a fast reload. Use the **show configuration session summary** command to verify that you have no active configuration sessions.
- Save, commit, or discard any active configuration sessions before performing a fast reload. Any active configuration sessions will be deleted without warning.
- Make any topology changes (such as Spanning Tree Protocol changes) before you perform a fast reload. However, do not make changes to the Layer 2 and routing topologies.
- Do not insert or remove any fans or power supplies during a fast reload.
- Schedule the fast reload when your network is stable and steady.
- BIOS upgrades are not supported by fast reload.
- The CPU stops responding between control plane disruption and data plane disruption.
- The **copy** *configuration-file* **startup-config** command is supported with fast reload for a limited set of configurations.
- Ensure that the username is specified in the configuration file before you perform a **copy** *configuration-file* **startup-config** followed by the **fast-reload** or **reload** command. Otherwise, you will not be able to the access the switch and will need to complete the password recovery procedure to get the system back online. For information on the password recovery procedure, see the "Power Cycling the Device to Recover the Administrator Password" section in the Cisco Nexus 9000 Series NX-OS Troubleshooting Guide.

- Fast reload currently supports the following two configuration profiles:

  Fast-reload profile 1

  - 48 Layer 2 links
  - 1 VLAN and SVI
  - 16 Layer 3 ECMP links
  - 6000 IPv4 LPM routes, 3000 IPv6 LPM routes, 200 IPv4 VIPs, and 200 IPv6 VIPs
  - 2000 IPv4 ARPs and 2000 IPv6 neighbor discovery (ND)

  Fast-reload profile 2

  - 24 Layer 2 port channels with two members each
  - 24 VLANs and SVIs
  - 8 Layer 3 port-channel ECMPs with two members each
  - 6000 IPv4 LPM routes, 3000 IPv6 LPM routes, 50 IPv4 VIPs, and 50 IPv6 VIPs
  - 2000 IPv4 ARPs and 2000 IPv6 neighbor discovery (ND)

Performing a Fast Reload and Upgrading the Cisco NX-OS Software

You can use this procedure to reboot the device faster than during a cold reboot. If you specify a software image, the software on the switch is upgraded.

Before you begin

Ensure that you have a working software image and that you have analyzed the impact of the fast reload operation.

Procedure

Step 1 Log in to the switch.

Step 2 Use the **fast-reload** [**save-config**] [**trigger-gr**] [**nxos bootflash:***nxos-image-name*] [**non-interruptive**] command to perform a fast reload.

Example:

```
switch# fast-reload nxos bootflash:nxos.9.2.1.bin
```

The following options are available:

- **save-config**—Ensures that subsequent fast reload operations use the new NXOS software image as the boot variable. If you do not use the **save-config** option, this command does not save the boot variable, and subsequent fast reload operations use the old software image as the boot variable.

- **trigger-gr**—By default, the fast reload feature requires Border Gateway Protocol (BGP) peers to be graceful restart capable. The **trigger-gr** option adds support for restarts with aggressive timers.
- **nxos bootflash:***nxos-image-name*—Specifies the name of the NXOS software image. Make sure to specify a software version that supports the fast reload feature.
- **non-interruptive**—Performs a fast reload without any prompts. Before you choose this option, verify that fast reload works on your system because this option skips all error and sanity checks.

Example

This example shows how to use fast reload to upgrade the Cisco NX-OS software on the switch:

```
switch# fast-reload nxos bootflash:nxos.9.2.1.bin
```

Saving the Configuration with Fast Reload

This table shows the expected behavior for saving the configuration with different variations of the **fast-reload** command:

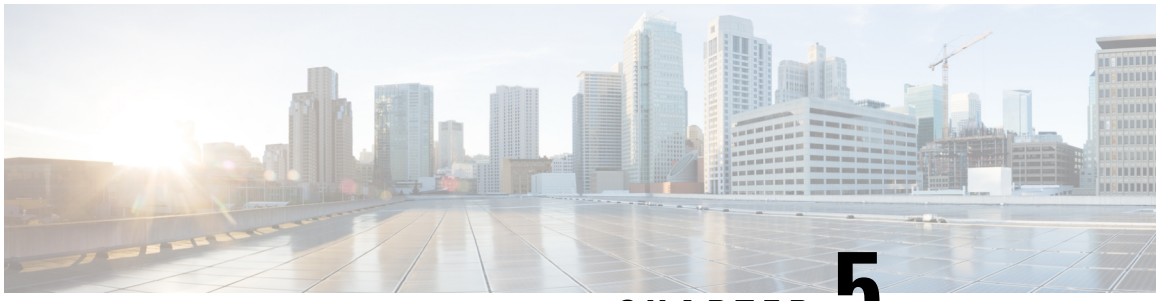
Command	Expected Behavior
fast-reload	Prompts you if there is a configuration change and performs a **copy running-config startup-config** based on your response.
fast-reload non-interruptive	No prompts appear, and the configuration is not saved. You need to save the configuration using the **save-config** option or the **copy running-config startup-config** command.
fast-reload nxos bootflash:*nxos-image-name* [**non-interruptive** | **trigger-gr**]	Implicitly performs a **copy running-config startup-config**, even if the image is the same image.
copy *configuration-file* **startup-config** **fast-reload**	After bootup, implicitly performs a **copy** *configuration-file* **startup-config** and sets the boot variable to the booted image.
copy *configuration-file* **startup-config** **fast-reload nxos bootflash:***nxos-image-name*	After bootup, implicitly sets the boot variable to the specified image and performs a **copy** *configuration-file* **startup-config**.

Note Ensure that the username is specified in the configuration file before you perform a **copy** *configuration-file* **startup-config** followed by the **fast-reload** or **reload** command. Otherwise, you will not be able to the access the switch and will need to complete the password recovery procedure to get the system back online. For information on the password recovery procedure, see the "Power Cycling the Device to Recover the Administrator Password" section in the Cisco Nexus 9000 Series NX-OS Troubleshooting Guide.

Additional References

Related Documents

Related Topic	Document Title
reload command	Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide



CHAPTER 5

Converting from Cisco NX-OS to ACI Boot Mode and from ACI Boot Mode Back to Cisco NX-OS

This chapter describes how to convert a Cisco Nexus 9000 Series switch from Cisco NX-OS to Cisco Application Centric Infrastructure (ACI) boot mode. It contains the following sections:

Converting to ACI Boot Mode

You can convert any Cisco Nexus 9000 Series switch from Cisco NX-OS to ACI boot mode.

Note You cannot convert a Cisco Nexus 3164Q or 31128PQ switch to ACI boot mode.

Note Use this procedure to convert a Cisco Nexus 9000 Series switch to ACI boot mode. If you are converting to ACI boot mode from an earlier Cisco NX-OS release, follow the instructions in the Nexus 9000 Standalone (NXOS) to ACI Conversion document.

Before you begin

Verify whether your switch hardware is supported in ACI boot mode by checking the "Supported Hardware" section of the Release Notes for Cisco Nexus 9000 Series ACI-Mode Switches. For example, line cards are not compatible between Cisco 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/retry mechanism before powering down the unsupported modules, which can cause delays in the conversion process.

For dual-supervisor systems, use the **show module** command to make sure that the standby supervisor module is in the ha-standby state.

Verify that the Application Policy Infrastructure Controller (APIC) is running Release 1.0(2j) or a later release.

Make sure that the ACI image is 11.0(2*x*) or a later release.

Use the **show install all impact epld** *epld-image-name* command to verify that the switch does not require any EPLD image upgrades. If any upgrades are required, follow the instructions in the Cisco 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
```

Cisco NX-OS filenames begin with "nxos".

Step 2 Follow these steps to copy the ACI image from the APIC:

a) Set the IP address on the mgmt0 interface of the switch to allow connectivity between this interface and the APIC.

b) Enable SCP services on the switch.

Example:

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

c) From the APIC CLI, use SCP to copy the firmware image from the APIC to the active supervisor module on the switch.

Example:

```
admin@apic1:aci> scp -r /firmware/fwrepos/fwrepo/switch-image-name
admin@switch-ip-address:switch-image-name
```

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

Example:

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

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

a) Configure the switch to not boot from Cisco NX-OS.

Example:

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

b) Save the configuration.

Example:

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

Note You must run the **copy running-config startup-config** command prior to booting the ACI image. Do not run it after you enter the **boot aci** command.

c) Boot the active and standby supervisor modules with the ACI image.

Example:

```
switch(config)# boot aci bootflash:aci-image-name
```

Caution Do not enter the **copy running-config startup-config** command after the **boot aci** command. If you do, the switch will go 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 "Cisco 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 Cisco Nexus 9000 Series switches that were shipped prior to May 2014.

To run this command, contact TAC.

What to do next

See the ACI and APIC documentation to configure and operate your switch in ACI mode: http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html.

Converting 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 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
```

Converting Back to Cisco NX-OS

You can convert a Cisco Nexus 9000 series switch from ACI boot mode back to Cisco NX-OS.

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 Configure the boot process to stop at the switch(boot)# prompt.

Example:

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

Step 4 Boot the active supervisor module with the Cisco NX-OS image.

Example:

```
loader> boot nxos.9.2.3.bin
```

Note If the Cisco NX-OS image mentioned in the bootvariable 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 Cisco NX-OS releases and Cisco Nexus 9000 Series switches, the following error message appears:

```
!!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 Cisco NX-OS prompt and skips the switch(boot)# prompt.

Step 5 Restores the switch's file system partitioning to the default settings. The bootflash filesystem is reset to Cisco NX-OS partitioning, and the Cisco NX-OS image is deleted.

Example:

```
switch(boot)# init system
```

Step 6 Completes the upload of the nx-os image file.

Example:

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

Note For some Cisco Nexus 9000 series switches, the device does not load with the normal Cisco 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 an USB method.

- For TFTP method - First assign a IP address and gateway to the device using the **set ip** *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 1.1.1.2 255.255.255.255.0
  loader>set gw 1.1.1.1
  ```

  Then use the **tftp** command to load the image.

  ```
  loader> boot tftp://<tftp server ip>/<nxos-image-name>
  ```

- For USB method - Mount the USB on the switch and execute the **dir** coammnd 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
  ```

Once you boot the Cisco NX-OS image, the device will load as an NX-OS switch and you can continue with the remaining steps.

Step 7 Re-copy the Cisco NX-OS image into bootflash: and set the appropriate boot variables to ensure that the system boots the Cisco 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, which could take approximately 15 to 20 minutes.

File system differences between ACI and Cisco NX-OS require a one-time reformatting change during the ACI to Cisco NX-OS conversion. Subsequent reloads with the Cisco NX-OS image will be faster.

Step 9 Verify that the active supervisor module and the system controllers are in the active state.

Example:

```
switch# show module
```

Step 10 For dual-supervisor systems, follow Steps 3 through 6 on the standby supervisor.

Step 11 Log in to the switch and verify that it is running Cisco NX-OS software.

Using SCP on the ACI Shell to Load NX-OS Image into Bootflash

Use this task if you have a switch in ACI mode and must convert it to NX-OS mode, but are unable to perform a TFTP boot and the USB option is not available. The following steps describe how to boot the switch on ACI mode, configure the management port, and copy the software image to the bootflash partition.

The leaf switch boots into ACI mode in fabric discovery state.

Procedure

Step 1 Log in with the username "admin" and no password. The command prompt appears:

```
#
```

Step 2 **configure terminal**

Example:

```
# configure terminal
(config)#
```

Step 3 **interface mgmt 0**

Example:

```
(config)# interface mgmt 0
(config-if)#
```

Step 4 **ip address** *ipv4−address* { [*/length*] | [*subnet−mask*] }

Example:

```
(config-if)# ip address 10.1.1.20/24
(config-if)#
```

Step 5 **no shutdown**

Example:

```
(config-if)# no shutdown
(config-if)#
```

Step 6 **exit**

Example:

```
(config-if)# exit
(config)#
```

Step 7 **vrf context management**

Example:

```
(config)# vrf context management
(config-vrf)#
```

Step 8 **ip route** *ipv4−address* { [*/length*] | [*subnet−mask*] } *default-gw-ipv4−address* { [*/length*] | [*subnet−mask*] }

Example:

```
(config-vrf)# ip route 0.0.0.0/0 10.1.1.30/24
(config-vrf)#
```

Step 9 **end**

Example:

```
(config-vrf)# end
#
```

Step 10 **cd /bootflash**

Example:

```
# cd /bootflash
#
```

Step 11 **scp** *username* **@** *scp-server-ip-address* **:** *nxos-image*

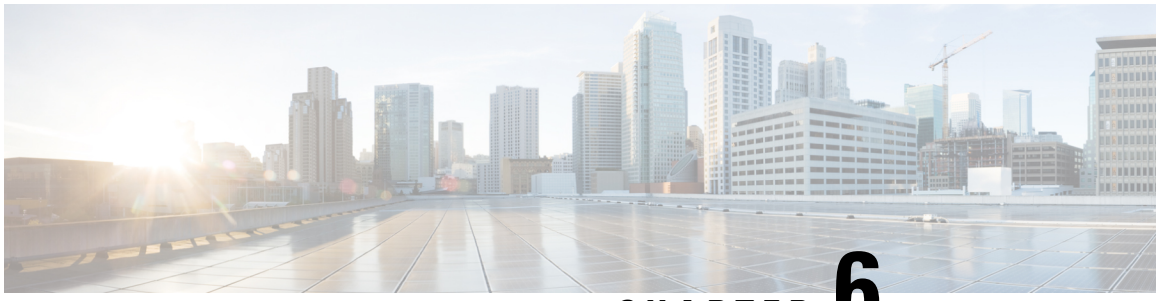
Example:

```
# scp user1@10.1.1.25:n9000-dk9.7.0.3.I1.1.bin
#
```

Step 12 Reload the switch, break into the loader prompt, and follow the steps to load the NX-OS image as shown in the previous Converting Back to Cisco NX-OS the procedure. The newly copied software image appears in the bootflash.

Example

```
# configure terminal
(config)# interface mgmt 0
(config-if)# ip address 10.1.1.20/24
(config-if)# no shutdown
(config-if)# exit
(config)# vrf context management
(config-vrf)# ip route 0.0.0.0/0 10.1.1.30/24
(config-vrf)#end
# cd /bootflash
# scp user1@10.1.1.25:n9000-dk9.7.0.3.I1.1.bin
```



CHAPTER 6

Migrating Switches in a vPC Topology

This chapter describes how to migrate from one pair of switches to another in a vPC topology. It contains the following sections:

vPC Forklift Upgrade

In a vPC topology, you can migrate from a pair of Cisco Nexus 9000 Series switches to a different pair of Cisco Nexus 9000 Series switches. For example, you might migrate from a pair of Cisco Nexus 9508 vPC peer nodes to a pair of Cisco 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, Cisco NX-OS Releases 7.0(3)F3(3c) and 7.0(3)F3(4) are not compatible with Cisco NX-OS Release 9.2(x) for vPC peer switches. Both vPC peers must be upgraded simultaneously to Cisco 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 Cisco 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.

1. Switch A and B are running a Cisco NX-OS release. Switch A is the primary switch, and switch B is the secondary switch. Use the **copy r s** command on both switches.

```
primary_switch# show vpc role
vPC Role status
----------------------------------------------------
```

```
vPC role : primary
vPC system-mac : 00:23:04:ee:be:64
vPC system-priority : 32667
vPC local system-mac : 70:df:2f:eb:86:1f
vPC local role-priority : 90
vPC peer system-mac : 70:df:2f:eb:1c:ab
vPC peer role-priority : 100
primary_switch#

secondary_switch# show vpc role
vPC Role status
----------------------------------------------------
vPC role : secondary
vPC system-mac : 00:23:04:ee:be:64
vPC system-priority : 32667
vPC local system-mac : 70:df:2f:eb:1c:ab
vPC local role-priority : 100
vPC peer system-mac : 70:df:2f:eb:86:1f
vPC peer role-priority : 90
secondary_switch#


primary_switch# copy r s v
[########################################] 100%
Copy complete.

secondary_switch# copy r s v
[########################################] 100%
Copy complete.
```

2. Bring down the peer link (PL) on the primary switch. The secondary switch brings down its vPC legs.

```
primary_switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
primary_switch(config)# int port-channel 100
primary_switch(config-if)# shutdown

Reload the secondary switch with Release 9.2.1 image (change bootvar /reload)

secondary_switch(config)# boot nxos nxos.9.2.1.bin
Performing image verification and compatibility check, please wait....
secondary_switch(config)#
secondary_switch(config)# copy r s v
[########################################] 100%
Copy complete.


secondary_switch# reload
This command will reboot the system. (y/n)? [n] y


After reload
---------------------
secondary_switch# show vpc
Legend:
(*) - local vPC is down, forwarding via vPC peer-link
vPC domain id : 100
Peer status : peer link is down
vPC keep-alive status : peer is alive
Configuration consistency status : failed
Per-vlan consistency status : success
Configuration inconsistency reason: Consistency Check Not Performed
Type-2 inconsistency reason : Consistency Check Not Performed
```

```
vPC role : none established
Number of vPCs configured : 20
Peer Gateway : Enabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Disabled (due to peer configuration)
Auto-recovery status : Disabled
Delay-restore status : Timer is off.(timeout = 90s)
Delay-restore SVI status : Timer is off.(timeout = 10s)
Operational Layer3 Peer-router : Disabled
vPC Peer-link status
---------------------------------------------------------------------
id Port Status Active vlans
-- ---- ------ --------------------------------------------------
1 Po100 down -

secondary_switch#

primary_switch(config-if)# show vpc
Legend:
(*) - local vPC is down, forwarding via vPC peer-link
vPC domain id : 100
Peer status : peer link is down
vPC keep-alive status : peer is alive
Configuration consistency status : success
Per-vlan consistency status : success
Type-2 consistency status : success
vPC role : primary
Number of vPCs configured : 20
Peer Gateway : Enabled
Peer gateway excluded VLANs : -
Dual-active excluded VLANs and BDs : -
Graceful Consistency Check : Enabled
Auto-recovery status : Enabled, timer is off.(timeout = 240s)
Operational Layer3 Peer-router : Disabled
vPC Peer-link status
---------------------------------------------------------------------
id Port Status Active vlans
-- ---- ------ --------------------------------------------------
1 Po100 down -
```

3. Configure vPC auto-recovery under the vPC domain on the secondary switch. Enable **vpc upgrade** (exec command).

```
secondary_switch(config)# vpc domain 100
secondary_switch(config-vpc-domain)# auto-recovery
secondary_switch(config-vpc-domain)# end

secondary_switch# show running-config vpc
!Command: show running-config vpc
!Running configuration last done at: Wed May 16 06:34:10 2018
!Time: Wed May 16 06:34:14 2018
version 9.2(1) Bios:version 01.11
feature vpc
vpc domain 100
peer-switch
role priority 100
peer-keepalive destination 10.1.31.30 source 10.1.31.29
delay restore 90
peer-gateway
auto-recovery
ipv6 nd synchronize
ip arp synchronize
```

```
interface port-channel100
vpc peer-link
interface port-channel2001
vpc 101


secondary_switch# show vpc upgrade
vPC upgrade : TRUE
SVI Timer : 0
Delay Restore Timer : 0
Delay Orphan Port Timer : 0
secondary_switch#

secondary_switch# show vpc upgrade   >> Hidden command
vPC upgrade : FALSE
SVI Timer : 10
Delay Restore Timer : 90
Delay Orphan Port Timer : 0


secondary_switch# vpc upgrade   >> Hidden command
```

4. After Layer 3 routes are learned on the secondary switch, reload the primary switch with the new release image. The secondary switch takes over the primary role and brings up its vPC legs in approximately 5 seconds.

```
primary_switch(config)# show boot
Current Boot Variables:
sup-1
NXOS variable = bootflash:/nxos.9.2.1.bin
No module boot variable set
Boot Variables on next reload:
sup-1
NXOS variable = bootflash:/nxos.9.2.1.bin

No module boot variable set
primary_switch(config)# end

primary_switch# show boot
Current Boot Variables:
sup-1
NXOS variable = bootflash:/nxos.9.2.1.bin
No module boot variable set
Boot Variables on next reload:
sup-1
NXOS variable = bootflash:/nxos.9.2.1.bin

No module boot variable set
primary_switch# reload
This command will reboot the system. (y/n)? [n] y

secondary_switch# show vpc
Legend:
(*) - local vPC is down, forwarding via vPC peer-link
vPC domain id : 100
Peer status : peer link is down
vPC keep-alive status : peer is not reachable through peer-keepalive
Configuration consistency status : failed
Per-vlan consistency status : success
Configuration inconsistency reason: Consistency Check Not Performed
Type-2 inconsistency reason : Consistency Check Not Performed
vPC role : primary
```

```
Number of vPCs configured : 20
Peer Gateway : Enabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Disabled (due to peer configuration)
Auto-recovery status : Enabled, timer is off.(timeout = 240s)
Delay-restore status : Timer is off.(timeout = 0s)
Delay-restore SVI status : Timer is off.(timeout = 0s)
Operational Layer3 Peer-router : Disabled
vPC Peer-link status
---------------------------------------------------------------------
id Port Status Active vlans
-- ---- ------ --------------------------------------------------
1 Po100 down -
vPC status
```

5. When the primary switch comes back up, the peer link on it is operationally up.

```
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.