



## **Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide, Release 10.1(x)**

**First Published:** 2021-02-16

**Last Modified:** 2021-10-13

### **Americas Headquarters**

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

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

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

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at [www.cisco.com/go/offices](http://www.cisco.com/go/offices).

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

© 2021 Cisco Systems, Inc. All rights reserved.



## CONTENTS

---

### PREFACE

|  |            |
|--|------------|
| <b>Preface</b>   | <b>vii</b> |
| Audience   | vii        |
| Document Conventions                                       | vii        |
| Related Documentation for Cisco Nexus 9000 Series Switches | viii       |
| Documentation Feedback                                     | viii       |
| Communications, Services, and Additional Information       | viii       |
| Cisco Bug Search Tool                                      | ix         |
| Documentation Feedback                                     | ix         |

---

### CHAPTER 1

|                                    |          |
|------------------------------------|----------|
| <b>New and Changed Information</b> | <b>1</b> |
| New and Changed Information        | 1        |

---

### CHAPTER 2

|  |          |
|--|----------|
| <b>Upgrading or Downgrading the Cisco Nexus 9000 Series NX-OS Software</b> | <b>3</b> |
| About the Software Image   | 3        |
| About ISSU   | 4        |
| Recommendations for Upgrading the Cisco NX-OS Software                     | 6        |
| Prerequisites for Upgrading the Cisco NX-OS Software                       | 6        |
| Prerequisites for Downgrading the Cisco NX-OS Software                     | 7        |
| Cisco NX-OS Software Upgrade Guidelines                                    | 7        |
| ISSU Platform Support  | 14       |
| Cisco NX-OS Software Downgrade Guidelines                                  | 17       |
| Upgrade Paths  | 19       |
| Upgrade Patch Instructions   | 19       |
| Configuring Enhanced ISSU  | 28       |
| Upgrading the Cisco NX-OS Software   | 30       |
| Upgrade Process for vPCs   | 34       |

|  |    |
|--|----|
| Upgrade Process for a vPC Topology on the Primary Switch   | 34 |
| Upgrade Process for a vPC Topology on the Secondary Switch | 35 |
| Downgrading to an Earlier Software Release                 | 35 |
| Cisco NX-OS Upgrade History                                | 37 |

**CHAPTER 3****Optionality in Cisco NX-OS Software 39**

|   |    |
|---|----|
| Optionality in Cisco NX-OS Software                         | 39 |
| Using Modular Packages                                      | 40 |
| Booting the NX-OS Image in Base or Full Mode                | 41 |
| Information About RPMs                                      | 42 |
| Format of the RPM   | 42 |
| Optional RPMs and Their Associated Features                 | 43 |
| Guidelines for NX-OS Feature RPM Installation               | 44 |
| Guidelines for Third-Party RPM Installation                 | 46 |
| Using Install CLIs for Feature or Third-Party RPM Operation | 46 |
| Using Install CLIs for Digital Signature Support            | 48 |
| Querying All Installed RPMs                                 | 49 |
| Installing the RPMs Using One Step Procedure                | 50 |
| Installing the RPMs Using Two Steps Procedure               | 51 |
| Upgrading the RPMs Using One Step                           | 52 |
| Downgrading the RPMs  | 53 |
| Uninstalling the RPMs                                       | 54 |
| Removing the RPMs   | 55 |
| Information About DNF Commands                              | 56 |
| Performing Package Operations Using the DNF Commands        | 56 |
| Finding the Base Version RPM of the Image                   | 56 |
| Checking the List of the Installed RPMs                     | 57 |
| Getting Details of the Installed RPMs                       | 57 |
| Installing the RPMs   | 58 |
| Upgrading the RPMs  | 60 |
| Downgrading the RPMs  | 62 |
| Deleting the RPMs   | 64 |
| Support for DNF Groups                                      | 65 |
| Finding Repositories  | 68 |

|  |    |
|--|----|
| Finding the Installed DNF Version  | 69 |
| Mapping the NX-OS CLI to the DNF Commands                                    | 69 |
| Configuring an FTP server and Setting up a Local FTP YUM Repository          | 70 |
| Creating an FTP Server on Red Hat Enterprise Linux 7 (RHEL7) Virtual Machine | 71 |
| Creating a Local FTP YUM Repository  | 71 |
| Configuring a Switch to Reach an FTP Server                                  | 73 |
| Creating User Roles for Install Operation                                    | 74 |
| Compacting Cisco NX-OS Software Images                                       | 74 |

---

|                  |  |           |
|------------------|--|-----------|
| <b>CHAPTER 4</b> | <b>Converting from Cisco NX-OS to ACI Boot Mode and from ACI Boot Mode Back to Cisco NX-OS</b> | <b>77</b> |
|                  | Converting to ACI Boot Mode  | 77        |
|                  | Converting a Replacement Standby Supervisor to ACI Boot Mode                                   | 79        |
|                  | Converting Back to Cisco NX-OS   | 80        |
|                  | Using SCP on the ACI Shell to Load NX-OS Image into Bootflash                                  | 83        |

---

|                  |   |           |
|------------------|---|-----------|
| <b>CHAPTER 5</b> | <b>Migrating Switches in a vPC Topology</b>                           | <b>85</b> |
|                  | vPC Forklift Upgrade  | 85        |
|                  | vPC Upgrade and Downgrade Procedure for Nexus 9000 -R series switches | 85        |





## Preface

---

This preface includes the following sections:

- [Audience, on page vii](#)
- [Document Conventions, on page vii](#)
- [Related Documentation for Cisco Nexus 9000 Series Switches, on page viii](#)
- [Documentation Feedback, on page viii](#)
- [Communications, Services, and Additional Information, on page viii](#)

## 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   |
|---------------|---|
| <b>bold</b>   | Bold text indicates the commands and keywords that you enter literally as shown.  |
| <i>Italic</i> | 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   |
|-----------------|---|
| <i>variable</i> | 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   |
|-----------------------------|---|
| <code>screen font</code>    | Terminal sessions and information the switch displays are in screen font.                                 |
| <b>boldface screen font</b> | Information that you must enter is in boldface screen font.   |
| <i>italic screen font</i>   | 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](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](mailto: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 DevNet](#).
- 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 gateway to the Cisco bug-tracking system, which maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. The BST provides you with detailed defect information about your products and software.

## Documentation Feedback

To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.





# 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 10.1(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 10.1(x)* and tells you where they are documented.

**Table 1: New and Changed Features for Cisco NX-OS Release 10.1(x)**

| Feature                                       | Description  | Changed in Release | Where Documented  |
|---|--|--------------------|---|
| CoPP  | Added support to Cisco NX-OS N9K-X9624D-R2 and N9K-C9508-FM-R2 platform switches   | 10.1(2)            | <a href="#">Cisco NX-OS Software Upgrade Guidelines, on page 7</a>  |
| RACL  | Added support to Cisco NX-OS N9K-X9624D-R2 and N9K-C9508-FM-R2 platform switches   | 10.1(2)            | <a href="#">Cisco NX-OS Software Upgrade Guidelines, on page 7</a>  |
| Redesign of the <b>install deactivate</b> CLI | Support for options to either downgrade to the base version of RPM or to uninstall RPM.  | 10.1(1)            | <a href="#">Using Install CLIs for Feature or Third-Party RPM Operation, on page 46</a><br><a href="#">Uninstalling the RPMs, on page 54</a>                        |
| Third-party RPM validation                    | Introduced a restriction on installation of unsigned or non-Cisco signed third-party RPMs, with an option to bypass the restriction. | 10.1(1)            | <a href="#">Guidelines for Third-Party RPM Installation , on page 46</a><br><a href="#">Using Install CLIs for Feature or Third-Party RPM Operation, on page 46</a> |

| <b>Feature</b>                    | <b>Description</b>  | <b>Changed in Release</b> | <b>Where Documented</b>  |
|-----------------------------------|---|---------------------------|--|
| Enhanced ISSU with FC/FCoE        | Support for Enhanced ISSU on FC/FCoE switch mode.         | 10.1(1)                   | <a href="#">Cisco NX-OS Software Upgrade Guidelines, on page 7</a> |
| Standard ISSU                     | Support for Standard ISSU on 9300-GX platform switches.   | 10.1(1)                   | <a href="#">ISSU Platform Support, on page 14</a>                  |
| 64-bit Cisco NX-OS software image | Introduced a new 64-bit image of the Cisco NX-OS software | 10.1(1)                   | <a href="#">About the Software Image, on page 3</a>                |



## 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, on page 3](#)
- [About ISSU, on page 4](#)
- [Recommendations for Upgrading the Cisco NX-OS Software, on page 6](#)
- [Prerequisites for Upgrading the Cisco NX-OS Software, on page 6](#)
- [Prerequisites for Downgrading the Cisco NX-OS Software, on page 7](#)
- [Cisco NX-OS Software Upgrade Guidelines, on page 7](#)
- [ISSU Platform Support, on page 14](#)
- [Cisco NX-OS Software Downgrade Guidelines, on page 17](#)
- [Upgrade Paths, on page 19](#)
- [Upgrade Patch Instructions, on page 19](#)
- [Configuring Enhanced ISSU, on page 28](#)
- [Upgrading the Cisco NX-OS Software, on page 30](#)
- [Upgrade Process for vPCs, on page 34](#)
- [Downgrading to an Earlier Software Release, on page 35](#)
- [Cisco NX-OS Upgrade History, on page 37](#)

## 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. Only this image is required to load the Cisco NX-OS operating system.

In Cisco NX-OS Release 10.1(1), 10.1(2) there are 32 and 64 bit images.

- The 32-bit Cisco NX-OS image file has the image filename that begins with "nxos" (for example, nxos.10.1.1.bin).
- The 64-bit Cisco NX-OS image file has the image filename that begins with "nxos64" (for example, nxos64.10.1.1.bin).




---

**Note** Beginning with Cisco NX-OS Release 10.1(x), only 9300-GX platforms support 64-bit image.

---

For 32-bit or 64-bit image support on respective platforms, see the relevant version of the Cisco Nexus 9000 Series Release Notes on [Cisco.com](http://Cisco.com).

The Cisco Nexus 9000 Series 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](http://Cisco.com) and generally include the ID number of the resolved defect in the filename (for example, n9000-dk10.1.1.CSCab00001.gbin). For more information on SMUs, see the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*.

---




---

**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 non-disruptive upgrade, for some switches. (See the [ISSU Platform Support, on page 14](#) for a complete list of supported platforms.)

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

Enhanced ISSUs are supported for some Cisco Nexus 9000 Series switches.

The following ISSU scenarios are supported:

- Performing standard ISSU on Top-of-Rack (ToR) switches with a single supervisor
- 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 are the NX-OS switches with single supervisors. Performing ISSU on the Cisco Nexus 9000 Series switches causes the supervisor CPU to reset and to load the new software version. After the CPU loads the updated version of the 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 Series switches do not see any traffic disruption. The control plane downtime during the ISSU process is approximately less than 120 seconds.

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



**Note** Enhanced ISSU may not be supported if there are any underlying kernel differences. The system will prompt the following message:

```
Host kernel is not compatible with target image. Full ISSU will be performed and control
plane will be impacted.
```

In effect, system will perform non-disruptive ISSU instead of enhanced ISSU.

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 non-disruptive downgrades, are not supported.

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

## Recommendations for Upgrading the Cisco NX-OS Software

Cisco recommends performing a Nexus Health and Configuration Check before performing an upgrade. The benefits include identification of potential issues, susceptible Field Notices and Security Vulnerabilities, missing recommended configurations and so on. For more information about the procedure, see [Perform Nexus Health and Configuration Check](#).

## 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 Nexus 9000 and 3000 Upgrade and ISSU 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:

- For a device that is running on Cisco Nexus Release 10.1(2), ND-ISSU is not supported if L2 sub-interfaces are configured.
- For ISSU compatibility for all releases, see the [Cisco Nexus 9000 and 3000 Upgrade and ISSU Matrix](#).
- Non-disruptive upgrade to 64-bit image is supported from Cisco NX-OS Release 9.3(9) onwards. See supported platforms in [Cisco Nexus 9000 and 3000 Upgrade and ISSU Matrix](#).
- Beginning with Cisco NX-OS Release 10.1(1), during the disruptive upgrade to the 64-bit image or a downgrade from 64-bit to 32-bit image, if feature ITD is enabled, refer to *Guidelines and Limitations for ITD* in the *Cisco Nexus 9000 Series NX-OS Intelligent Traffic Director Configuration Guide, Release 10.1(x)*, if the upgrade or downgrade proceeds with an ASIC reload.
- 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 upgrading from Cisco NX-OS Release 9.3(3) to Cisco NX-OS Release 9.3(6) or later, 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 the *Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide, Release 10.1(x)*.
- When you upgrade a Cisco Nexus 9000 device to Cisco NX-OS Release 10.1(x), if a QSFP port is configured with the manual breakout command and is using a QSA, the configuration of the interface Ethernet 1/50/1 is no longer supported and must be removed. To restore the configuration, you must manually configure the interface Ethernet 1/50 on the device.
- When redistributing static routes, Cisco NX-OS requires the **default-information originate** command to successfully redistribute the default static route starting in 7.0(3)I7(6).
- To perform an EPLD upgrade after an ISSU upgrade from Cisco NX-OS Release 7.x to Cisco NX-OS Release 9.3(x), before starting the EPLD upgrade, add the copy run start command.
- 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 10.1(x)*.
- When upgrading from Cisco NX-OS Release 9.2(2) or earlier releases to Cisco NX-OS Release 10.1(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.
- Beginning with Cisco NX-OS Release 10.1(1), ISSU is supported on FC/FCoE switch mode on Cisco Nexus 93360YC-FX2. For more information about the FC/FCoE switch mode and supported hardware, see *Cisco Nexus 9000 Series NX-OS SAN Switching Configuration Guide, Release 10.1(x)*.
- Beginning with Cisco NX-OS Release 10.1(1), Enhanced ISSU is supported on FC/FCoE switch mode for Cisco Nexus 93180YC-FX and 93360YC-FX2 switches. For more information about the FC/FCoE switch mode and supported hardware, see *Cisco Nexus 9000 Series NX-OS SAN Switching Configuration Guide, Release 10.1(x)*.
- Beginning with Cisco NX-OS Release 10.1(1), Enhanced ISSU is supported on FC/FCoE NPV mode for Cisco Nexus 93180YC-FX and 93360YC-FX2 switches. 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, Release 10.1(x)*.
- 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 10.1(x) or 9.2(3) to 10.1(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 10.1(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 10.1(x), you should manually copy the image to the standby SUP and perform the disruptive upgrade.
- When upgrading directly to Cisco NX-OS Release 10.1(x) from any release prior to 7.0(x), the upgrade will be disruptive. For a non-disruptive upgrade, an intermediate upgrade to Cisco NX-OS Release 9.x is required. We recommend upgrading to the latest release of Cisco NX-OS Release 9.3(x) as an intermediate hop for the upgrade. For information about the supported upgrade paths, see the [Cisco Nexus 9000 and 3000 Upgrade and ISSU Matrix](#).
- When upgrading from Cisco NX-OS Release 7.0(3)I6(1) or 7.0(3)I7(1) to Cisco NX-OS Release 10.1(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 10.1(x) by using the **install all** command, BIOS will not be upgraded due to CSCve24965. When the upgrade to Cisco NX-OS Release 10.1(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 10.1(x) might result in the VLANs being unable to be added to the existing FEX HIF trunk ports. To recover from this, the following steps should be performed after all FEXs have come online and the HIFs are operationally up:
  1. Enter the copy run bootflash:fex\_config\_restore.cfg command at the prompt.
  2. Enter the copy bootflash:fex\_config\_restore.cfg running-config echo-commands command at the prompt.
- In 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 10.1(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 10.1(x). Uploading the system creates the required match qualifiers for the FHS TCAM region, ifacl.
- Before enabling the FHS, we recommend that you carve the ing-redirect TCAM region on 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 10.1(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.
- The install all command is the recommended method for software upgrades because it performs configuration compatibility checks and BIOS upgrades automatically. In contrast, changing the boot variables and reloading the device bypasses these checks and the BIOS upgrade and therefore it is not recommended.
- 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.
- 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 10.1(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, SHA256 or SHA512 checksum of the software image. To verify the MD5 checksum of the software image, run the **show file bootflash:<IMAGE-NAME>md5sum** command and compare the resulting value to the published MD5 checksum for the software image on [Cisco's Software Download](#) website. To verify the SHA512 checksum of the software image, run the **show file bootflash:<IMAGE-NAME>sha512sum** command and compare the resulting value to the published SHA512 checksum for the software image on [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 10.1(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 10.1(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 10.1(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.
- When performing ND ISSU using BGP non-default hold timers, ensure that the BGP graceful-restart timer is reasonably long enough, for example, 180 seconds.
- During an ISSU on a Cisco Nexus 9300 Series switch, all First-Hop Redundancy Protocols (FHRPs) will cause the other peer to become active if the node undergoing the ISSU is active.
- 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, 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-aggr1 %$ 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-aggr1 %$ 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.
- ISSU supports EPLD image upgrades using **install all nxos <nxos-image> epld <epld-image>** command, during disruptive system (NX-OS) upgrade.
- A simplified NX-OS numbering format is used for platforms that are supported in Cisco NX-OS 10.1(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 10.1(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

```

- 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.
- Beginning with Cisco NX-OS Release 10.1(1), **Fs\_daemon** does not support **snmpwalk** on devices with more than 5000 files. When performing snmpwalk on a device with more than 5000 files, the error **resourceUnavailable (This is likely a out-of-memory failure within the agent)** is an expected behaviour.
- Beginning with Cisco NX-OS Release 10.1(2), CoPP is supported on N9K-X9624D-R2 and N9K-C9508-FM-R2 platform switches.
- Beginning with Cisco NX-OS Release 10.1(2), RACL is supported on N9K-X9624D-R2 and N9K-C9508-FM-R2 platform switches.
- ISSU is blocked when the delay config is present in track list Boolean/weight.
- If there is a VRF scale, for a non-disruptive ISSU under each VRF, you must configure graceful restart timer to 300 seconds.
- 

## ISSU Platform Support

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





**Note** Enhanced ISSU: Enhanced ISSU to Cisco NX-OS Release 10.1(x) is not supported as there are kernel updates that cannot take effect without reloading the underlying kernel. The system will prompt the following message:

```
Host kernel is not compatible with target image. Full ISSU will be performed and control plane will be impacted.
```

In effect, system will perform nondisruptive ISSU instead of enhanced ISSU.

### 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):<br>Cisco Nexus 92300YC<br><br>Beginning with Cisco NX-OS Release 9.3(3):<br>Cisco Nexus 92348GC-X | Both ISSU types are disruptive for Cisco Nexus 9200 platform switches configured with the following features: <ul style="list-style-type: none"> <li>• Segment routing</li> <li>• Tetration</li> </ul> |
| Enhanced  | Cisco Nexus 92300YC   |  |

### 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 9.3(3):<br>Cisco Nexus 9332C<br>Cisco Nexus 9364C<br><br><b>Note</b> ISSU on Cisco Nexus 9300 platform switches is supported when the switch is the spanning tree root. You can use the <b>show spanning-tree issu-impact</b> 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: <ul style="list-style-type: none"> <li>• Dual-homed FEX</li> <li>• Segment routing</li> <li>• VXLAN</li> </ul> |
| Enhanced  | Beginning with Cisco NX-OS Release 9.3(5):<br>Cisco Nexus 9332C<br>Cisco Nexus 9364C<br><br><b>Note</b> ISSU on Cisco Nexus 9300 platform switches is supported when the switch is the spanning tree root. You can use the <b>show spanning-tree issu-impact</b> command to verify if the switch meets this criteria. |  |

**ISSU for Cisco Nexus 9300-EX Platform Switches**

| <b>ISSU Type</b> | <b>Release/Supported Platforms</b>  | <b>Features Not Supported with Non-disruptive ISSU</b>  |
|------------------|---|---|
| Standard         | Beginning with Cisco NX-OS Release 7.0(3)I6(1):<br>Cisco Nexus 93108TC-EX<br>Cisco Nexus 93180YC-EX | Both ISSU types are disruptive for Cisco Nexus 9300-EX platform switches configured with the following features: <ul style="list-style-type: none"> <li>• Segment routing</li> <li>• Tetration</li> </ul> |
| Enhanced         | Beginning with Cisco NX-OS Release 7.0(3)I7(3):<br>Cisco Nexus 93108TC-EX<br>Cisco Nexus 93180YC-EX |   |

**ISSU for Cisco Nexus 9300-FX Platform Switches**

| <b>ISSU Type</b> | <b>Release/Supported Platforms</b>  | <b>Features Not Supported with Non-disruptive ISSU</b>   |
|------------------|---|--|
| Standard         | Cisco NX-OS Release 9.3(1) and 9.3(2): None<br>Beginning with Cisco NX-OS Release 9.3(3):<br>Cisco Nexus 9336C-FX2<br>Cisco Nexus 93240YC-FX2<br>Cisco Nexus 93240YC-FX2Z<br>Cisco Nexus 9348GC-FXP<br>Cisco Nexus 93108TC-FX<br>Cisco Nexus 93180YC-FX | Standard ISSU is disruptive for Cisco Nexus 9300-FX platform switches configured with the following features: <ul style="list-style-type: none"> <li>• Segment Routing</li> <li>• TRM Feature</li> </ul> |

| ISSU Type | Release/Supported Platforms  | Features Not Supported with Non-disruptive ISSU  |
|-----------|--|--|
| Enhanced  | <p>Cisco NX-OS Release 9.3(1), 9.3(2), and 9.3(3): None</p> <p>Beginning with Cisco NX-OS Release 9.3(5):</p> <p>Cisco Nexus 9336C-FX2</p> <p>Cisco Nexus 93240YC-FX2</p> <p>Cisco Nexus 93216TC-FX2</p> <p>Cisco Nexus 93360YC-FX2</p> <p>Cisco Nexus 93240YC-FX2Z</p> <p>Cisco Nexus 9348GC-FXP</p> <p>Cisco Nexus 93108TC-FX</p> <p>Cisco Nexus 93180YC-FX</p> <p>Beginning with Cisco NX-OS Release 10.1(1), Enhanced ISSU is supported on the following platforms with FC/FCoE features:</p> <p>Cisco Nexus 93360YC-FX2</p> <p>Cisco Nexus 93180YC-FX</p> | <p>Enhanced ISSU is disruptive for Cisco Nexus 9300-FX platform switches configured with the following features:</p> <ul style="list-style-type: none"> <li>• Segment Routing</li> <li>• TRM Feature</li> </ul> <p><b>Note</b> In Cisco NX-OS Releases 9.3(x), Enhanced ISSU on Cisco Nexus 93360YC-FX2 and Cisco Nexus 93180YC-FX with FC/FCoE features will be disruptive.</p> |

#### ISSU for Cisco Nexus 9300-GX Platform Switches

| ISSU Type | Release/Supported Platforms  | Features Not Supported with Non-disruptive ISSU  |
|-----------|--|--|
| Standard  | <p>Beginning with Cisco NX-OS Release 10.1(1):</p> <p>Cisco Nexus 9364C-GX</p> <p>Cisco Nexus 9316D-GX</p> <p>Cisco Nexus 93600CD-GX</p> | <ul style="list-style-type: none"> <li>• TRM Feature</li> <li>• Segment Routing</li> </ul> |
| Enhanced  | <p>Beginning with Cisco NX-OS Release 10.1(1):</p> <p>Cisco Nexus 9364C-GX</p> <p>Cisco Nexus 9316D-GX</p> <p>Cisco Nexus 93600CD-GX</p> | <ul style="list-style-type: none"> <li>• TRM Feature</li> <li>• Segment Routing</li> </ul> |

## 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
- 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 10.1(x) to earlier releases (9.2(x) or 7.x), the TCAM region that applies to NetFlow (ing-netflow) should be carved to zero (0) using the following command:

#### **hardware access-list tcam region ing-netflow 0**

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

- When downgrading from the Cisco NX-OS Release 10.1(x) to a release prior to 9.3(x), make sure that the ACL TCAM usage for ingress features does not exceed the allocated TCAM space in the absence of the label sharing feature. Label sharing is a new feature in 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, Release 10.1(x)*.
- Software downgrades are disruptive. In-service software downgrades (ISSDs), also known as nondisruptive downgrades, are not supported.
- When downgrading from Cisco NX-OS Release 9.3(3) or later to 7.0(3)I7(7), disable BFD for the BGP neighbor prefix peer using the **no bfd** command.

## Upgrade Paths

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

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

```

```

..done Thu Mar 3 04:42:11 UTC 2016
Creating /dev/mcelog
Starting mcelog daemon
Overwriting dme stub lib
Replaced dme stub lib
INIT: Entering runlevel: 3
Running S93thirdparty-script...

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

User Access Verification
switchlogin:
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
  of Module 1
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
  of Module 6
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
  of Module 8
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
  of Module 9
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
  of Module 10
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
  of Module 11
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
  of Module 14
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
  of Module 15
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
  of Module 16
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
  of Module 21
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
  of Module 22
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
  of Module 23
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
  of Module 24
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
  of Module 25

```

```

2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
of Module 26
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
of Module 28
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
of Module 29
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
of Module 30
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-PS_OK: Power supply 1 ok (Serial
number XYZ284014RR)
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-PS_FANOK: Fan in Power supply 1 ok
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-PS_OK: Power supply 2 ok (Serial
number XYZ285111TC)
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-PS_FANOK: Fan in Power supply 2 ok
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-PS_OK: Power supply 3 ok (Serial
number XYZ285111QQ)
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-PS_FANOK: Fan in Power supply 3 ok
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-PS_OK: Power supply 4 ok (Serial
number XYZ284014TI)
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-PS_FANOK: Fan in Power supply 4 ok
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-PS_OK: Power supply 5 ok (Serial
number XYZ284014TS)
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-PS_FANOK: Fan in Power supply 5 ok
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-FANMOD_FAN_OK: Fan module 1
(Fan1(sys_fan1) fan) ok
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-FANMOD_FAN_OK: Fan module 2
(Fan2(sys_fan2) fan) ok
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-FANMOD_FAN_OK: Fan module 3
(Fan3(sys_fan3) fan) ok
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 30 detected (Serial
number ABC1234DE56) Module-Type System Controller Model N9K-SC-A
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 30 powered up (Serial
number ABC1234DE56)
2016 Mar 3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 28 detected (Serial
number :unavailable) Module-Type Supervisor Module Model :unavailable
2016 Mar 3 04:43:58 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 29 detected (Serial
number ABC1234DEFG) Module-Type System Controller Model N9K-SC-A
2016 Mar 3 04:43:58 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 29 powered up (Serial
number ABC1234DEFG)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 21 detected (Serial
number ABC1213DEFG) Module-Type Fabric Module Model N9K-C9516-FM
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 22 detected (Serial
number ABC1211DEFG) Module-Type Fabric Module Model N9K-C9516-FM
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 21 powered up (Serial
number ABC1213DEFG)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 22 powered up (Serial
number ABC1211DEFG)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 23 detected (Serial
number ABC1234D5EF) Module-Type Fabric Module Model N9K-C9516-FM
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 23 powered up (Serial
number ABC1234D5EF)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 24 detected (Serial
number ABC1211DE3F) Module-Type Fabric Module Model N9K-C9516-FM
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 24 powered up (Serial
number ABC1211DE3F)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 25 detected (Serial
number ABC1213DEFG) Module-Type Fabric Module Model N9K-C9516-FM
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 25 powered up (Serial
number ABC1213DEFG)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 26 detected (Serial
number ABC1211DE34) Module-Type Fabric Module Model N9K-C9516-FM
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 26 powered up (Serial
number ABC1211DE34)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED: All

```

```
Ejectors closed for module 1. Ejector based shutdown enabled
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 1 detected (Serial
number ABC1217DEFG) Module-Type 32p 40G Ethernet Module Model N9K-X9432PQ
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 1 powered up (Serial
number ABC1217DEFG)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED: All
Ejectors closed for module 9. Ejector based shutdown enabled
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 9 detected (Serial
number ABC1236D4E5) Module-Type 48x1/10G-T 4x40G Ethernet Module Model N9K-X9564TX
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 9 powered up (Serial
number ABC1236D4E5)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED: All
Ejectors closed for module 10. Ejector based shutdown enabled
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 10 detected (Serial
number ABC1217EFGH) Module-Type 32p 40G Ethernet Module Model N9K-X9432PQ
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 10 powered up (Serial
number ABC1217EFGH)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED: All
Ejectors closed for module 11. Ejector based shutdown enabled
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 11 detected (Serial
number ABC123DEF4) Module-Type 36p 40G Ethernet Module Model N9K-X9536PQ
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 11 powered up (Serial
number ABC123DEF4)
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED: All
Ejectors closed for module 15. Ejector based shutdown enabled
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 15 detected (Serial
number ABC1212DEFG) Module-Type 36p 40G Ethernet Module Model N9K-X9536PQ
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 15 powered up (Serial
number ABC1212DEFG)
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED: All
Ejectors closed for module 16. Ejector based shutdown enabled
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 16 detected (Serial
number ABCD1235DEFG) Module-Type 48x1/10G SFP+ 4x40G Ethernet Module Model N9K-X9464PX
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 16 powered up (Serial
number ABCD1235DEFG)
2016 Mar 3 04:44:08 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED: All
Ejectors closed for module 14. Ejector based shutdown enabled
2016 Mar 3 04:44:08 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 14 detected (Serial
number ABC9876DE5F) Module-Type 8p 100G Ethernet Module Model N9K-X9408PC-CFP2
2016 Mar 3 04:44:08 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 14 powered up (Serial
number ABC9876DE5F)
2016 Mar 3 04:44:09 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED: All
Ejectors closed for module 6. Ejector based shutdown enabled
2016 Mar 3 04:44:09 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 6 detected (Serial
number ABC9876DE3F) Module-Type 8p 100G Ethernet Module Model N9K-X9408PC-CFP2
2016 Mar 3 04:44:09 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 6 powered up (Serial
number ABC9876DE3F)
2016 Mar 3 04:44:10 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED: All
Ejectors closed for module 8. Ejector based shutdown enabled
2016 Mar 3 04:44:10 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 8 detected (Serial
number ABC3456D7E8) Module-Type 48x1/10G-T 4x40G Ethernet Module Model N9K-X9564TX
2016 Mar 3 04:44:10 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 8 powered up (Serial
number ABC3456D7E8)
2016 Mar 3 04:44:56 switch%$ VDC-1 %$ %USBHSD-STANDBY-2-MOUNT: logflash: online
2016 Mar 3 04:47:31 switch%$ VDC-1 %$ %ASCII-CFG-2-CONF_CONTROL: System ready
2016 Mar 3 04:47:51 switch%$ VDC-1 %$ %VMAN-2-ACTIVATION_STATE: Successfully activated
virtual service 'guestshell+'
2016 Mar 3 04:47:51 switch%$ VDC-1 %$ %VMAN-2-GUESTSHELL_ENABLED: The guest shell has
been enabled. The command 'guestshell' may be used to access it, 'guestshell destroy'
to remove it.
```

```
User Access Verification
```

```
switch# show version
```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

plugin
  Core Plugin, Ethernet Plugin

```

```

Active Package(s):
switch#

```

## Configuring Enhanced ISSU

You can enable or disable enhanced (LXC) ISSU.

**Note**

- Enhanced ISSU to Cisco NX-OS Release 10.1(x) is not supported as there are kernel updates that cannot take effect without reloading the underlying kernel. The system will prompt the following message:

Host kernel is not compatible with target image. Full ISSU will be performed and control plane will be impacted.

In effect, system will perform nondisruptive ISSU instead of enhanced ISSU.

**Before you begin**

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

**SUMMARY STEPS**

1. **configure terminal**
2. **[no] boot mode lxc**
3. (Optional) **show boot mode**
4. **copy running-config startup-config**
5. **reload**

**DETAILED STEPS**

|        | Command or Action  | Purpose  |
|--------|--|--|
| Step 1 | <b>configure terminal</b><br><b>Example:</b><br><pre>switch# configure terminal switch(config#)</pre>  | Enters global configuration mode.  |
| Step 2 | <b>[no] boot mode lxc</b><br><b>Example:</b><br><pre>switch(config)# boot mode lxc Using LXC boot mode</pre> <b>Example:</b><br><pre>switch(config)# no boot mode lxc Using normal native boot mode</pre>        | Enables or disables enhanced (LXC) ISSU.<br><b>Note</b> In order to perform a nondisruptive enhanced ISSU, you must first boot the switch in LXC mode. |
| Step 3 | (Optional) <b>show boot mode</b><br><b>Example:</b><br><pre>switch(config)# show boot mode LXC boot mode is enabled</pre> <b>Example:</b><br><pre>switch(config)# show boot mode LXC boot mode is disabled</pre> | Shows whether enhanced (LXC) ISSU is enabled or disabled.  |
| Step 4 | <b>copy running-config startup-config</b><br><b>Example:</b>   | Saves the running configuration to the startup configuration.  |

|               | Command or Action  | Purpose  |
|---------------|--|--|
|               | <code>switch(config)# copy running-config startup-config</code>  |  |
| <b>Step 5</b> | <b>reload</b><br><br><b>Example:</b><br><pre>switch(config)# reload This command will reboot the system. (y/n)? [n] Y loader&gt;</pre> | Reloads the device. When prompted, press <b>Y</b> 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 10.1(x) release.



**Note** Beginning with Cisco NX-OS Release 10.1(1), the Cisco Nexus -GX series platforms use the 64-bit Cisco NX-OS image file, which has the image filename that begins with "nxos64" (for example, nxos64.10.1.1.bin). The 64-bit software image, which supports software scalability, is available for the Cisco Nexus C9316D-GX, C93600CD-GX, C9364C-GX switches. The non-GX series platforms use the 32-bit Cisco NX-OS image file, which has the image filename that begins with "nxos" (for example, nxos.10.1.1.bin).



**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 10.1(1). To upgrade from Cisco NX-OS Release 9.2(2) or later, 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, Release 10.1(x)* for a list of possible causes and solutions.

### Before you begin

Before performing a nondisruptive ISSU to Cisco NX-OS Release 10.1(1), you must configure the BGP graceful restart timer to 180 seconds for Cisco Nexus 3132Q-V platform switches.

### SUMMARY STEPS

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](#).
2. Log in to the device on the console port connection.



3. Ensure that the required space is available for the image file to be copied.
4. If you need more space on the active supervisor module, delete unnecessary files to make space available.
5. Verify that there is space available on the standby supervisor module.
6. If you need more space on the standby supervisor module, delete any unnecessary files to make space available.
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>.
8. Copy the software image to the active supervisor module using a transfer protocol. You can use FTP, TFTP, SCP, or SFTP.
9. You can detect an incomplete or corrupt NX-OS software image prior to performing an upgrade by verifying the MD5, SHA256 or SHA512 checksum of the software image. To verify the MD5 checksum of the software image, run the **show file bootflash:<IMAGE-NAME>md5sum** command and compare the resulting value to the published MD5 checksum for the software image on [Cisco's Software Download](#) website. To verify the SHA512 checksum of the software image, run the **show file bootflash:<IMAGE-NAME>sha512sum** command and compare the resulting value to the published SHA512 checksum for the software image on [Cisco's Software Download](#) website.
10. Check the impact of upgrading the software before actually performing the upgrade.
11. Save the running configuration to the startup configuration.
12. If required, upgrade the EPLD image using the **install all nxos <nxos-image> epld <epld-image>** command.
13. Upgrade the Cisco NX-OS software using the **install all nxos bootflash:filename [no-reload | non-disruptive | non-interruptive | serial]** command.
14. (Optional) Display the entire upgrade process.
15. (Optional) Log in and verify that the device is running the required software version.
16. (Optional) If necessary, install any licenses to ensure that the required features are available on the device. See the [Cisco NX-OS Licensing Guide](#).

## DETAILED STEPS

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

**Note** We recommend that you have the image file for at least one previous release of the 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:nxos.9.2.1.bin
```

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

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

**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/nxos.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.10.1.1.bin bootflash:nxos.10.1.1.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.10.1.1.bin
bootflash:nxos.10.1.1.bin compact vrf management use-kstack
```

```
user1@10.65.42.196's password:
nxos.10.1.1.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, SHA256 or SHA512 checksum of the software image. To verify the MD5 checksum of the software image, run the **show file bootflash:<IMAGE-NAME>md5sum** command and compare the resulting value to the published MD5 checksum for the software image on [Cisco's Software Download](#) website. To verify the SHA512 checksum of the software image, run the **show file bootflash:<IMAGE-NAME>sha512sum** command and compare the resulting value to the published SHA512 checksum for the software image on [Cisco's Software Download](#) website.

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

switch# show file bootflash:nxos.10.1.1.bin sha512sum
7f25cce57ca137a79211fb3835338aae64acf9b021b75c0c5d4156e873b4274ca4f98e9a74fe4c8961f5ace99ed65f3826650599369f84ab07265d7c5d61b57f
```

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

```
switch# # show install all impact nxos bootflash:nxos.10.1.1.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.   |
| Default upgrade is not hitless | By default, the software upgrade process is disruptive. You must configure the <b>non-disruptive</b> option to perform an ISSU. |

**Step 11** Save the running configuration to the startup configuration.

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

**Step 12** If required, upgrade the EPLD image using the **install all nxos <nxos-image> epld <epld-image>** command.

The following is an example output of the **install all nxos <nxos-image> epld <epld-image>** command:

```
switch# install all nxos nxos.10.1.1.bin epld n9000-epld.10.1.1.img

Installer will perform compatibility check first. Please wait.
Installer is forced disruptive

Verifying image bootflash:/nxos.10.1.1.bin for boot variable "nxos".
[#####] 100% -- SUCCESS

Verifying EPLD image bootflash:/ n9000-epld.10.1.1.img.
[#####] 100% -- SUCCESS

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

Preparing "nxos" version info using image bootflash:/nxos.10.1.1.bin.
[#####] 100% -- SUCCESS

Preparing "bios" version info using image bootflash:/nxos.10.1.1.bin.
[#####] 100% -- SUCCESS

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

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

**Step 13** 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.10.1.1.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 14** (Optional) Display the entire upgrade process.

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

**Step 15** (Optional) Log in and verify that the device is running the required software version.

```
switch# show version
```

**Step 16** (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.

In a dual-homed non-vPC access network (either a triangle or Y shaped access network), with or without STP configured, with BFD enabled and with HSRP configured on the SVIs and with HSRP configured as BFD client, transient traffic may drop for both IPv4 native unicast and/or labeled traffic after performing an ND-ISSU in fallback mode.

To counter this, configure in all the HSRP IPv4 groups for all the HSRP enabled SVIs the **timer 2 120** on both HSRP peers prior to performing the ND-ISSU. The configuration of the **timer 3 120** may lead to traffic loss.

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, Release 10.1(x)* for a list of possible causes and solutions.

---

## SUMMARY STEPS

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](#).
2. Log in to the device on the console port connection.
3. Verify that the image file for the downgrade is present on the active supervisor module bootflash:.
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>.
5. Copy the software image to the active supervisor module using a transfer protocol. You can use FTP, TFTP, SCP, or SFTP.
6. Check for any software incompatibilities.
7. Disable any features that are incompatible with the downgrade image.
8. Check for any hardware incompatibilities.
9. Power off any unsupported modules.
10. Save the running configuration to the startup configuration.
11. Downgrade the Cisco NX-OS software.
12. (Optional) Display the entire downgrade process.
13. (Optional) Log in and verify that the device is running the required software version.

## DETAILED STEPS

---

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

**Step 6** Check for any software incompatibilities.

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

The resulting output displays any incompatibilities and remedies.

**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:nxos.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.libnbproxyccli_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.libnbproxyccli_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, on page 39](#)
- [Using Modular Packages, on page 40](#)
- [Booting the NX-OS Image in Base or Full Mode, on page 41](#)
- [Information About RPMs, on page 42](#)
- [Information About DNF Commands, on page 56](#)
- [Configuring an FTP server and Setting up a Local FTP YUM Repository, on page 70](#)
- [Creating User Roles for Install Operation, on page 74](#)
- [Compacting Cisco NX-OS Software Images, on page 74](#)

## 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.

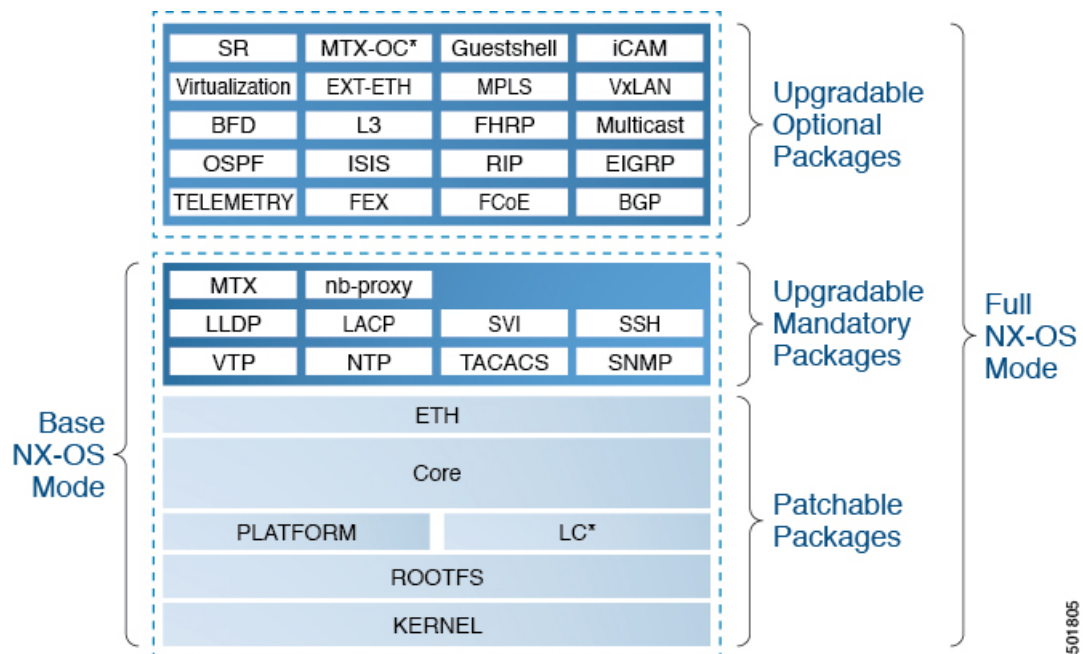
The advantages for using modular Cisco NX-OS software are:

- 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 1: Optionality in Cisco NX-OS Software



501805

- 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 DNF

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

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

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

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

## 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 DNF 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<br>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 **dnf** 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           | <ul style="list-style-type: none"> <li>• feature openflow</li> <li>• feature evb</li> <li>• feature imp</li> <li>• feature netflow</li> <li>• feature sla_sender</li> <li>• feature sla_responder</li> <li>• feature sla twamp-server</li> <li>• feature sflow</li> </ul> |
| EXT_ETH_LOWMEM    | <ul style="list-style-type: none"> <li>• feature evb</li> <li>• feature netflow</li> </ul>  |
| FCoE              | <ul style="list-style-type: none"> <li>• feature-set fcoe</li> <li>• feature-set fcoe-npv</li> </ul>  |
| FEX               | feature-set fex   |
| FHRP              | <ul style="list-style-type: none"> <li>• feature hsrp</li> <li>• feature vrrpv3</li> </ul>  |
| HW TELEMETRY      | feature hw telemetry  |
| iCAM              | feature icam  |
| ISIS              | feature isis  |

| Package Name   | Associated Features   |
|----------------|---|
| MPLS           | <ul style="list-style-type: none"> <li>• feature mpls segment-routing</li> <li>• feature mpls evpn</li> </ul>                             |
| Multicast      | <ul style="list-style-type: none"> <li>• feature pim</li> <li>• feature pim6</li> <li>• feature msdp</li> <li>• feature ngmvpn</li> </ul> |
| NIA            | NA  |
| NXSDK          | NA  |
| OSPF           | <ul style="list-style-type: none"> <li>• feature ospf</li> <li>• feature ospfv3</li> </ul>  |
| RIP            | feature rip   |
| SDAA           | NA  |
| Services       | feature catena  |
| SR             | feature mpls segment-routing traffic-engineering  |
| TELEMETRY      | feature telemetry   |
| Virtualization | NA  |
| VM Tracker     | feature vmtracker   |
| VXLAN          | <ul style="list-style-type: none"> <li>• feature nv overlay</li> <li>• feature fabric forwarding</li> </ul>                               |

## 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 DNF commands.

---

Table 4: RPM Repositories That Are Present in the Switches

| Repository Name | Repository Path                         | Description   |
|-----------------|---|---|
| groups-repo     | /rpms                                   | Part of the bundled NX-OS image. It is used to keep all the RPMs that are bundled as part of the NX-OS image. All RPMs based in this repository are known as base RPMs.   |
| localdb         | /bootflash/.rpmstore/patching/localrepo | Used for RPM persistency. When a user adds a NX-OS feature RPM as part of <b>install add</b> command, the RPM is copied to this location and it is persisted during the reloads. User has the responsibility to clean the repository.<br><br>To add a RPM to this repository, use <b>install add</b> command.<br><br>To remove a RPM from this repository, use <b>install remove</b> command.<br><br>DNF commands can be used to populate the repository too.<br><br>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. DNF 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.

## Guidelines for Third-Party RPM Installation

Currently, any third-party package that is not provided by Cisco is allowed to be installed on the device, even when it is not signed by Cisco.

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

## Using Install CLIs for Feature or Third-Party 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  |
|--------------------------------|--|
| <b>install reset</b>           | <p>This operation removes all the patches, persisted configurations, upgraded packages, third-party installed packages, unsaved configurations, and reloads the switch's previous mode (Full/Base) with the default packages.</p> <p>The <b>install reset</b> command also performs write erase operation. The following message is displayed at the prompt:</p> <pre>switch(config)# install reset</pre> <hr/> <p>WARNING!!This operation will remove all patches, upgraded packages, persisted etc configs, third party packages installed, startup configuration(write erase) and reload the switch with default packages.</p> <hr/> <p>Do you want to proceed with reset operation? (y/n)? [n]</p> |
| <b>install reset nxos base</b> | <p>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.</p>  |
| <b>install reset nxos full</b> | <p>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).</p>  |



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

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



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

## Using Install CLIs for Digital Signature Support

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

**SUMMARY STEPS**

1. switch#install add bootflash:<keyfile> gpg-key
2. switch#install verify package <package-name>
3. OR switch#install verify bootflash:<RPM file>

**DETAILED STEPS**

|               | Command or Action  | Purpose  |
|---------------|--|--|
| <b>Step 1</b> | switch#install add bootflash:<keyfile> gpg-key<br><br><b>Example:</b><br><br><pre>install add bootflash:RPM-GPG-KEY-puppetlabs gpg-key [#####] 100% Install operation 304 completed successfully at Thu Jun 19 16:40:28 2018</pre> | Cisco release RPMs are signed with Cisco GPG (GNU Privacy Guard) key. The public GPG key is present at <b>/etc/pki/rpm-gpg/arm-Nexus9k-rel.gpg</b> . To add other public keys from different sources, use the steps in this section. |
| <b>Step 2</b> | switch#install verify package <package-name>   | Verifies the package.  |
| <b>Step 3</b> | OR switch#install verify bootflash:<RPM file><br><br><b>Example:</b><br><br><pre>switch# install verify bootflash:vxlan-2.0.0.0-9.2.1.lib32_n9000.rpm  RSA signed switch#</pre>  | 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:

**SUMMARY STEPS**

1. show install packages

**DETAILED STEPS**

|               | Command or Action  | Purpose                         |
|---------------|--|---------------------------------|
| <b>Step 1</b> | show install packages<br><br><b>Example:</b><br><br><pre>switch# show install packages  Boot Image: NXOS Image: bootflash:/nxos.9.2.1.bin  ----- Installed Packages attr.x86_64 2.4.47-r0.0 installed Unsigned</pre> | Queries all the installed RPMs. |

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

## 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                         |
|--|---------------------------------|
| <p><b>Step 1</b></p> <pre>install add &lt;rpm&gt; activate</pre> <p><b>Example:</b></p> <pre>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)</pre> | Installs and activates the RPM. |

|  | Command or Action   | Purpose |
|--|---|---------|
|  | <pre>[#####] 100% Install operation 869 completed successfully at Tue May 8 11:20:20 2018</pre> |         |

### 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:

### SUMMARY STEPS

1. `install add <rpm>`
2. `install activate <rpm>`

### DETAILED STEPS

|        | Command or Action   | Purpose           |
|--------|---|-------------------|
| Step 1 | <pre>install add &lt;rpm&gt;  Example:  switch# install add bootflash:vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm</pre> | Installs the RPM. |

|               | Command or Action   | Purpose            |
|---------------|---|--------------------|
|               | <pre>[#####] 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</pre> |                    |
| <b>Step 2</b> | <b>install activate &lt;rpm&gt;</b><br><b>Example:</b>  | 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           |
|---------------|--|-------------------|
| <b>Step 1</b> | <b>install add &lt;rpm&gt;activate upgrade</b><br><b>Example:</b><br><pre>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</pre> | Installs the RPM. |

|  | Command or Action   | Purpose |
|--|---|---------|
|  | <pre> 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                     </pre> |         |

**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                    |
|----------------------|--|----------------------------|
| <p><b>Step 1</b></p> | <pre> install add &lt;rpm&gt;activate downgrade  Example:  switch(config)# install add bootflash:bgp-2.0.1.0-9.2.1.lib32_n9000.rpm activate downgrade  Adding the patch                     </pre> | <p>Downgrades the RPM.</p> |

|  | Command or Action  | Purpose |
|--|--|---------|
|  | <pre>(/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</pre> |         |

### Example

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

Active Packages:
  bgp-2.0.1.0-9.2.1.lib32_n9000
  chef-12.0.0alpha.2+20150319234423.git.1608.b6eb10f-1.e15.x86_64

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

## Uninstalling the RPMs

See the following steps to uninstall the RPMs:

### SUMMARY STEPS

1. **install deactivate** *<rpm>*



## DETAILED STEPS

|        | Command or Action  | Purpose   |
|--------|--|---|
| Step 1 | <p><b>install deactivate</b> &lt;rpm&gt;</p> <p><b>Example:</b></p> <pre>switch(config)# install deactivate bgp Base RPM found. Do you want to downgrade to base version(y/n) [n] y Downgrading to the base version [#####] 100% Install operation 190 completed successfully at Tue Nov 17 04:10:40 2020</pre> <p><b>Example:</b></p> <pre>switch(config)# install deactivate bgp Base RPM found. Do you want to downgrade to base version(y/n) [n] n</pre> <hr/> <pre>WARNING!! This operation will remove 'bgp-3.0.0.0-9.4.1.lib32_n9000' related configuration from running-configuration on successful completion. Update startup-configuration accordingly.</pre> <hr/> <pre>[#####] 100% Install operation 9 completed successfully at Tue Nov 17 05:05:59 2020</pre> | <p>Downgrades to the base version of RPM, if one exists in the groups-repo (/rpms), or uninstalls the RPM completely from the switch.</p> <p>To downgrade to the base version, enter <b>y</b>. To completely uninstall the RPM, enter <b>n</b> in the command prompt.</p> |

## Removing the RPMs

See the following steps to remove the RPMs:

## SUMMARY STEPS

1. **install remove** <rpm>

## DETAILED STEPS

|        | Command or Action   | Purpose                              |
|--------|---|--------------------------------------|
| Step 1 | <p><b>install remove</b> &lt;rpm&gt;</p> <p><b>Example:</b></p> <pre>switch(config)# show install inactive   grep vxlan  vxlan-2.0.0.0-9.2.1.lib32_n9000 switch(config)# install remove vxlan</pre> <pre>Proceed with removing vxlan? (y/n)? [n] y [#####] 100%</pre> | Removes the RPM from the repository. |

| Command or Action   | Purpose |
|---|---------|
| Install operation 890 Removal of base rpm package is not permitted at Thu Jun 7 13:52:15 2018 |         |

## Information About DNF Commands

See the following sections for more information about DNF commands.



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

## Performing Package Operations Using the DNF Commands

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



**Note** DNF commands are accessed only from the BASH shell on the box and they are not allowed from the 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
ospf-2.0.0.0-9.2.1.lib32_n9000.rpm
fhrp-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
  rip-2.0.0.0-9.2.1.lib32_n9000.rpm
icam-2.0.0.0-9.2.1.lib32_n9000.rpm
  services-2.0.0.0-9.2.1.lib32_n9000.rpm
mpls-2.0.0.0-9.2.1.lib32_n9000.rpm
vtp-2.0.0.0-9.2.1.lib32_n9000.rpm
mtx-device-2.0.0.0-9.2.1.lib32_n9000.rpm
mtx-grpc-agent-2.0.0.0-9.2.1.lib32_n9000.rpm
mtx-infra-2.0.0.0-9.2.1.lib32_n9000.rpm

```

## Checking the List of the Installed RPMs

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

```

bash-4.2# dnf list installed | grep lib32_n9000

bfd.lib32_n9000                2.0.0.0-9.2.1          @groups-repo
core.lib32_n9000              2.0.0.0-9.2.1          installed
eth.lib32_n9000               2.0.0.0-9.2.1          installed
guestshell.lib32_n9000        2.0.0.0-9.2.1          @groups-repo
lACP.lib32_n9000              2.0.0.0-9.2.1          installed
linecard2.lib32_n9000         2.0.0.0-9.2.1          installed
lldp.lib32_n9000              2.0.0.0-9.2.1          installed
mcast.lib32_n9000             2.0.0.0-9.2.1          @groups-repo
mtx-device.lib32_n9000        2.0.0.0-9.2.1          installed
mtx-grpc-agent.lib32_n9000    2.0.0.0-9.2.1          installed
mtx-infra.lib32_n9000         2.0.0.0-9.2.1          installed
mtx-netconf-agent.lib32_n9000 2.0.0.0-9.2.1          installed
mtx-restconf-agent.lib32_n9000 2.0.0.0-9.2.1          installed
mtx-telemetry.lib32_n9000     2.0.0.0-9.2.1          installed
nbproxy.lib32_n9000           2.0.0.0-9.2.1          installed
ntp.lib32_n9000               2.0.0.0-9.2.1          installed
nxos-ssh.lib32_n9000          2.0.0.0-9.2.1          installed
ospf.lib32_n9000              2.0.0.0-9.2.1          @groups-repo
platform.lib32_n9000          2.0.0.0-9.2.1          installed
snmp.lib32_n9000              2.0.0.0-9.2.1          installed
svi.lib32_n9000               2.0.0.0-9.2.1          installed
tacacs.lib32_n9000            2.0.0.0-9.2.1          installed
tor.lib32_n9000               2.0.0.0-9.2.0.77       installed
virtualization.lib32_n9000    2.0.1.0-9.2.1          @localdb
vtp.lib32_n9000               2.0.0.0-9.2.1          installed
vxlan.lib32_n9000             2.0.0.0-9.2.1          @groups-repo
...

```

## Getting Details of the Installed RPMs

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

```
dnf info vxlan
```

```

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

```

```

| 1.1 kB    00:00 ...
localdb
| 951 B     00:00 ...

```

```

patching
| 951 B    00:00 ...
thirdparty
| 951 B    00:00 ...

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

```

## 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# dnf install
http://10.0.0.2/modularity/rpms/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm

Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
groups-repo
| 1.1 kB    00:00 ...
localdb
| 951 B     00:00 ...
localdb/primary
| 886 B     00:00 ...
localdb
| 1/1
patching
| 951 B     00:00 ...
thirdparty
| 951 B     00:00 ...

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

Dependencies Resolved

```

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

## Transaction Summary

```

=====
Install          1 Package

Total size: 6.4 M
Installed size: 6.4 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : vxlan-2.0.1.0-9.2.1.lib32_n9000                1/1

starting pre-install package version mgmt for vxlan
pre-install for vxlan complete
starting post-install package version mgmt for vxlan
post-install for vxlan complete

Installed:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1

```

Complete!

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

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

```

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

```

```

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

```

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 Repository
Version Size
-----
Updating:
  vxlan lib32_n9000
2.0.1.0-9.2.1 /vxlan-2.0.1.0-9.2.1.lib32_n9000

```

6.4 M

## Transaction Summary

Upgrade 1 Package

Total size: 6.4 M

Is this ok [y/N]: y

Downloading Packages:

Running Transaction Check

Running Transaction Test

Transaction Test Succeeded

Running Transaction

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:

**dnf 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# dnf upgrade
http://10.0.0.2/modularity/rpms/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm

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

Setting up Upgrade Process
vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm | 1.6 MB 00:00
Examining /var/tmp/yum-root-RaANgb/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm:
vxlan-2.0.1.0-9.2.1.lib32_n9000
Marking /var/tmp/yum-root-RaANgb/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm as an update to
vxlan-2.0.0.0-9.2.1.lib32_n9000
Resolving Dependencies
--> Running transaction check

```

```

---> Package vxlan.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
--> Finished Dependency Resolution

```

Dependencies Resolved

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

Transaction Summary

Upgrade 1 Package

Total size: 6.4 M

Is this ok [y/N]: y

Downloading Packages:

Running Transaction Check

Running Transaction Test

Transaction Test Succeeded

Running Transaction

\*\* Found 1 pre-existing rpmdb problem(s), 'yum check' output follows:

busybox-1.23.2-r0.0.x86\_64 has missing requires of busybox-syslog

Updating : vxlan-2.0.1.0-9.2.1.lib32\_n9000

1/2

starting pre-install package version mgmt for vxlan

pre-install for vxlan complete

starting post-install package version mgmt for vxlan

post-install for vxlan complete

Cleanup : vxlan-2.0.0.0-9.2.1.lib32\_n9000

2/2

Updated:

vxlan.lib32\_n9000 0:2.0.1.0-9.2.1

Complete!

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

```
sudo dnf upgrade /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm
```

```

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

```

```

localdb          | 1.1 kB    00:00 ...

```

```

patching        | 951 B     00:00 ...

```

```

thirdparty     | 951 B     00:00 ...

```

```

                | 951 B     00:00 ...

```

Setting up Upgrade Process

Examining /bootflash/vxlan-2.0.1.0-9.2.1.lib32\_n9000.rpm: vxlan-2.0.1.0-9.2.1.lib32\_n9000

```

Marking /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm as an update to
vxlan-2.0.0.0-9.2.1.lib32_n9000
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
--> Finished Dependency Resolution

```

Dependencies Resolved

| Package<br>Version                  | Arch<br>Size         | Repository                       |
|-------------------------------------|----------------------|----------------------------------|
| Updating:<br>vxlan<br>2.0.1.0-9.2.1 | lib32_n9000<br>6.4 M | /vxlan-2.0.1.0-9.2.1.lib32_n9000 |

Transaction Summary

Upgrade 1 Package

```

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

```

1/2

```

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

```

2/2

```

Updated:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1

```

Complete!

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

```
dnf upgrade eigrp
```

## Downgrading the RPMs

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

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



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

```
localdb          | 1.1 kB    00:00 ...
localdb/primary  | 951 B    00:00 ...
localdb          | 1.3 kB    00:00 ...
                2/2
patching
thirdparty      | 951 B    00:00 ...
                | 951 B    00:00 ...
```

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

Dependencies Resolved

---

| Package             | Version       | Size  | Arch        | Repository  |
|---------------------|---------------|-------|-------------|-------------|
| Downgrading:        |               |       |             |             |
| vxlan               | 2.0.0.0-9.2.1 | 1.6 M | lib32_n9000 | groups-repo |
| Transaction Summary |               |       |             |             |

---

Downgrade 1 Package

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

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

```
2/2
Removed:
vxlan.lib32_n9000 0:2.0.1.0-9.2.1
```

```
Installed:
  vxlan.lib32_n9000 0:2.0.0.0-9.2.1
```

```
Complete!
```

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

```
dnf 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:

```
dnf downgrade eigrp
```

## Deleting the RPMs

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

```
bash-4.2# sudo dnf erase vxlan
```

```
Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching,
protect-packages
Setting up Remove Process
Resolving Dependencies
--> Running transaction check
---> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be erased
--> Finished Dependency Resolution
```

```
Dependencies Resolved
```

| Package   | Arch        | Repository                        | Version       | Size  |
|-----------|-------------|-----------------------------------|---------------|-------|
| Removing: |             |                                   |               |       |
| vxlan     | lib32_n9000 | @/vxlan-2.0.1.0-9.2.1.lib32_n9000 | 2.0.1.0-9.2.1 | 6.4 M |

```
Transaction Summary
```

```
Remove      1 Package
```

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

```
1/1
```

```
starting pre-remove package version mgmt for vxlan
```

```
pre-remove for vxlan complete

Removed:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1

Complete!
```

## Support for DNF Groups

The support for DNF 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. DNF supports the following group commands:

- `groupinstall`
- `groupinfo`
- `groupinstall`
- `groupremove`
- `groupupdate`

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

### Using the `groupinstall` Command

In Linux, number of packages are bundled to particular group. Instead of installing individual packages with `dnf`, you can install particular group that will install all the related packages that belongs to the group. For example to list all the available groups, use the `dnf groupinstall` command:

```
bash-4.4# dnf groupinstall
Last metadata expiration check: 0:00:00 ago on Fri 08 Mar 2024 12:26:33 PM UTC.
] --- B/s | 0 B    --:-- ETA
Available Groups:
  management
  routing
  L2
  L3
bash-4.4#
```

### Using the `groupmembers` Command

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

```
bash-4.4# dnf groupinfo l2
Last metadata expiration check: 0:00:00 ago on Fri 08 Mar 2024 12:27:44 PM UTC.
] --- B/s | 0 B    --:-- ETA

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

```

vtp
bash-4.4#

```

## Using the groupinstall Command

This command is for both install and 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-4.4# dnf groupinstall l3
Last metadata expiration check: 0:00:00 ago on Fri 08 Mar 2024 12:38:05 PM UTC.
] --- B/s | 0 B    --:-- ETA
Not a redundant system. Nothing todo
Dependencies resolved.

```

---

| Group | Packages |
|-------|----------|
|-------|----------|

---

Marking packages as installed by the group:

|     |     |
|-----|-----|
| @L3 | bfd |
|-----|-----|

Is this ok [y/N]: y

Complete!

Install operation 10 completed successfully at Fri Mar 8 12:38:08 2024.

[#####] 100%

## Using the groupupdate Command

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

```

bash-4.4# dnf groupupdate l3

```

```

Last metadata expiration check: 0:00:00 ago on Wed 13 Mar 2024 12:30:11 PM UTC.

```

```

] --- B/s | 0 B    --:-- ETA

```

Dependencies resolved.

---

| Group | Packages |
|-------|----------|
|-------|----------|

---

Marking packages as installed by the group:

|     |     |
|-----|-----|
| @L3 | bfd |
|-----|-----|

---

| Package | Arch<br>Repository | Size | Version |
|---------|--------------------|------|---------|
|---------|--------------------|------|---------|

---

Installing group packages:

|     |                               |       |                |
|-----|-------------------------------|-------|----------------|
| bfd | lib32_64_n9000<br>groups-repo | 562 k | 2.0.0.0-10.4.3 |
|-----|-------------------------------|-------|----------------|

Transaction Summary

Install 1 Package

Total size: 562 k

Installed size: 2.3 M

Is this ok [y/N]: y

Downloading Packages:

Running transaction check

Transaction check succeeded.

```

Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                               1/1

  Running scriptlet: bfd-2.0.0.0-10.4.3.lib32_64_n9000
                                                    1/1
starting pre-install package version mgmt for bfd
pre-install for bfd complete
  Installing      : bfd-2.0.0.0-10.4.3.lib32_64_n9000
                                                    1/1

  Running scriptlet: bfd-2.0.0.0-10.4.3.lib32_64_n9000
                                                    1/1
starting post-install package version mgmt for bfd
post-install for bfd complete
  Verifying      : bfd-2.0.0.0-10.4.3.lib32_64_n9000
                                                    1/1

Installed:
  bfd.lib32_64_n9000 2.0.0.0-10.4.3

Complete!
Install operation 14 completed successfully at Wed Mar 13 12:30:23 2024.

[#####] 100%
bash-4.4#

```

## Using the grouperase Command

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

```

bash-4.4# dnf grouperase l3
Dependencies resolved.

```

```

=====
Group                                     Packages
=====
Marking packages as removed by the group:
@L3                                       bfd
=====

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

Transaction Summary
=====
Remove 1 Package

Freed space: 2.3 M
Is this ok [y/N]: y
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                               1/1

  Running scriptlet: bfd-2.0.0.0-10.4.3.lib32_64_n9000

```

```

starting pre-remove package version mgmt for bfd
pre-remove for bfd complete
  Erasing          : bfd-2.0.0.0-10.4.3.lib32_64_n9000
                   1/1
  Running scriptlet: bfd-2.0.0.0-10.4.3.lib32_64_n9000
                   1/1
starting post-remove package version mgmt for bfd
post-remove for bfd complete

  Verifying       : bfd-2.0.0.0-10.4.3.lib32_64_n9000
                   1/1

Removed:
  bfd.lib32_64_n9000 2.0.0.0-10.4.3
                                           Complete!
Install operation 11 completed successfully at Fri Mar  8 12:38:41 2024.

[#####] 100%
bash-4.4#

```

## 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# dnf repolist all

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

localdb          | 1.1 kB    00:00 ...
patching         | 951 B     00:00 ...
thirdparty      | 951 B     00:00 ...
repo id          | 951 B     00:00 ...
repo name
status
groups-repo
  Groups-RPM Database
enabled: 37
localdb
  Local RPM Database
enabled: 6
patching
  Patch-RPM Database
enabled: 0
thirdparty
  Thirdparty RPM Database
enabled: 0
open-nxos
  open-nxos
disabled
repolist: 43

```

## Finding the Installed DNF Version

See the following example for listing the installed DNF version:

```
dnf --version
```

```
3.4.3
  Installed: rpm-5.4.14-r0.0.x86_64 at 2018-06-02 13:04
  Built    : Wind River <info@windriver.com> at 2018-04-27 08:36
  Committed: Wind River <info@windriver.com> at 2018-04-27

  Installed: yum-3.4.3-r9.0.x86_64 at 2018-06-02 13:05
  Built    : Wind River <info@windriver.com> at 2018-04-27 08:36
  Committed: Wind River <info@windriver.com> at 2018-04-27
```

## Mapping the NX-OS CLI to the DNF Commands

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

*Table 6: Patching Command Reference*

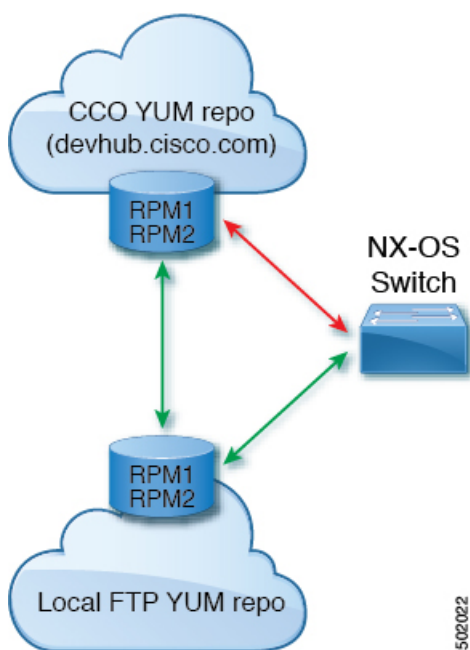
| <b>NX-OS CLI Commands</b>     | <b>DNF Commands</b>   |
|-------------------------------|---|
| <b>show install inactive</b>  | <b>dnf list --patch-only available</b>  |
| <b>show install active</b>    | <b>dnf list --patch-only installed</b>  |
| <b>show install committed</b> | <b>dnf list --patch-only committed</b>  |
| <b>show install packages</b>  | <b>dnf list --patch-only</b>  |
| <b>show install pkg-info</b>  | <b>dnf info --patch-only</b>  |
| <b>show install log</b>       | <b>dnf history --show-patch-log</b><br>where log_cmd: <ul style="list-style-type: none"> <li>• opid= - Log that is specific to an operation ID.</li> <li>• last - Shows the latest operation log.</li> <li>• reverse – Shows the log in reverse order.</li> <li>• detail – Show detailed log.</li> <li>• from= - Shows logging from a specific operation ID.</li> </ul> |
| <b>clear install log</b>      | <b>dnf history --clear-patch-log=</b><br>where clear_log_cmd: <ul style="list-style-type: none"> <li>• all - Clears the complete log.</li> <li>• - Clears the logs above this operation ID.</li> </ul>  |
| <b>install add</b>            | <b>dnf install --add bootflash:/</b>  |

| NX-OS CLI Commands      | DNF Commands  |
|-------------------------|---|
| install remove          | <code>dnf install --remove</code>   |
| install remove inactive | <code>dnf install --remove all</code>   |
| install activate        | <code>dnf install --no-persist --nocommit</code><br><b>Note</b> By default, all packages are activated and committed. |
| install deactivate      | <code>dnf erase --nocommit</code><br><b>Note</b> By default, all packages are de-activated and committed.             |
| install commit          | <code>dnf install --commit</code>   |
| Install commit          | <code>dnf install --commit all</code>   |

## 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 2: Configuring an FTP server and Setting up a Local FTP YUM Repository*



502022





**Note** For Cisco NX-OS Release 10.1(1), visit <https://devhub.cisco.com/artifactory/open-nxos/10.1.1/> 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):

### SUMMARY STEPS

1. `dnf install vsftpd`
2. `systemctl start vsftpd`
3. `systemctl status vsftpd`
4. `firewall-cmd --zone=public --permanent --add-port=21/tcp`
5. `firewall-cmd --zone=public --permanent --add-service=ftp`
6. `firewall-cmd --reload`
7. `wget ftp:// <ip of FTP server> /test.txt`

### DETAILED STEPS

|        | Command or Action   | Purpose   |
|--------|---|---|
| Step 1 | <code>dnf install vsftpd</code>                                       | Installs vsftpd, an FTP server.   |
| Step 2 | <code>systemctl start vsftpd</code>                                   | Starts the FTP Server.  |
| Step 3 | <code>systemctl status vsftpd</code>                                  | Checks the status of the FTP Server.  |
| Step 4 | <code>firewall-cmd --zone=public --permanent --add-port=21/tcp</code> | Allows access to the FTP services from the external systems and opens port 21.  |
| Step 5 | <code>firewall-cmd --zone=public --permanent --add-service=ftp</code> | Adds the FTP service.   |
| Step 6 | <code>firewall-cmd --reload</code>                                    | Reloads the server.   |
| Step 7 | <code>wget ftp:// &lt;ip of FTP server&gt; /test.txt</code>           | Hosts a file in the FTP server (for example, test.txt) and attempts Wget of that file.<br><br><b>Note</b> Note that <code>/var/ftp/</code> 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:

## SUMMARY STEPS

1. **touch /etc/yum.repos.d/local.repo**
2. **vim /etc/yum.repos.d/local.repo**
3. **cat /etc/yum.repos.d/local.repo**
4. **dnf repolist**
5. **nohup reposync -r <repo-name mentioned in the local.repo> -p <directory path to sync>&**
6. **tail -f nouhup.out**

## DETAILED STEPS

|               | Command or Action   | Purpose   |
|---------------|---|---|
| <b>Step 1</b> | <b>touch /etc/yum.repos.d/local.repo</b><br><b>Example:</b><br><pre>bash-4.3#touch /etc/yum.repos.d/local.repo</pre>  | Creates a repository file under <b>/etc/yum.repos.d/</b> , for example, creates <b>local.repo</b> repository and adds the base URL. |
| <b>Step 2</b> | <b>vim /etc/yum.repos.d/local.repo</b><br><b>Example:</b><br><pre>bash-4.3#vim /etc/yum.repos.d/local.repo  [localrepo] name=localrepo baseurl= https://devhub.cisco.com/artifactory/open-nxos/7.0-3-I2-1/x86_64/ enabled=1 gpgcheck=0 sslverify=0</pre>  | Edit the repository file and copy the localrepo details.<br><b>Note</b> Modify the base URL to the required repository URL.         |
| <b>Step 3</b> | <b>cat /etc/yum.repos.d/local.repo</b><br><b>Example:</b><br><pre>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</pre>  | Verify the local repository data to proceed further.  |
| <b>Step 4</b> | <b>dnf repolist</b><br><b>Example:</b><br><pre>bash-4.3# dnf repolist Loaded plugins: fastestmirror, langpacks Loading mirror speeds from cached hostfile * base: mirror.dhakacom.com * extras: mirror.dhakacom.com * updates: mirror.dhakacom.com repo id repo name status base/7/x86_64 CentOS-7 - Base 9,911 extras/7/x86_64 CentOS-7 - Extras 313 localrepo localrepo 687</pre> | Checks the reachability of the repository.  |

|               | Command or Action  | Purpose  |
|---------------|--|--|
|               | updates/7/x86_64 CentOS-7 - Updates 711<br>repolist: 11,622  |  |
| <b>Step 5</b> | <p><b>nohup reposync -r</b> &lt;repo-name mentioned in the local.repo&gt; <b>-p</b> &lt;directory path to sync&gt;&amp;</p> <p><b>Example:</b></p> <pre>nohup reposync -r localrepo -p /var/ftp/ &amp;</pre> <p>This command creates a directory with the name <b>local.repo</b> inside <b>/var/ftp/</b> and downloads all the packages from <b>devhub.cisco.com</b> to the directory.</p> | Synchronizes all the packages from the external repository to the FTP server home directory. |
| <b>Step 6</b> | <b>tail -f nouhup.out</b>  | 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:

### SUMMARY STEPS

1. **run bash sudo su**
2. **ip netns exec management ping** <ip\_address>
3. **touch /etc/yum/repos.d/ftp.repo**
4. **vim /etc/yum/repos.d/ftp.repo**
5. **cat /etc/yum/repos.d/ftp.repo**
6. **ip netns exec management bash**
7. **dnf repolist**
8. **dnf list available**

### DETAILED STEPS

|               | Command or Action   | Purpose  |
|---------------|---|--|
| <b>Step 1</b> | <b>run bash sudo su</b>   | Logs in as a sudo user.  |
| <b>Step 2</b> | <b>ip netns exec management ping</b> <ip_address>   | Checks the reachability of the FTP server address from the switch using the <b>ping</b> command.                                     |
| <b>Step 3</b> | <p><b>touch /etc/yum/repos.d/ftp.repo</b></p> <p><b>Example:</b></p> <pre>bash-4.3#touch /etc/yum/repos.d/ftp.repo</pre>                                | Creates a repository file under <b>/etc/yum/repos.d/</b> , for example, creates <b>ftp.repo</b> repository.                          |
| <b>Step 4</b> | <p><b>vim /etc/yum/repos.d/ftp.repo</b></p> <p><b>Example:</b></p> <pre>bash-4.3#vim /etc/yum/repos.d/ftp.repo</pre> <pre>[ftp] name=ftp baseurl=</pre> | <p>Edit the repository file and copy the ftp repo details.</p> <p><b>Note</b> Modify the base URL to the required ftp server IP.</p> |

|               | Command or Action  | Purpose   |
|---------------|--|---|
|               | ftp://198.51.100.1/localrepo/<br>enabled=1<br>gpgcheck=0<br>sslverify=0  |   |
| <b>Step 5</b> | <b>cat /etc/yum/repos.d/ftp.repo</b><br><b>Example:</b><br><pre>bash-4.3# cat /etc/yum/repos.d/ftp.repo [ftp] name=ftp baseurl=ftp://198.51.100.1/localrepo/ enabled=1 gpgcheck=0 sslverify=0</pre>  | Creates a repository file on the switch with the FTP server address as the URL. |
| <b>Step 6</b> | <b>ip netns exec management bash</b>   | Uses the Bash shell prompt.   |
| <b>Step 7</b> | <b>dnf repolist</b><br><b>Example:</b><br><pre>bash-4.3# dnf repolist Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching, : protect-packages groups-repo   1.1 kB 00:00 ... localdb   951 B 00:00 ... patching   951 B 00:00 ... thirdparty   951 B 00:00 ... thirdparty/primary   758 B 00:00 ... thirdparty 1/1 repo id repo name status groups-repo Groups-RPM Database 37 localdb Local RPM Database 0 patching Patch-RPM Database 0 thirdparty Thirdparty RPM Database 1 ftp ftp 686 repolist: 724</pre> | Checks the reachability of newly created repository.                            |
| <b>Step 8</b> | <b>dnf list available</b>  | 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/nxos64.10.1.1.bin
bootflash:nxos64.10.1.1.bin compact vrf management use-kstack
```

```
user1@10.65.42.196's password:
nxos64.10.1.1.bin 100% 1501MB 8.4MB/s 02:58
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

# 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.



**Note** If you need to convert the mode of operation from NX-OS to ACI and the NX-OS switch model has only 16G memory, then the switch requires an RAM memory upgrade to support the ACI mode. For more information about upgrading the RAM of such switches, refer to [Adding an 8, 16, or 32 Gigabyte DIMM to a Cisco Nexus 9000 Series Switch](#).

This chapter contains the following sections:

- [Converting to ACI Boot Mode, on page 77](#)
- [Converting a Replacement Standby Supervisor to ACI Boot Mode, on page 79](#)
- [Converting Back to Cisco NX-OS, on page 80](#)

## Converting to ACI Boot Mode

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

### 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(2x) or a later release.

Use the **show install all impact epld** *epld-image-name* command to verify that the switch does not require any EPLD image upgrades. If any upgrades are required, follow the instructions in the [Cisco Nexus 9000 Series FPGA/EPLD Upgrade Release Notes](#).

## SUMMARY STEPS

1. Verify that the switch is running the latest release.
2. Follow these steps to copy the ACI image from the APIC:
3. Follow these steps to boot to the ACI image:
4. Verify whether you must install certificates for your device.

## DETAILED STEPS

**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.

**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.

### SUMMARY STEPS

1. Reload the switch.
2. Enter a break sequence (Ctrl-C or Ctrl-]) during the initial boot sequence to access the loader> prompt.
3. Configure the boot process to stop at the switch(boot)# prompt.
4. Boot the active supervisor module with the Cisco NX-OS image.

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.
6. Completes the upload of the Cisco NX-OS image file.
7. Recopy the Cisco NX-OS image into the bootflash: and set the appropriate boot variables to ensure that the system boots the Cisco NX-OS image on the next reload.
8. Wait for the system controllers to come up, which could take approximately 15–20 minutes.
9. Verify that the active supervisor module and the system controllers are in the active state.
10. For dual-supervisor systems, follow Steps 3–6 on the standby supervisor.
11. Log in to the switch and verify that it is running Cisco NX-OS software.

## DETAILED STEPS

**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 that is mentioned in the boot variable is not present in the bootflash, the system falls back to the loader prompt during the boot sequence. To recover the switch from the loader prompt, boot the system through a different image present in the bootflash, perform a **tftpboot**, or boot through a USB device.

**Note** For some 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 Cisco 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 a USB method.

- For the TFTP method - First assign an 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 the USB method - Mount the USB on the switch and execute the **dir** command on the loader to see the contents of the bootflash folder and the USB device.

**Example**

```
loader > dir
usb1::
lost+found
/nxos.9.x.y.bin
```

Then boot the NX-OS image using the following command:

```
loader> boot usb1:/nxos-image
Example: boot usb1:/nxos.9.x.y.bin
```

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

**Step 7** Recopy the Cisco NX-OS image into the 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 15–20 minutes.
- File system differences between ACI and Cisco NX-OS require one-time reformatting change during the ACI to Cisco NX-OS conversion. Subsequent reloads with the Cisco NX-OS image are faster.
- Step 9** Verify that the active supervisor module and the system controllers are in the active state.
- Example:**
- ```
switch# show module
```
- Step 10** For dual-supervisor systems, follow Steps 3–6 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.

---

- 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 5

# 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, on page 85](#)
- [vPC Upgrade and Downgrade Procedure for Nexus 9000 -R series switches, on page 85](#)

## 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.





## INDEX

### B

boot [80](#)  
boot aci bootflash [79](#)

### C

copp profile [11](#)  
copy [32, 36, 80](#)  
copy bootflash: [78](#)

### D

delete bootflash [32, 36](#)  
dir bootflash [31, 36](#)

### F

features scp-server [78](#)

### G

guestshell destroy [18](#)

### I

init system [82](#)  
install all nxos bootflash [33, 37](#)

### L

load-nxos [82](#)

### N

no boot nxos [78](#)

### P

ping [6](#)  
poweroff module [37, 77](#)

### R

reload [79–80](#)

### S

setup [11](#)  
show configuration session summary [6](#)  
show file bootflash [32, 79](#)  
show incompatibility nxos bootflash: [7](#)  
show incompatibility-all nxos bootflash [36](#)  
show install all impact epld [78](#)  
show install all impact nxos bootflash [33, 37](#)  
show install all status [34, 37](#)  
show module [77, 83](#)  
show version [34, 37, 78](#)

