Cisco Nexus 3600 Series NX-OS Software Upgrade and Downgrade Guide, Release 9.2(x)

First Published: 2018-07-16

Last Modified: 2020-08-20

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 NONINFRINGEMENT 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.

CONTENTS

PREFACE

CHAPTER 1

CHAPTER 2

CHAPTER 3

CHAPTER 4

Preface

This preface includes the following sections:

Audience

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

Document Conventions

Command descriptions use the following conventions:

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

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

Examples use the following conventions:

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

Related Documentation for Cisco Nexus 9000 Series Switches

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

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

Documentation Feedback

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

Communications, Services, and Additional Information

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

Cisco Bug Search Tool

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

CHAPTER 1

New and Changed Information

This chapter provides release-specific information for each new and changed feature in the *Cisco Nexus 3600 Series NX-OS Software Upgrade and Downgrade Guide, Release 9.x*.

New and Changed Information

This table summarizes the new and changed features for the *Cisco Nexus 3600 Series NX-OS Software Upgrade and Downgrade Guide, Release 9.x* and tells you where they are documented.

Table 1: New and Changed Features for Cisco NX-OS Release 9.x

Feature	Description	Changed in Release	Where Documented
NX-OS Optionality	Added support for NX-OS optionality.	9.2(1)	Optionality in Cisco NX-OS Software, on page 17
vPC Upgrade and Downgrade	Added the updated procedure for vPC upgrade and downgrade.	9.2(1)	vPC Upgrade and Downgrade Procedure for Nexus 9000 -R series switches, on page 11
Upgrade Process	Revised the procedure to upgrade the Cisco NX-OS software.	9.2(1)	Upgrading the Cisco NX-OS Software, on page 6

CHAPTER 2

Upgrading or Downgrading the Cisco Nexus 3600 Series NX-OS Software

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

About the Software Image

Each device is shipped with the Cisco NX-OS software. The Cisco NX-OS software consists of one NXOS software image. The image filename begins with "nxos".

Only this image is required to load the Cisco NX-OS operating system. This image runs on all Cisco Nexus 3600 Series switches.

Note Another type of binary file is the software maintenance upgrade (SMU) package file. SMUs contain fixes for specific defects. They are created to respond to immediate issues and do not include new features. SMU package files are available for download from Cisco.com and generally include the ID number of the resolved defect in the filename. For more information on SMUs, see the Cisco Nexus 3600 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.

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.

Cisco NX-OS Software Upgrade Guidelines

Note The Cisco Nexus 3600 Series NX-OS Release Notes contain specific upgrade guidelines for each release. See the Release Notes before starting the upgrade.

The following upgrade paths are supported for upgrading from an earlier release to a Cisco NX-OS 9.2(x) release:

- Cisco NX-OS Release 7.0(3)F3(3)/(3c) to Cisco NX-OS Release 7.0(3)F3(4) to Cisco NX-OS Release 9.2(x)
- Cisco NX-OS Release 7.0(3)F3(5) to Cisco NX-OS Release 9.2(x)
- Cisco NX-OS Release 9.2(x) to Cisco NX-OS Release 9.2(x)

Note For information on upgrading in a vPC environment, see vPC Upgrade and Downgrade Procedure for Nexus 9000 -R series switches, on page 11.

iCAM must be disabled before upgrading from Cisco NX-OS Release 7.0(3)I7(1) to Cisco NX-OS Release 9.2(x) or 9.3(x). Only upgrading from Cisco NX-OS Release 9.2(4) to Cisco NX-OS Release 9.3(1) can be performed if iCAM is enabled.

To upgrade from any release prior to Cisco NX-OS Release 7.0(3)F3(3c), you must backup the switch configuration, perform a write erase, and reload the device. To upgrade from Cisco NX-OS Release 7.0(3)F3(4) to any later release, we recommend that you use the **install all** command.

Note Upgrading from Cisco NX-OS Release 7.0(3)F3(4) to 9.2(x) or later releases may take a longer time to boot due to the ASCII configuration being replayed. Please perform a **copy running-config startup-config** after upgrading to 9.2(x) or higher releases to avoid long boot times during future reloads.

Beginning with Cisco NX-OS Release 9.2(1), a simplified NX-OS numbering format is used for the platforms that are supported in the release. In order to support a software upgrade from previous releases that have the old release format, an installer feature supplies an I9(x) label as a suffix to the actual release during the upgrade operation. This label is printed as part of the image during the upgrade operation from any prior release to a Cisco NX-OS 9.2(x) release, and it can be ignored.

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

- 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.
- 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 3600 Series NX-OS Security Configuration Guide.
- 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 3600 Series NX-OS System Management Configuration Guide.
- By default, the software upgrade process is disruptive.
- For a Cisco Nexus 36180YC-R switch with configured egress RACLs, prior to upgrading from a 7.x release to a 9.x release, follow these steps to ensure the RACLs are maintained and the upgrade is completed without issue:

  1. Add TCAM entries for egress RACL using the **hardware access-list tcam region e-racl** command.
  2. Save the configuration and reload.
  3. Upgrade to a 9.x release.

  For more information about configuring TCAM regions, see the *Cisco Nexus 3600 NX-OS Security Configuration Guide*.

Prerequisites for Upgrading the Cisco NX-OS Software

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

- 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.
- Ensure that the device has a route to the remote server. The device and the remote server must be in the same subnetwork if you do not have a router to route traffic between subnets. To verify connectivity to the remote server, use the **ping** command.

```
switch# ping 172.18.217.1 vrf management
PING 172.18.217.1 (172.18.217.1): 56 data bytes
64 bytes from 172.18.217.1: icmp_seq=0 ttl=239 time=106.647 ms
64 bytes from 172.18.217.1: icmp_seq=1 ttl=239 time=76.807 ms
64 bytes from 172.18.217.1: icmp_seq=2 ttl=239 time=76.593 ms
64 bytes from 172.18.217.1: icmp_seq=3 ttl=239 time=81.679 ms
64 bytes from 172.18.217.1: icmp_seq=4 ttl=239 time=76.5 ms
```

```
--- 172.18.217.1 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 3000 Series NX-OS System Management Configuration Guide*.

Upgrading the Cisco NX-OS Software

Use this procedure to upgrade from a Cisco NX-OS 9.2(x) release to a later 9.2(x) release or from Cisco NX-OS Release 7.0(3)F3(4) to a Cisco NX-OS 9.2(x) release.

Procedure

Step 1 **Read the release notes for the software image file for any exceptions to this upgrade procedure.** See the Cisco Nexus 3600 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:
```

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 supervisor module, delete unnecessary files to make space available.

```
switch# delete bootflash:nxos.7.0.3.F3.3.bin
```

Step 5 Verify that there is space available on the active and the standby supervisor modules.

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

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

Step 8 Copy the software image to the active supervisor module using a transfer protocol. You can use FTP, TFTP, SCP, or SFTP.

```
switch# copy scp://user@scpserver.cisco.com//download/nxos.9.2.2.bin bootflash:nxos.9.2.2.bin
```

Step 9 Display the SHA256 checksum for the file to verify the operating system integrity and ensure that the downloaded image is safe to install and use.

```
switch# show file bootflash://sup-1/nxos.9.2.2.bin sha256sum
5214d563b7985ddad67d52658af573d6c64e5a9792b35c458f5296f954bc53be
```

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

```
switch# show install all impact nxos bootflash:nxos.9.2.2.bin
```

Step 11 Save the running configuration to the startup configuration.

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

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

```
switch# install all nxos bootflash:nxos.9.2.2.bin
```

The following options are available:

- **no-reload**—Exits the software upgrade process before the device is reloaded.
- **non-interruptive**—Upgrades the software without any prompts. This option skips all error and sanity checks.

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 13 (Optional) Display the entire upgrade process.

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

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

```
switch# show version
```

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

Cisco NX-OS Software Downgrade Guidelines

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

- Software downgrades from a Cisco NX-OS 9.2(x) release to an earlier 9.2(x) release or to Cisco NX-OS Release 7.0(3)F3(4) should be performed using the **install all** command.
- iCAM must be disabled before downgrading from Release 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.
- 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 3600 Series NX-OS System Management Configuration Guide.

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

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.

Downgrading to an Earlier Software Release

Use this procedure to downgrade from a Cisco NX-OS 9.2(x) release to an earlier 9.2(x) release or from a Cisco NX-OS 9.2(x) release to Cisco NX-OS Release 7.0(3)F3(4).

Procedure

Step 1 **Read the release notes for the software image file for any exceptions to this downgrade procedure.** See the Cisco Nexus 3600 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.

Step 5 Copy the software image to the active supervisor module using a transfer protocol. You can use FTP, TFTP, SCP, or SFTP.

```
switch# copy scp://user@scpserver.cisco.com//download/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.

```
switch# install all nxos bootflash:nxos.9.2.1.bin
switch# install all nxos nxos.9.2.1.bin.CCO
Installer will perform compatibility check first. Please wait.
Installer is forced disruptive

Verifying image bootflash:/nxos.9.2.1.bin.CCO for boot variable "nxos".
[####################] 100% -- SUCCESS

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

Preparing "nxos" version info using image bootflash:/nxos.9.2.1.bin.CCO.
[####################] 100% -- SUCCESS

Preparing "bios" version info using image bootflash:/nxos.9.2.1.bin.CCO.
[####################] 100% -- SUCCESS

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

Notifying services about system upgrade.
2018 Jul 12 09:59:20 Bifrost_L3_Snake %$ VDC-1 %$ %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configured
 from vty by admin on vsh.bin.30370
[####################] 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 9.2(2) 9.2(1) yes
1 bios v01.11(08/06/2018):v01.11(08/06/2018) v01.10(05/15/2018) no
```

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

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.

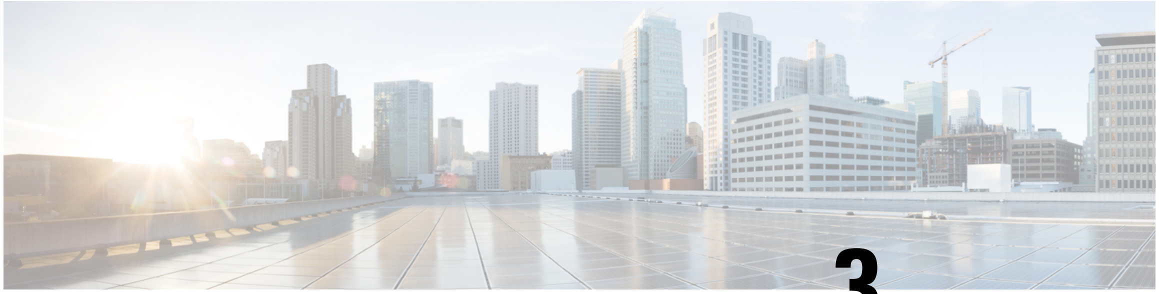
Step 12 (Optional) Display the entire downgrade process.

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
```



CHAPTER 3

Migrating Switches in a vPC Topology

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

vPC Forklift Upgrade

In a vPC topology, you can migrate from a pair of Cisco Nexus 3600 platform switches to a different pair of Cisco Nexus 3600 platform switches. For more information, see the "vPC Forklift Upgrade Scenario" section in the *Cisco Nexus 3600 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.

CHAPTER 4

Optionality in Cisco NX-OS Software

This chapter describes optionality in Cisco NX-OS software.

Optionality in Cisco NX-OS Software

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

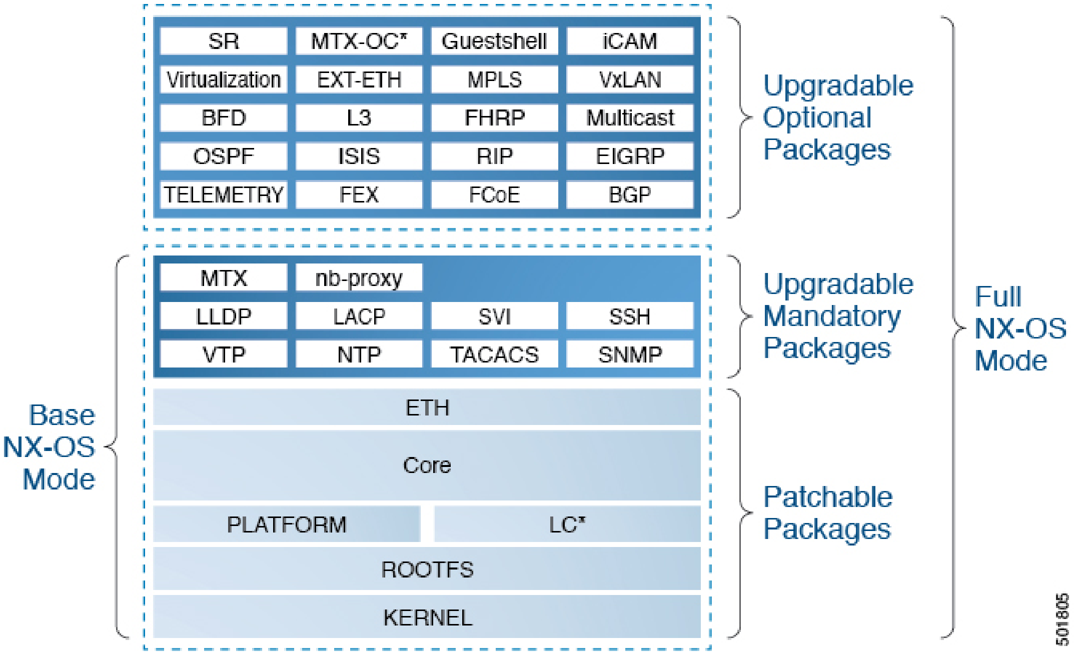
The advantages for using modular Cisco NX-OS software are:

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

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

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

Figure 1: Optionality in Cisco NX-OS Software



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

Note The default mode is full NX-OS mode.

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

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

Guidelines for Cisco NX-OS Patchable Packages/RPM Installation

For guidelines on Cisco NX-OS patchable packages/RPM installation (Release 7.x feature), see the Performing Software Maintenance Upgrades section in the Cisco Nexus 9000 Series NX-OS System Management Configuration Guide.

Using Modular Packages

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

This section describes a new package management for the Cisco NX-OS software image. Beginning with Cisco NX-OS Release 9.2(1), some NXOS features are considered as optional, for example, BGP, OSPF, VXLAN, MPLS, Segment Routing.

Each modular package has the following important characteristics:

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

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

RPM and YUM

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

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

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

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

Booting the NX-OS Image in Base or Full Mode

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

To use the command line option, see the following steps:

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

For more information, see Using Install CLIs for Feature RPM Operation section.

Information About RPMs

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

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

Format of the RPM

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

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

See the following table for more information on the naming convention, for example, fex-2.0.0.0-9.2.1.lib32_n9000.rpm:

Table 2: RPM Naming Convention

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

RPM Naming Convention Example: fex-2.0.0.0-9.2.1.lib32_n9000.rpm	Description
0	Indicates the incremental API changes/CLI changes/Schema changes with backward compatibility. It is applicable to the new features on top of the existing capabilities. No configuration is lost during an upgrade.
0	Indicates a bug fix without any functionality change. No configuration is lost during an upgrade.
0	This number tracks how many times the component has changed during the development cycle of a release. This value will be 0 for all the release images.
9.2.1	Indicates the release number or the distribution version for the RPM. It aligns to the NVR format. Since the feature RPM is only applicable to a NXOS release, this field has NXOS release version number present.
lib32_n9000	Indicates the architecture type of the RPM.

Optional RPMs and Their Associated Features

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

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

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

Table 3: List of Optional RPMs and Their Associated Features

Package Name	Associated Features
BGP	feature bgp
BFD	feature bfd
Container-tracker	feature container-tracker
EIGRP	feature eigrp

Package Name	Associated Features
Ext-Eth	• feature openflow • feature evb • feature imp • feature netflow • feature sla_sender • feature sla_responder • feature sla twamp-server • feature sflow
FCoE	• feature-set fcoe • feature-set fcoe-npv
FEX	feature-set fex
FHRP	• feature hsrp • feature vrrpv3
iCAM	feature icam
ISIS	feature isis
MPLS	• feature mpls segment-routing • feature mpls evpn
Multicast	• feature pim • feature pim6 • feature msdp • feature ngmvpn
OSPF	• feature ospf • feature ospfv3
RIP	feature rip
Services	feature catena
SR	feature mpls segment-routing traffic-engineering
TELEMETRY	feature telemetry

Package Name	Associated Features
Virtualization	NA
VXLAN	• feature nv overlay • feature fabric forwarding

Guidelines for NX-OS Feature RPM Installation

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

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

Table 4: RPM Repositories That Are Present in the Switches

Repository Name	Repository Path	Description
groups-repo	/rpms	Part of the bundled NX-OS image. It is used to keep all the RPMs that are bundled as part of the NX-OS image. All RPMs based in this repository are known as base RPMs.

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

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

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

- Only RPMs with the same NX-OS release number should be selected for the installation.
- Base RPMs cannot be added to the **localdb** repository.

Using Install CLIs for Feature RPM Operation

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

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

CLI	Description

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

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

CLI	Description
sh install inactive	Displays the list of the RPMs that are present in the repository but they are not installed.
sh install packages	Lists all the RPMs that are installed including rootfs RPMs.

Using Install CLIs for Digital Signature Support

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

Procedure

	Command or Action	Purpose

Step 1	switch#**install add bootflash**:<*keyfile*> **gpg-key**
	Example:
	`install add`
	`bootflash:RPM-GPG-KEY-puppetlabs gpg-key`
	`[####################] 100%`
	`Install operation 304 completed`
	`successfully at Thu Jun 19 16:40:28 2018`
		Cisco release RPMs are signed with Cisco GPG (GNU Privacy Guard) key. The public GPG key is present at **/etc/pki/rpm-gpg/arm-Nexus9k-rel.gpg**. To add other public keys from different sources, use the steps in this section.

Step 2	switch#**install verify package** <*package-name*>	Verifies the package.

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

Querying All Installed RPMs

Complete the following step to query all the installed RPMs:

Procedure

	Command or Action	Purpose

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

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

	Command or Action	Purpose

	` Unsigned`
	`switch#`	

Installing the RPMs Using One Step Procedure

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

Procedure

	Command or Action	Purpose

Step 1	**install add** *<rpm>* **activate**
	Example:
	`switch# install add bootflash:chef.rpm activate`
	`Adding the patch (/chef.rpm)`
	`[####################] 100%`
	`Install operation 868 completed successfully at Tue May  8 11:20:10 2018`
	`Activating the patch (/chef.rpm)`
	`[####################] 100%`
	`Install operation 869 completed successfully at Tue May  8 11:20:20 2018`
		Installs and activates the RPM.

Example

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

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

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

Installing the RPMs Using Two Steps Procedure

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

Procedure

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td><b>Step 1</b></td><td><b>install add</b> <i>&lt;rpm&gt;</i>
<b>Example:</b>
<code>switch# <b>install add bootflash:vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm</b></code>
<code>[####################] 100%</code>
<code>Install operation 892 completed successfully at Thu Jun 7 13:56:38 2018</code>
<code>switch(config)# <b>sh install inactive | grep vxlan</b></code>
<code>vxlan-2.0.1.0-9.2.1.lib32_n9000</code></td><td>Installs the RPM.</td></tr>
<tr><td><b>Step 2</b></td><td><b>install activate</b> <i>&lt;rpm&gt;</i>
<b>Example:</b></td><td>Activates the RPM.</td></tr>
</table>

Example

```
switch#install activate vxlan

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

switch# show install active | grep vxlan

vxlan-2.0.0.0-9.2.1.lib32_n9000

switch# sh install inactive | grep vxlan

switch#
```

Upgrading the RPMs Using One Step

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

Procedure

	Command or Action	Purpose

Step 1	**install add** *<rpm>***activate upgrade**
	Example:
	`switch(config)# `**`install add bootflash:bgp-2.0.2.0-9.2.1.lib32_n9000.rpm activate upgrade`**
	
	`Adding the patch (/bgp-2.0.2.0-9.2.1.lib32_n9000.rpm)`
	`[####################] 100%`
	`Install operation 870 completed successfully at Tue May 8 11:22:30 2018`
	
	`Activating the patch (/bgp-2.0.2.0-9.2.1.lib32_n9000.rpm)`
	`[####################] 100%`
	`Install operation 871 completed successfully at Tue May 8 11:22:40 2018`
		Installs the RPM.

Example

```
switch(config)# show install active

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

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

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

Downgrading the RPMs

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

Procedure

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td><b>Step 1</b></td><td><b>install add</b> <i>&lt;rpm&gt;</i><b>activate downgrade</b>
<b>Example:</b>
<pre>
switch(config)# install add
bootflash:bgp-2.0.1.0-9.2.1.lib32_n9000.rpm
 activate downgrade

Adding the patch
(/bgp-2.0.1.0-9.2.1.lib32_n9000.rpm)
[####################] 100%
Install operation 872 completed
successfully at Tue May 8 11:24:43 2018

Activating the patch
(/bgp-2.0.1.0-9.2.1.lib32_n9000.rpm)
[####################] 100%
Install operation 873 completed
successfully at Tue May 8 11:24:52 2018
</pre></td><td>Downgrades the RPM.</td></tr>
</table>

Example

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

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

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

Removing the RPMs

See the following steps to remove the RPMs:

Procedure

	Command or Action	Purpose

Step 1	**install remove** *<rpm>*
	Example:
	`switch(config)#` **`show install inactive | grep vxlan`**
	`        vxlan-2.0.0.0-9.2.1.lib32_n9000`
	`switch(config)#` **`install remove vxlan`**
	`Proceed with removing vxlan? (y/n)?  [n]`
	` y`
	`[####################] 100%`
	`Install operation 890 Removal of base rpm package is not permitted  at Thu Jun  7 13:52:15 2018`
		Removes the RPM from the repository.

Information About YUM Commands

See the following sections for more information about YUM commands.

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

Performing Package Operations Using the YUM Commands

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

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

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

Finding the Base Version RPM of the Image

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

```
#ls /rpms

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

Checking the List of the Installed RPMs

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

```
bash-4.2# yum list installed | grep lib32_n9000

bfd.lib32_n9000                        2.0.0.0-9.2.1           @groups-repo
core.lib32_n9000                       2.0.0.0-9.2.1           installed
eth.lib32_n9000                        2.0.0.0-9.2.1           installed
guestshell.lib32_n9000                 2.0.0.0-9.2.1           @groups-repo
lacp.lib32_n9000                       2.0.0.0-9.2.1           installed
linecard2.lib32_n9000                  2.0.0.0-9.2.1           installed
lldp.lib32_n9000                       2.0.0.0-9.2.1           installed
mcast.lib32_n9000                      2.0.0.0-9.2.1           @groups-repo
mtx-device.lib32_n9000                 2.0.0.0-9.2.1           installed
mtx-grpc-agent.lib32_n9000             2.0.0.0-9.2.1           installed
mtx-infra.lib32_n9000                  2.0.0.0-9.2.1           installed
mtx-netconf-agent.lib32_n9000          2.0.0.0-9.2.1           installed
mtx-restconf-agent.lib32_n9000         2.0.0.0-9.2.1           installed
mtx-telemetry.lib32_n9000              2.0.0.0-9.2.1           installed
nbproxy.lib32_n9000                    2.0.0.0-9.2.1           installed
ntp.lib32_n9000                        2.0.0.0-9.2.1           installed
nxos-ssh.lib32_n9000                   2.0.0.0-9.2.1           installed
ospf.lib32_n9000                       2.0.0.0-9.2.1           @groups-repo
platform.lib32_n9000                   2.0.0.0-9.2.1           installed
```

```
snmp.lib32_n9000                       2.0.0.0-9.2.1           installed
svi.lib32_n9000                        2.0.0.0-9.2.1           installed
tacacs.lib32_n9000                     2.0.0.0-9.2.1           installed
tor.lib32_n9000                        2.0.0.0-9.2.0.77        installed
virtualization.lib32_n9000             2.0.1.0-9.2.1           @localdb
vtp.lib32_n9000                        2.0.0.0-9.2.1           installed
vxlan.lib32_n9000                      2.0.0.0-9.2.1           @groups-repo
...
```

Getting Details of the Installed RPMs

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

```
yum info vxlan

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

                     | 1.1 kB     00:00 ...
localdb

                     |  951 B     00:00 ...
patching

                     |  951 B     00:00 ...
thirdparty

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

Installing the RPMs

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

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

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

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

Dependencies Resolved

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

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

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

Installed:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1


Complete!
```

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

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

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

                     | 1.1 kB     00:00 ...
localdb

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

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

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

Dependencies Resolved

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

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

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

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

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

                                   2/2

Updated:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1



Complete!
```

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

```
yum install eigrp
```

Upgrading the RPMs

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

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

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

Dependencies Resolved

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

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

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

Updated:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1


Complete!
```

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

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

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

                     | 1.1 kB     00:00 ...
localdb

                     |  951 B     00:00 ...
patching

                     |  951 B     00:00 ...
thirdparty

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

Dependencies Resolved

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

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

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

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

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

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

                                   2/2

Updated:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1



Complete!
```

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

`yum upgrade eigrp`

Downgrading the RPMs

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

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

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

                     | 1.1 kB     00:00 ...
localdb

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

                     | 1.3 kB     00:00 ...
localdb

                                        2/2
patching

                     |  951 B     00:00 ...
thirdparty

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

Dependencies Resolved

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

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

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

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

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

                                   2/2

Removed:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1



Installed:
  vxlan.lib32_n9000 0:2.0.0.0-9.2.1



Complete!
```

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

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

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

```
yum downgrade eigrp
```

Deleting the RPMs

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

```
bash-4.2# sudo yum erase vxlan

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

```
--> Finished Dependency Resolution

Dependencies Resolved

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

 Package                   Arch                                              Version
                                 Repository                                           Size
=========================================================================================
Removing:
 vxlan                     lib32_n9000                                       2.0.1.0-9.2.1
                                 @/vxlan-2.0.1.0-9.2.1.lib32_n9000                     6.4 M
Transaction Summary
=========================================================================================
Remove        1 Package

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

                                   1/1
starting pre-remove package version mgmt for vxlan
pre-remove for vxlan complete

Removed:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1



Complete!
```

Support for YUM Groups

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

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

- grouplist
- groupinfo
- groupinstall
- groupremove
- groupupdate

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

Using the grouplist Command

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

```
bash-4.2# sudo yum grouplist

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

                     | 1.1 kB     00:00 ...
localdb

                     |  951 B     00:00 ...
patching

                     |  951 B     00:00 ...
thirdparty

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

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

bash-4.3$
```

Using the groupmembers Command

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

```
bash-4.2# sudo yum groupinfo l2


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

                     | 1.1 kB     00:00 ...
localdb

                     |  951 B     00:00 ...
patching

                     |  951 B     00:00 ...
thirdparty

                     |  951 B     00:00 ...

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

```
vtp
```

Using the groupinstall Command

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

```
bash-4.2# sudo yum groupinstall routing

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

                      | 1.1 kB     00:00 ...
localdb

                      |  951 B     00:00 ...
patching

                      |  951 B     00:00 ...
thirdparty

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

Dependencies Resolved

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

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

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

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

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

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

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

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

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

Complete!
```

Using the groupupdate Command

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

```
bash-4.3# yum groupupdate routing

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

                    | 1.1 kB     00:00 ...
localdb

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

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

```
                                        6/6
patching

                     |  951 B     00:00 ...
thirdparty

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

Dependencies Resolved

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

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

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

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

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

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

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

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

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

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

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

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

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

                                 10/10

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

Complete!
```

Using the grouperase Command

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

```
bash-4.3$ sudo yum grouperase routing

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

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

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

                     |  951 B     00:00 ...
thirdparty

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

Dependencies Resolved

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

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

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

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

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

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

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

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

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

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

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

Complete!
```

Finding Repositories

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

```
bash-4.3# yum repolist all

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

                     | 1.1 kB     00:00 ...
localdb

                     |  951 B     00:00 ...
patching

                     |  951 B     00:00 ...
thirdparty

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

```
                                 disabled
repolist: 43
```

Finding the Installed YUM Version

See the following example for listing the installed YUM version:

yum --version

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

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

Mapping the NX-OS CLI to the YUM Commands

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

Table 6: Patching Command Reference

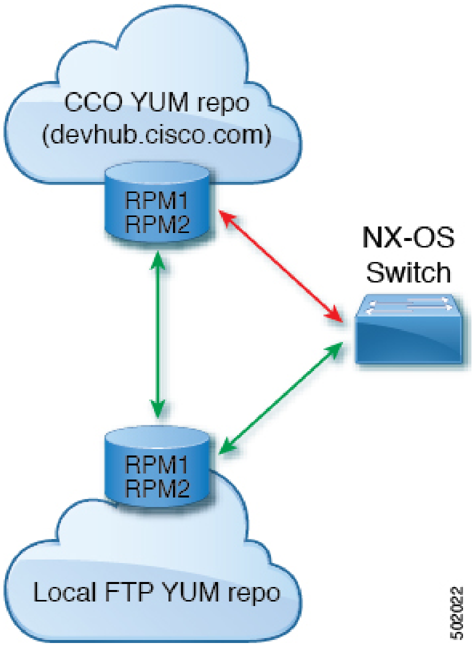
NX-OS CLI Commands	YUM Commands
show install inactive	**yum list --patch-only available**
show install active	**yum list --patch-only installed**
show install committed	**yum list --patch-only committed**
show install packages	**yum list --patch-only**
show install pkg-info	**yum info --patch-only**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

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

Configuring an FTP server and Setting up a Local FTP YUM Repository

For setting up a local FTP YUM repository, you have to first create an FTP server, create a local FTP YUM repository, and configure the Cisco NX-OS switch to reach the FTP server as outlined in the following illustration.

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



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

Creating an FTP Server on Red Hat Enterprise Linux 7 (RHEL7) Virtual Machine

Complete the following steps to create an FTP server on Red Hat Enterprise Linux 7 (RHEL7) Virtual Machine (VM):

Procedure

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

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

Creating a Local FTP YUM Repository

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

Procedure

	Command or Action	Purpose

Step 1	**cat /etc/yum.repos.d/local.repo**
	Example:
	`bash-4.3#`**`cat /etc/yum.repos.d/local.repo`**
	`[localrepo]`
	`name=localrepo`
	`baseurl=`
	`https://devhub.cisco.com/artifactory/open-nxos/7.0-3-I2-1/x86_64/`
	`enabled=1`
	`gpgcheck=0`
	`sslverify=0`
		Creates a repository file under **/etc/yum.repos.d/**, for example, creates **local.repo** repository and adds the base URL.

Step 2	bash-4.3#**yum repolist**
	Example:
	`bash-4.3# yum repolist`
	`Loaded plugins: fastestmirror, langpacks`
	`Loading mirror speeds from cached hostfile`
	`* base: mirror.dhakacom.com`
	`* extras: mirror.dhakacom.com`
	`* updates: mirror.dhakacom.com`
	`repo id repo name status`
	`base/7/x86_64 CentOS-7 - Base 9,911`
	`extras/7/x86_64 CentOS-7 - Extras 313`
	`localrepo localrepo 687`
	`updates/7/x86_64 CentOS-7 - Updates 711`
	`repolist: 11,622`
		Checks the reachability of the repository.

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

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

Configuring a Switch to Reach an FTP Server

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

Procedure

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

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

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

Step 5	**yum repolist**
	Example:
	`bash-4.3# yum repolist`
	`Loaded plugins: downloadonly,`
	`importpubkey, localrpmDB, patchaction,`
	`patching,`
	`: protect-packages`
	`groups-repo | 1.1 kB 00:00 ...`
	`localdb | 951 B 00:00 ...`
	`patching | 951 B 00:00 ...`
	`thirdparty | 951 B 00:00 ...`
	`thirdparty/primary | 758 B 00:00 ...`
	`thirdparty 1/1`
	`repo id repo name status`
	`groups-repo Groups-RPM Database 37`
	`localdb Local RPM Database 0`
	`patching Patch-RPM Database 0`
	`thirdparty Thirdparty RPM Database 1`
	`ftp ftp 686`
	`repolist: 724`
		Checks the reachability of newly created repository.

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

Creating User Roles for Install Operation

The **install** command is only available to the users of admin role. The **install** command can be available to a user by RBAC. See RBAC configuration guidelines for the same.