



Cisco Nexus 3600 シリーズ NX-OS ソフトウェア アップグレード/ダウングレードガイド、リリース 10.5(x)

最終更新：2025 年 11 月 24 日

シスコシステムズ合同会社

〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー

<http://www.cisco.com/jp>

お問い合わせ先：シスコ コンタクトセンター

0120-092-255（フリーコール、携帯・PHS含む）

電話受付時間：平日 10:00～12:00、13:00～17:00

<http://www.cisco.com/jp/go/contactcenter/>

【注意】シスコ製品をご使用になる前に、安全上の注意（www.cisco.com/jp/go/safety_warning/）をご確認ください。本書は、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

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

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

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

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

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

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

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

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

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

© 2024–2025 Cisco Systems, Inc. All rights reserved.



目次

はじめに :

はじめに vii

対象読者 vii

表記法 vii

Cisco Nexus 3600 プラットフォーム スイッチの関連資料 viii

マニュアルに関するフィードバック ix

通信、サービス、およびその他の情報 ix

第 1 章

新機能および変更された機能に関する情報 1

新機能および変更された機能に関する情報 1

第 2 章

Cisco Nexus 3600 シリーズ NX-OS ソフトウェアのアップグレードまたはダウングレード 3

ソフトウェア イメージ 3

Cisco NX-OS ソフトウェアのアップグレードの推奨事項 4

Cisco NX-OS ソフトウェア アップグレードの注意事項 4

Cisco NX-OS ソフトウェア アップグレードの前提条件 5

Cisco NX-OS ソフトウェアのアップグレード 6

Cisco NX-OS ソフトウェア ダウングレードの注意事項 10

Cisco NX-OS ソフトウェア ダウングレードの前提条件 11

以前のソフトウェア リリースへのダウングレード 11

NX-OS アップグレード履歴 14

第 3 章

vPC トポロジでのスイッチの移行 15

vPC フォークリフト アップグレード 15

第 4 章

Cisco NX-OS ソフトウェアのオプション 17

Cisco NX-OS ソフトウェアのオプション 17

Cisco NX-OS パッチ適用可能パッケージ/RPM インストールのガイドライン 19

モジュラ パッケージの使用 19

ベース モードまたはフル モードでの NX-OS イメージの起動 20

RPM に関する情報 20

RPM の形式 21

オプション RPM とその関連機能 22

NX-OS 機能 RPM インストールに関するガイドライン 24

サードパーティ製RPMインストールのガイドライン 26

機能またはサードパーティ RPM の操作に install CLI を使用する 26

デジタル署名のサポートにインストール CLI を使用する 29

インストールされているすべての RPM のクエリ 29

1 ステップ手順による RPM のインストール 31

2 ステップ手順による RPM のインストール 32

1 ステップの RPM のアップグレード 33

RPM のダウングレード 34

RPM のインストール 35

RPM の削除 35

について DNF コマンドの対応関係について コマンド 36

以下を使用したパッケージの操作：DNF コマンド 36

イメージのベース バージョン RPM を特定する 37

インストール済み RPM のリストをチェックする 37

インストール済み RPM の詳細を取得する 38

RPM のインストール 38

RPM のアップグレード 41

RPM のダウングレード 43

RPM の削除 45

以下をサポート DNF グループ 45

リポジトリを特定する 49

インストール済みの DNF バージョンを特定する方法を示します。 バージョン	49
NX-OS CLI と DNF コマンドの対応関係 コマンド	50
FTP サーバの設定とローカル FTP YUM リポジトリのセットアップ	51
Red Hat Enterprise Linux 7 (RHEL7) 仮想マシン上に FTP サーバを作成する	52
ローカル FTP YUM リポジトリを作成する	53
FTP サーバに到達するようにスイッチを設定する	55
インストール操作用ユーザ ロールの作成	56
Compacting Cisco NX-OS ソフトウェア イメージの圧縮	57



はじめに

この前書きは、次の項で構成されています。

- [対象読者](#) (vii ページ)
- [表記法](#) (vii ページ)
- [Cisco Nexus 3600 プラットフォーム スイッチの関連資料](#) (viii ページ)
- [マニュアルに関するフィードバック](#) (ix ページ)
- [通信、サービス、およびその他の情報](#) (ix ページ)

対象読者

このマニュアルは、Cisco Nexus スイッチの設置、設定、および維持に携わるネットワーク管理者を対象としています。

表記法

コマンドの説明には、次のような表記法が使用されます。

表記法	説明
bold	太字の文字は、表示どおりにユーザが入力するコマンドおよびキーワードです。
<i>italic</i>	イタリック体の文字は、ユーザが値を入力する引数です。
[x]	省略可能な要素（キーワードまたは引数）は、角かっこで囲んで示しています。
[x y]	いずれか1つを選択できる省略可能なキーワードや引数は、角カッコで囲み、縦棒で区切って示しています。
{x y}	必ずいずれか1つを選択しなければならない必須キーワードや引数は、波かっこで囲み、縦棒で区切って示しています。

表記法	説明
[x {y z}]	角かっこまたは波かっこが入れ子になっている箇所は、任意または必須の要素内の任意または必須の選択肢であることを表します。角かっこ内の波かっこと縦棒は、省略可能な要素内で選択すべき必須の要素を示しています。
<i>variable</i>	ユーザが値を入力する変数であることを表します。イタリック体が使用できない場合に使用されます。
string	引用符を付けない一組の文字。 string の前後には引用符を使用しません。引用符を使用すると、その引用符も含めて string とみなされます。

例では、次の表記法を使用しています。

表記法	説明
screen フォント	スイッチが表示する端末セッションおよび情報は、スクリーンフォントで示しています。
太字の screen フォント	ユーザが入力しなければならない情報は、太字のスクリーンフォントで示しています。
イタリック体の screen フォント	ユーザが値を指定する引数は、イタリック体の screen フォントで示しています。
<>	パスワードのように出力されない文字は、山カッコ (<>) で囲んで示しています。
[]	システム プロンプトに対するデフォルトの応答は、角カッコで囲んで示しています。
!、#	コードの先頭に感嘆符 (!) またはポンド記号 (#) がある場合には、コメント行であることを示します。

Cisco Nexus 3600 プラットフォーム スイッチの関連資料

Cisco Nexus 3600 プラットフォーム スイッチ全体のマニュアルセットは、次の URL にあります。

<http://www.cisco.com/c/en/us/support/switches/nexus-3000-series-switches/tsd-products-support-series-home.html>

マニュアルに関するフィードバック

このマニュアルに関する技術的なフィードバック、または誤りや記載もれなどお気づきの点がございましたら、HTML ドキュメント内のフィードバック フォームよりご連絡ください。ご協力をよろしくお願いいたします。

通信、サービス、およびその他の情報

- シスコからタイムリーな関連情報を受け取るには、[Cisco Profile Manager](#) でサインアップしてください。
- 重要なテクノロジーによって求めるビジネス成果を得るには、[シスコサービス](#)にアクセスしてください。
- サービスリクエストを送信するには、[シスコサポート](#)。
- 安全で検証済みのエンタープライズクラスのアプリケーション、製品、ソリューション、およびサービスを探して参照するには、[Cisco DevNet](#)。
- 一般的なネットワーク、トレーニング、認定関連の出版物を入手するには、[Cisco Press](#)にアクセスしてください。
- 特定の製品または製品ファミリの保証情報を探すには、[Cisco Warranty Finder](#) にアクセスしてください。

Cisco Bug Search Tool

[Cisco バグ検索ツール](#) (BST) は、シスコ製品とソフトウェアの障害と脆弱性の包括的なリストを管理する Cisco バグ追跡システムへのゲートウェイとして機能する、Web ベースのツールです。BST は、製品とソフトウェアに関する詳細な障害情報を提供します。



第 1 章

新機能および変更された機能に関する情報

- [新機能および変更された機能に関する情報（1 ページ）](#)

新機能および変更された機能に関する情報

表 1: 新機能および変更された機能

特長	説明	変更が行われたリリース	参照先
N/A	このリリースの機能更新はありません。	10.5(1)F	N/A



第 2 章

Cisco Nexus 3600 シリーズ NX-OS ソフトウェアのアップグレードまたはダウングレード

この章では、Cisco NX-OS ソフトウェアのアップグレードまたはダウングレードの方法について説明します。ここで説明する内容は、次のとおりです。

- [ソフトウェア イメージ \(3 ページ\)](#)
- [Cisco NX-OS ソフトウェアのアップグレードの推奨事項 \(4 ページ\)](#)
- [Cisco NX-OS ソフトウェア アップグレードの注意事項 \(4 ページ\)](#)
- [Cisco NX-OS ソフトウェア アップグレードの前提条件 \(5 ページ\)](#)
- [Cisco NX-OS ソフトウェアのアップグレード \(6 ページ\)](#)
- [Cisco NX-OS ソフトウェア ダウングレードの注意事項 \(10 ページ\)](#)
- [Cisco NX-OS ソフトウェア ダウングレードの前提条件 \(11 ページ\)](#)
- [以前のソフトウェア リリースへのダウングレード \(11 ページ\)](#)
- [NX-OS アップグレード履歴 \(14 ページ\)](#)

ソフトウェア イメージ

各デバイスには、Cisco NX-OS ソフトウェアが同梱されます。Cisco NX-OS ソフトウェアは、1つのNXOS ソフトウェアイメージで構成されています。イメージのファイル名は、「nxos」から始まります。 **NXOS**。

Cisco NX-OS オペレーティングシステムをロードするために必要なのは、このイメージだけです。このイメージは、すべての Cisco Nexus 3600 シリーズ スイッチで実行されます。



- (注) • バイナリ ファイルのもう 1 つのタイプは、ソフトウェア メンテナンス アップグレード (SMU) パッケージファイルです。SMU には、特定の不具合に対する修正プログラムが含まれています。これらは、直近の不具合に対処するために作成されたものであり、新しい機能は含まれていません。SMU パッケージファイルは、Cisco.com からダウンロードできます。通常、解決された不具合の ID 番号がファイル名に含まれています。SMU の詳細については、ソフトウェア メンテナンス アップグレード (SMU) の実行の章を参照してください。次に記載されています: [Cisco Nexus 3600-T システム管理の構成ガイド](#)。



- (注) シスコでは、Electronic Programmable Logic Device (EPLD) イメージアップグレードも提供しており、ハードウェア機能の強化や既知のハードウェア問題の解決を行っています。EPLD イメージアップグレードは、Cisco NX-OS ソフトウェア アップグレードとは独立しています。

Cisco NX-OS ソフトウェアのアップグレードの推奨事項

Cisco では、アップグレードを実行する前に、Nexus Health and Configuration Check を実行することを推奨します。利点には、潜在的な問題の特定、影響を受けやすいフィールド通知とセキュリティの脆弱性、推奨される設定の欠落などがあります。手順の詳細については、[Nexus の健全性とコンフィギュレーション チェックの実行](#)を参照してください。

Cisco NX-OS ソフトウェア アップグレードの注意事項



- (注) 値は、[『Cisco Nexus 3600 Series NX-OS Release Notes』](#) 各リリースに固有のアップグレードのガイドラインが含まれています。アップグレードを開始する前に、リリース ノートを参照してください。

以前のリリースから Cisco NX-OS リリース 10.1 (1) へのアップグレードでは、次のアップグレードパスがサポートされています。

- リリース 9.2 (x) → リリース 10.1 (1)
- リリース 9.3 (x) → リリース 10.1 (1)

Cisco NX-OS リリース 9.2 (1) からアップグレードするには、ブート変数を設定し、実行構成をスタートアップ構成にコピーして、デバイスをリロードする必要があります。Cisco NX-OS リリース 9.2(2) 以降または Cisco NX-OS リリース 9.3(x) にアップグレードするには、**install all** コマンドを使用します。

ソフトウェア イメージのアップグレードを試行する前に、次の注意事項に従ってください。

- アップグレードは、ネットワークが安定しているときにスケジュールします。
 - ソフトウェアイメージの破損につながるため、インストール手順の実行中の電源中断は回避してください。
 - デュアル スーパーバイザ モジュールのあるデバイスでは、ソフトウェアのアップグレード中に切り替えが発生した場合に接続を維持するため、両方のスーパーバイザモジュールがコンソール ポートで接続されている必要があります。詳細については、[ハードウェア 設置ガイド](#)（使用しているシャーシのもの）を参照してください。
 - CoPP 機能をサポートする Cisco NX-OS リリースから、新しいプロトコルの追加クラスを含む CoPP 機能をサポートする Cisco NX-OS リリースにアップグレードする場合は、**setup** コマンドまたは **copp profile** コマンドを使用してセットアップ ユーティリティを実行し、新しい CoPP クラスを使用可能にする必要があります。これらのコマンドの詳細については、「コントロールプレーン ポリシング」の章を参照してください。次に記載されています：『[Cisco Nexus 3600 Series NX-OS Security Configuration Guide](#)』。
 - 旧リリースからスイッチ プロファイルをサポートする Cisco NX-OS リリースにアップグレードする場合、実行構成コマンドの一部をスイッチプロファイルに移動することができます。詳細については、次の資料を参照してください。『[Cisco Nexus 3600 Series NX-OS System Management Configuration Guide](#)』。
 - デフォルトでは、ソフトウェア アップグレード プロセスは中断されます。
 - Cisco NX-OS リリース 10.4 (2) F 以降、Nexus 3600-R プラットフォームの場合、BIOS を最新バージョンにアップグレードするには、最初に nxos イメージにアップグレードする必要があります。このリリース以降、install all nxos コマンドは nxos sw を最新バージョンにアップグレードするだけですが、BIOS イメージは 10.4 (2) F バージョンより前にリリースされた最後の BIOS にアップグレードされます。
- 10.4 (2) F 以降のバージョンでリリースされた BIOS にアップグレードするには、最初に nxos イメージをアップグレードしてから、bios-force オプションを使用して BIOS をアップグレードします。次の例を参考にしてください。
1. nxos bootflash:nxos64-msll.10.4.2.F.bin でインストールを実行します。
システムがリロードされ、10.4 (2) F イメージで起動します。
 2. all nxos bios-force でインストールを実行します。



(注) スイッチは 2 回リロードします。1 回目は nxos のアップグレードのため、もう 1 回目は BIOS のアップグレードのためです。

Cisco NX-OS ソフトウェア アップグレードの前提条件

Cisco NX-OS ソフトウェアのアップグレードには、次の前提条件があります。

- デバイスまたはネットワークにアクセス可能などのユーザも、スケジュールされた時間にはデバイスまたはネットワークを設定しないでください。アップグレード中はデバイスを設定できません。 **show configuration session summary** コマンドを使用して、アクティブなコンフィギュレーションセッションがないことを確認してください。
- デバイスで Cisco NX-OS ソフトウェア イメージをアップグレードまたはダウングレードする前に、アクティブなすべてのコンフィギュレーションセッションを保存、送信、または破棄します。

デュアルスーパーバイザを搭載したデバイスでは、アクティブなコンフィギュレーションセッションがある場合、Cisco NX-OS ソフトウェアのアップグレード中にアクティブスーパーバイザ モジュールをスタンバイ スーパーバイザ モジュールに切り替えられません。

- デバイスからリモート サーバへのルートを確認します。サブネット間のトラフィックをルーティングするルータがない場合は、デバイスとリモートサーバが同じサブネットワーク内に存在する必要があります。リモート サーバへの接続を確認するには、**ping** コマンドを使用します。

```
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
```

設定セッションの詳細については、次を参照してください：『Cisco Nexus 3000 Series NX-OS System Management Configuration Guide』。

Cisco NX-OS ソフトウェアのアップグレード

Cisco NX-OS 10 にアップグレードするには、次の手順を使用します。5(x) リリース <!--KR7M: Richard's comment on PRRQ-->アップグレードする前に、Cisco.com で入手可能な Cisco Nexus 9000 および 3000 ISSU サポート マトリックス に基づき、ソース元（現在のリリース）とデスティネーション（ターゲット リリース）のバージョンを確認することをお勧めします。
<https://www.cisco.com/c/dam/en/us/td/docs/dcn/tools/nexus-9k3k-issu-matrix/index.html>



- (注) Cisco NX-OS リリース 9.2 (1) からアップグレードするには、ブート変数を設定し、実行構成をスタートアップ構成にコピーして、デバイスをリロードする必要があります。

手順の概要

1. このアップグレード手順の例外については、ソフトウェアイメージファイルに関するリリース ノート を参照してください。詳細については、[『Cisco Nexus 3600 Series NX-OS Release Notes』](#)。
2. コンソール ポート接続のデバイスにログインします。
3. イメージ ファイルをコピーするために必要なスペースがあることを確認します。
4. スーパーバイザモジュールにさらに多くのスペースが必要な場合、不要なファイルを削除して使用できるスペースを広げてください。
5. アクティブおよびスタンバイ スーパーバイザモジュールに使用できるスペースがあることを確認します。
6. スーパーバイザモジュールにさらに多くのスペースが必要な場合、不要なファイルを削除して使用できるスペースを広げてください。
7. Cisco.com にログインし、使用しているデバイスのソフトウェア イメージ ファイルを次の URL から選択して、ファイル サーバにダウンロードします。<http://software.cisco.com/download/navigator.html>。
8. 転送プロトコルを使用して、ソフトウェア イメージをアクティブ スーパーバイザ モジュールにコピーします。FTP、TFTP、SCP、SFTP を使用できます。
9. ファイルの SHA256 チェックサムを表示して、オペレーティングシステムの整合性を検証し、ダウンロードしたイメージが安全にインストールおよび使用できるかを確認します。
10. 実際にアップグレードを実行する前に、ソフトウェアのアップグレードの影響を確認します。
11. 実行コンフィギュレーションをスタートアップコンフィギュレーションに保存します。
12. Cisco NX-OS ソフトウェアをアップグレードします。 **install all nxos bootflash:filename [no-reload | non-interruptive]** コマンドを使用します。
13. （任意） アップグレードプロセス全体を表示します。
14. （任意） ログインし、必要なソフトウェアバージョンがデバイスで実行されていることを確認します。
15. （任意） 必要に応じて、任意のライセンスをインストールし、必要な機能がデバイスで使用できるようにします。詳細については、[『Cisco NX-OS ライセンス ガイド』](#)。

手順の詳細

手順

- ステップ 1** このアップグレード手順の例外については、ソフトウェアイメージファイルに関するリリース ノート を参照してください。詳細については、[『Cisco Nexus 3600 Series NX-OS Release Notes』](#)。
- ステップ 2** コンソール ポート接続のデバイスにログインします。
- ステップ 3** イメージ ファイルをコピーするために必要なスペースがあることを確認します。

```
switch# dir bootflash:
```

（注）

新しいイメージファイルのロードが成功しない場合、使用するデバイスに、少なくとも 1 つ前のリリースの Cisco NX-OS ソフトウェアのイメージファイルをダウンロードすることを推奨します。

- ステップ 4** スーパーバイザ モジュールにさらに多くのスペースが必要な場合、不要なファイルを削除して使用できるスペースを広げてください。

```
switch# delete bootflash:nxos.9.3.6.bin
```

- ステップ 5** アクティブおよびスタンバイ スーパーバイザ モジュールに使用できるスペースがあることを確認します。

- ステップ 6** スーパーバイザ モジュールにさらに多くのスペースが必要な場合、不要なファイルを削除して使用できるスペースを広げてください。

- ステップ 7** Cisco.com にログインし、使用しているデバイスのソフトウェア イメージファイルを次の URL から選択して、ファイル サーバにダウンロードします。 <http://software.cisco.com/download/navigator.html>。

- ステップ 8** 転送プロトコルを使用して、ソフトウェア イメージをアクティブ スーパーバイザ モジュールにコピーします。FTP、TFTP、SCP、SFTP を使用できます。

```
switch# copy scp://user@scpserver.cisco.com//download/nxos.10.1.1.bin bootflash:nxos.10.1.1.bin
```

圧縮が必要なソフトウェア画像の場合は、送信元として SCP、HTTP、または HTTPS を使用し、宛先としてブートフラッシュまたは USB を使用する必要があります。次の例では SCP およびブートフラッシュを使用します。

```
switch# copy scp://user@scpserver.cisco.com//download/nxos.10.1.1.bin
bootflash:nxos.10.1.1.bin compact vrf management use-kstack
```

```
user1@10.65.42.196's password:
nxos.10.1.1.bin 100% 1887MB 6.6MB/s 04:47
Copy complete, now saving to disk (please wait)...
Copy complete.
```

値は、**compact** ファイルをスーパーバイザ モジュールにコピーする前に、キーワードで NX-OS のイメージを圧縮します。

(注)

ソフトウェア画像の圧縮は、SCP、HTTP、または HTTPS でのみサポートされています。その他のプロトコルで圧縮しようとする場合、システムは次のエラーを返します。

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

(注)

圧縮された画像は、LXC 起動モードではサポートされません。

- ステップ 9** ファイルの SHA256 チェックサムを表示して、オペレーティングシステムの整合性を検証し、ダウンロードしたイメージが安全にインストールおよび使用できるかを確認します。

```
switch# show file bootflash://sup-1/nxos.10.1.1.bin sha256sum
5214d563b7985ddad67d52658af573d6c64e5a9792b35c458f5296f954bc53be
```

- ステップ 10** 実際にアップグレードを実行する前に、ソフトウェアのアップグレードの影響を確認します。

```
switch# show install all impact nxos bootflash:nxos.10.1.1.bin
```

ステップ 11 実行コンフィギュレーションをスタートアップ コンフィギュレーションに保存します。

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

ステップ 12 Cisco NX-OS ソフトウェアをアップグレードします。 **install all nxos bootflash:filename [no-reload | non-interruptive]** コマンドを使用します。

```
switch# install all nxos bootflash:nxos.10.1.1.bin
```

次のオプションを使用できます。

- **no-reload** : デバイスをリロードする前にソフトウェア アップグレード プロセスを終了します。

(注)

次の **install all** が **no-reload with** オプションを使用する場合、保存した設定はデバイスをリロードするまで使用できません。この状態で設定を保存し、NX-OS の新しいバージョンでデバイスをリロードすると、スタートアップ コンフィギュレーションが不正になる可能性があります。。

- **no-save** : 設定を保存しません。開始する前に手動で設定を保存します。 **install all** コマンドを使用します。
- **non-interruptive** : プロンプトなしでソフトウェアをアップグレードします。このオプションは、エラーや健全性チェックをすべてスキップします。
- **skip-epld** : epld イメージではなく、nxos イメージのみをインストールします。
- **skip-bios-upgrade** : nxos イメージだけをインストールし、BIOS アップグレードをスキップします。
- **skip-kernel-upgrade** : nxos イメージのみをインストールし、カーネル アップグレードをスキップします。

(注)

- コマンドの **install all** コマンドを入力した場合は、コマンドにより互換性チェックが実行され、アップグレードされるモジュールが通知されます。さらに、インストールを続行するかどうかの確認が求められます。続行を選択すると、スイッチで現在実行されている NXOS ソフトウェア イメージがインストールされ、必要に応じて、実行中のイメージのさまざまなモジュールの BIOS がアップグレードされます。

ステップ 13 (任意) アップグレード プロセス全体を表示します。

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

ステップ 14 (任意) ログインし、必要なソフトウェアバージョンがデバイスで実行されていることを確認します。

```
switch# show version
```

ステップ 15 (任意) 必要に応じて、任意のライセンスをインストールし、必要な機能がデバイスで使用できるようにします。詳細については、[『Cisco NX-OS ライセンス ガイド』](#)。

Cisco NX-OS ソフトウェア ダウングレードの注意事項

以前のソフトウェアリリースへのダウングレードを試行する前に、次の注意事項に従ってください。

- 以前のリリースから Cisco NX-OS リリース 10.1 (1) へのダウングレードでは、次のダウングレードパスがサポートされています。
 - リリース 10.1 (1) > リリース 9.2 (x)
 - リリース 10.1 (1) > リリース 9.3 (x)

Cisco NX-OS リリース 9.2 (1) からダウングレードするには、ブート変数を設定し、実行構成をスタートアップ構成にコピーして、デバイスをリロードする必要があります。Cisco NX-OS リリース 9.2 (2) 以降または Cisco NX-OS リリース 9.3 (x) にアップグレードするには、次を推奨します。 **install all** コマンドを使用します。

- デュアル スーパーバイザ モジュールのあるデバイスでは、ソフトウェアのダウングレード中に切り替えが発生した場合に接続を維持するため、両方のスーパーバイザモジュールがコンソールポートで接続されている必要があります。詳細については、[ハードウェア 設置ガイド](#) (使用しているシャーシのもの) を参照してください。
- Cisco NX-OS は、デフォルトで自動的にゲストシェルのインストールおよび有効化を行います。ただし、ゲストシェルをサポートしない Cisco NX-OS イメージでデバイスがリロードされる場合、既存のゲストシェルが自動的に削除され、%VMAN-2-INVALID_PACKAGE メッセージが発行されます。ベストプラクティスとして、以前の Cisco NX-OS イメージへのダウングレードを実施する前に **guestshell destroy** コマンドでゲストシェルを削除してください。
- スイッチプロファイルをサポートする Cisco NX-OS リリースからスイッチプロファイルをサポートしない Cisco NX-OS リリースにダウングレードする場合、スイッチプロファイル (構成されている場合) を削除する必要があります。詳細については、次の資料を参照してください。 [『Cisco Nexus 3600 Series NX-OS System Management Configuration Guide』](#)。



(注) ソフトウェア ダウングレードには中断が伴います。In-service software downgrade (ISSD、無停止のダウングレード) はサポートされていません。

Cisco NX-OS ソフトウェア ダウングレードの前提条件

Cisco NX-OS ソフトウェアのダウングレードには、次の前提条件があります。

- コントロールプレーン ポリシング (CoPP) 機能をサポートしている Cisco NX-OS リリースから CoPP 機能をサポートしていない以前の Cisco NX-OS リリースへのダウングレードを実行する前に、**show incompatibility nxos bootflash:filename** コマンドを使用して互換性を確認しておく必要があります。非互換な部分が存在する場合は、ソフトウェアをダウングレードする前に、ダウングレードイメージと互換性がない機能をすべて無効化してください。

以前のソフトウェア リリースへのダウングレード

最新のCisco NX-OS リリース 10にダウングレードするには、次の手順を使用します。1(x)からCisco NX-OS リリース 9.3(x) またはCisco NX-OS リリース 9.2(2) 以降。



- (注) Cisco NX-OS リリース 9.2 (1) からダウングレードするには、ブート変数を設定し、実行構成をスタートアップ構成にコピーして、デバイスをリロードする必要があります。

手順の概要

1. このダウングレード手順の例外については、ソフトウェアイメージファイルに関するリリースノートを参照してください。詳細については、[Cisco Nexus 3600 シリーズ NX-OS リリース ノート](#)を参照してください。
2. コンソール ポート接続のデバイスにログインします。
3. ダウングレードするイメージファイルがアクティブ スーパーバイザ モジュール bootflash: に存在することを確認します。
4. ソフトウェア イメージファイルがない場合は、Cisco.com にログインし、使用しているデバイスのソフトウェア イメージ ファイルを次の URL から選択して、ファイル サーバーにダウンロードします。 <http://software.cisco.com/download/navigator.html>。
5. 転送プロトコルを使用して、ソフトウェア イメージをアクティブ スーパーバイザ モジュールにコピーします。FTP、TFTP、SCP、SFTP を使用できます。
6. ソフトウェアの非互換性に関してチェックします。
7. ダウングレード イメージと互換性のない機能をすべて無効化します。
8. ハードウェアの非互換性に関してチェックします。
9. サポートされていないモジュールの電源をすべてオフにします。
10. 実行コンフィギュレーションをスタートアップコンフィギュレーションに保存します。
11. Cisco NX-OS ソフトウェアをダウングレードします。
12. (任意) ダウングレード プロセス全体を表示します。

13. (任意) ログインし、必要なソフトウェアバージョンがデバイスで実行されていることを確認します。

手順の詳細

手順

- ステップ 1 このダウングレード手順の例外については、ソフトウェアイメージファイルに関するリリースノートを参照してください。詳細については、[Cisco Nexus 3600 シリーズ NX-OS リリース ノート](#)を参照してください。
- ステップ 2 コンソール ポート接続のデバイスにログインします。
- ステップ 3 ダウングレードするイメージファイルがアクティブ スーパーバイザ モジュール bootflash: に存在することを確認します。

```
switch# dir bootflash:
```
- ステップ 4 ソフトウェア イメージファイルがない場合は、Cisco.com にログインし、使用しているデバイスのソフトウェア イメージ ファイルを次の URL から選択して、ファイル サーバーにダウンロードします。
<http://software.cisco.com/download/navigator.html>
- ステップ 5 転送プロトコルを使用して、ソフトウェアイメージをアクティブ スーパーバイザモジュールにコピーします。FTP、TFTP、SCP、SFTP を使用できます。

```
switch# copy scp://user@scpserver.cisco.com//download/nxos.9.2.3.bin bootflash:nxos.9.2.3.bin
```
- ステップ 6 ソフトウェアの非互換性に関してチェックします。

```
switch# show incompatibility-all nxos bootflash:nxos.9.2.3.bin
```


Checking incompatible configuration(s)
No incompatible configurations

結果の出力に非互換性と解決策が表示されます。
- ステップ 7 ダウングレード イメージと互換性のない機能をすべて無効化します。
- ステップ 8 ハードウェアの非互換性に関してチェックします。

```
switch# show install all impact nxos bootflash:nxos.9.2.3.bin
```
- ステップ 9 サポートされていないモジュールの電源をすべてオフにします。

```
switch# poweroff module module-number
```
- ステップ 10 実行コンフィギュレーションをスタートアップ コンフィギュレーションに保存します。

```
switch# copy running-config startup-config
```
- ステップ 11 Cisco NX-OS ソフトウェアをダウングレードします。

```

switch# install all nxos bootflash:nxos.9.2.3.bin
switch# install all nxos nxos.9.2.3.bin.CCO
Installer will perform compatibility check first. Please wait.
Installer is forced disruptive

Verifying image bootflash:/nxos.9.2.3.bin.CCO for boot variable "nxos".
[#####] 100% -- SUCCESS

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

Preparing "nxos" version info using image bootflash:/nxos.9.2.3.bin.CCO.
[#####] 100% -- SUCCESS

Preparing "bios" version info using image bootflash:/nxos.9.2.3.bin.CCO.
[#####] 100% -- SUCCESS

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

Notifying services about system upgrade.
2019 Jun 06 09:59:20 Switch %$ 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 9.3(1) 9.2(3) yes
1 bios v01.11(06/06/2019):v01.11(06/06/2019) v01.10(03/15/2019) no

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

```

(注)

コマンドの **install all** コマンドを入力した場合は、コマンドにより互換性チェックが実行され、アップグレードされるモジュールが通知されます。さらに、インストールを続行するかどうかの確認が求められます。続行を選択すると、スイッチで現在実行されている NXOS ソフトウェア イメージがインストールされ、必要に応じて、実行中のイメージのさまざまなモジュールの BIOS がアップグレードされます。

ステップ 12 (任意) ダウングレード プロセス全体を表示します。

例：

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

ステップ 13 (任意) ログインし、必要なソフトウェアバージョンがデバイスで実行されていることを確認します。

```
switch# show version
```

NX-OS アップグレード履歴

Cisco Nexus 3600 スイッチの耐用期間中は、多くのアップグレード手順を実行できます。メンテナンス目的で実行する場合や、オペレーティングシステムを更新して新しい機能を取得する場合にアップグレードが発生します。時間の経過とともに、スイッチは何度も更新される可能性があります。アップグレードのタイプと発生時刻を表示すると、問題のトラブルシューティングやスイッチの履歴の把握に役立ちます。

Cisco NX-OS リリース 9.3(5) 以降 : Cisco Nexus 3600 スイッチは、時間の経過とともに実行されたすべてのアップグレードアクティビティをログに記録し、これらのイベントの包括的な履歴を提供します。保存されているアップグレード履歴タイプは次のとおりです。

- Cisco NX-OS システムのアップグレード
- 9Electronic Programmable Logic Device (EPLD) アップグレード
- Software Maintenance Upgrade (SMU) インストール

Cisco NX-OS のアップグレード履歴を表示します。 **show upgrade history** コマンドを使用します。出力には、以前にスイッチで発生したアップグレードアクティビティが表示され、各イベントの開始時刻と終了時刻が定義されます。以下は、次のコマンドの出力例です。 **show upgrade history** コマンドに適用されます。

```
switch# show upgrade history
```

TYPE	VERSION	DATE	STATUS
NXOS system image	9.3(6)	29 Jan 2021 05:41:11	Installation started
NXOS system image	9.3(6)	29 Jan 2021 05:55:13	Installation End
NXOS system image	10.1(1)	29 Jan 2021 05:56:06	Installation started
NXOS system image	10.1(1)	29 Jan 2021 14:59:05	Installation End



第 3 章

vPC トポロジでのスイッチの移行

この章では、vPC トポロジでスイッチのペアを別のペアに移行する方法について説明します。ここで説明する内容は、次のとおりです。

- [vPC フォークリフト アップグレード \(15 ページ\)](#)

vPC フォークリフト アップグレード

vPC トポロジでは、Cisco Nexus 3600 プラットフォーム スイッチのペアから別の Cisco Nexus 3600 プラットフォーム スイッチのペアに移行できます。詳細については、次の「vPC フォークリフト アップグレードシナリオ」の項を参照してください：『*Cisco Nexus 3600 Series NX-OS Interfaces Configuration Guide*』。



第 4 章

Cisco NX-OS ソフトウェアのオプション

この章では、Cisco NX-OS ソフトウェアのオプション性について説明します。

- [Cisco NX-OS ソフトウェアのオプション \(17 ページ\)](#)
- [Cisco NX-OS パッチ適用可能パッケージ/RPM インストールのガイドライン \(19 ページ\)](#)
- [モジュラ パッケージの使用 \(19 ページ\)](#)
- [ベース モードまたはフル モードでの NX-OS イメージの起動 \(20 ページ\)](#)
- [RPM に関する情報 \(20 ページ\)](#)
- [について DNF コマンドの対応関係について コマンド \(36 ページ\)](#)
- [FTP サーバの設定とローカル FTP YUM リポジトリのセットアップ \(51 ページ\)](#)
- [インストール操作用ユーザ ロールの作成 \(56 ページ\)](#)
- [Compacting Cisco NX-OS ソフトウェア イメージの圧縮 \(57 ページ\)](#)

Cisco NX-OS ソフトウェアのオプション

Cisco NX-OS リリース 9.2 (1) 以降では、Cisco NX-OS ソフトウェア イメージでモジュラ パッケージの管理をサポートします。これにより、Cisco NX-OS ソフトウェアは、基礎となる NX-OS ソフトウェアを変更することなく、機能を選択的に追加、削除、およびアップグレードする柔軟性を提供します。

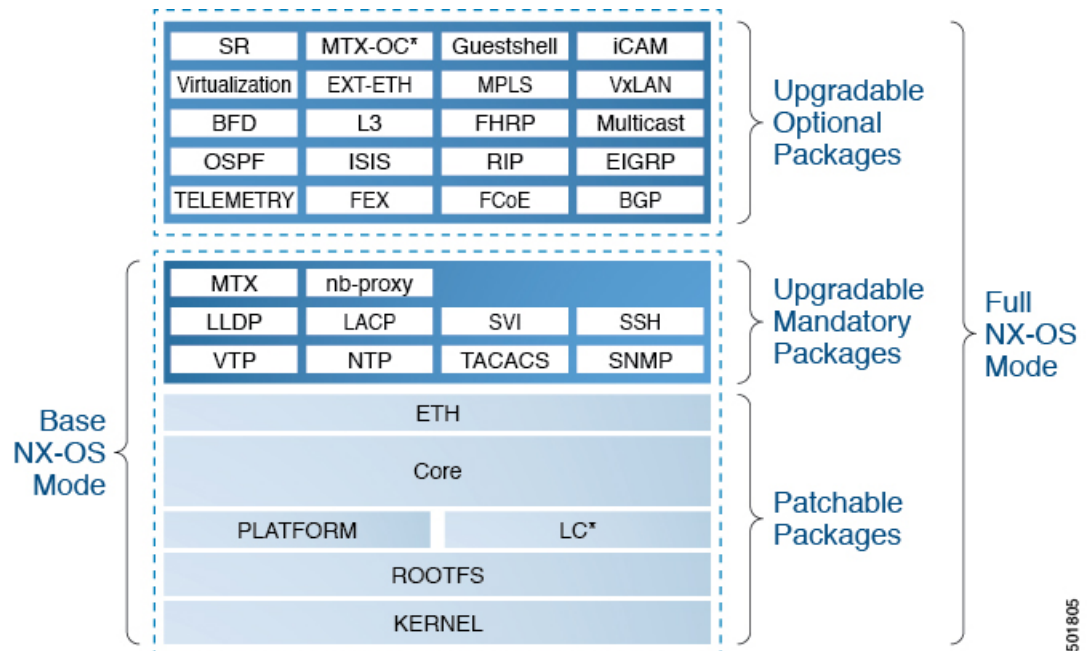
モジュール式の Cisco NX-OS ソフトウェアを使用する利点は次のとおりです。

- 無駄のない NX-OS ソフトウェア
- 機能と修正の非同期的な提供：クイックフィックス（新機能を含む）はリリースとは独立して提供されます。
- 実行時のバイナリとライブラリのフットプリントの削減

Cisco NX-OS ソフトウェアは、次の図に示すように2つのモードでNX-OS ソフトウェアをブートするようにプロビジョニングされています。

- ベース NX-OS モード
- フル NX-OS モード

図 1: Cisco NX-OS ソフトウェアのオプション



- ベース NX-OS モードには次が含まれます。
 - アップグレード可能な必須パッケージ
 - パッチ適用可能パッケージ
- フル NX-OS モードには次が含まれます。
 - アップグレード可能なオプション パッケージ
 - アップグレード可能な必須パッケージ
 - パッチ適用可能パッケージ



(注) デフォルトのモードは、フル NX-OS モードです。

ベース NX-OS モードでは、レイヤ 2 およびレイヤ 3 の基本的な機能が提供されます。すべてのダイナミック ルーティング機能（BGP、OSPF、EIGRP、RIP、ISIS など）やその他のオプション機能 RPM はデフォルトでは使用できません。オプションの機能 RPM は、ベースイメージの上にインストールする必要があります。

フル NX-OS モードでは、ブート時にイーサネット プラグインがプラグイン マネージャによりアクティブ化されるときにすべての機能 RPM がインストールされます。以前のリリースと比較して、ユーザの動作に変更はありません。

Cisco NX-OS パッチ適用可能パッケージ/RPM インストールのガイドライン

Cisco NX-OS のパッチ適用可能パッケージ/RPM のインストール（リリース 7.x の機能）のガイドラインについては、次の文書の、「ソフトウェア メンテナンスアップグレードの実施」の項を参照してください。『[Cisco Nexus 9000 Series NX-OS System Management Configuration Guide](#)』。

モジュラ パッケージの使用

Cisco NX-OS ソフトウェア イメージは、以前から Cisco Linux ディストリビューションを形成するパッケージングで構成されています。各パッケージのサイズが大きいため、特定のパッケージのアップグレードが困難になっています。

このセクションでは、Cisco NX-OS ソフトウェア イメージの新しいパッケージの管理について説明します。Cisco NX-OS リリース 9.2(1)以降では、BGP、OSPF、VXLAN、MPLS、セグメントルーティングなどの一部の NX-OS 機能はオプションと見なされます。

各モジュラ パッケージには、次の重要な特徴があります。

- アップグレード機能：モジュラ パッケージは個別にアップグレード可能です。モジュラ パッケージは、同じリリースのものを使用する必要があります。複数のリリースにまたがるパッケージでのアップグレードの実行はサポートされていません。
- オプション性：モジュラ パッケージはオプションです。たとえば、これらのパッケージは実行時に削除またはアンインストールが可能です。モジュラ パッケージの削除はシステムの稼働に影響を与えず、スイッチのその他の機能にも影響を与えません。



(注) モジュラ パッケージでエクスポートされたすべての API は、機能のインストール後にのみ使用する必要があります。

RPM と DNF

RPM (Red Hat Package Manager) は、Linux Standard Base (LSB) 内のパッケージングに使用されるパッケージ管理システムです。RPM コマンド オプションは、次の 3 つのサブグループにまとめられます。

- パッケージのクエリと確認
- パッケージのインストール、アップグレードおよび削除
- その他の機能の実行

rpm は RPM で使用されるメイン コマンドのコマンド名です。一方、**.rpm** は RPM ファイルに使用される拡張子です。

Dandified YUM YUM : Yellowdog Updater Modified または DNF は、RPM ベース Linux システム用のオープン ソース コマンドライン ツールです。これにより、ユーザとシステム管理者はシステム上のソフトウェア パッケージのインストール、アップデート、削除、または検索を簡単に行うことができます。DNF により、自動アップデートとパッケージ管理（依存関係管理を含む）の機能がシステムに追加されます。システムにインストールされているパッケージを理解することに加えて、DNF は、パッケージのコレクションであるリポジトリと連携します。通常、リポジトリにはネットワーク接続を介してアクセスできます。

ベース モードまたはフル モードでの NX-OS イメージの起動

NX-OS イメージは、ベース モードまたはフル モードでブートできるようになりました。フル ブート モードでは、以前のリリースのソフトウェアと同様な完全な NX-OS ソフトウェアがインストールされます。これは、デフォルトのブート モードです。ベース ブート モードでは、オプションの RPM はインストールされません。

コマンドライン オプションを使用するには、次の手順を参照してください。

- **install reset nxos base** オプションを VSH プロンプトで使用して、NX-OS イメージを基本ブート モードでインストールします。リロード後にスイッチはベース モードになり、オプション パッケージはインストールされません。
- **install reset nxos full** オプションを VSH プロンプトで使用して、NX-OS イメージをフルブート モードでインストールします。リロード後にスイッチはフル モードになり、オプションのパッケージが自動的にインストールされます。

詳細については、「機能 RPM の操作に install CLI を使用する」セクションを参照してください。

RPM に関する情報

RPM は、NXOS の **install** コマンドを使用して、または次のコマンドを使用して新しいソフトウェア バージョンにアップグレードまたはダウングレードできます：DNF コマンドにも表示されません。アップグレード可能な RPM には、オプションと必須があります。



- (注) NX-OS の起動プロセス中、イメージ抽出段階が行われている間、署名された RPM はメモリに残ります。ただし、この方法はメモリ消費の点で最も効率的ではありません。Cisco NX-OS リリース 10.4(3)F では、システムが安定した状態になり、十分な SSD スペースにアクセスできるようになると、RPM はメモリから永続ストレージに転送されます。この機能は、N9K-C92348GC-X およびすべての Nexus 9300 TOR スイッチでサポートされています。

オプションおよび必須の RPM の詳細については、以降のセクションを参照してください。

RPM の形式

RPM の一般的な形式は <name>-<version>-<release><arch>.rpm です。同じ形式が NX-OS 機能 RPM にも適用されます。

- name : パッケージ名 (例 : BGP)
- <x.y.x.b> 形式のバージョン: <major.minor.patch.build_number>。例: 2.0.1.0
- release : RPM 作成元のブランチ (例 : 9.2.1)
- arch : RPM のアーキテクチャ タイプ (例 : lib32_n9000)

次の表では、fex-2.0.0.0-9.2.1.lib32_n9000.rpm を例として命名規則の詳細を説明しています。

表 2: RPM 命名規則

RPM 命名規則 例 : fex-2.0.0.0-9.2.1.lib32_n9000.rpm	説明
fex	コンポーネントの名前を示しています。
2	RPM に後方互換性がないことを示します。アップグレード中に設定の損失が発生します。
0	後方互換性がある増分 API 変更/CLI 変更/スキーマ変更を示します。既存の機能上の新しい機能が該当します。アップグレード中に失われる設定はありません。
0	機能の変更がないバグ修正を示します。アップグレード中に失われる設定はありません。
0	この番号は、リリースの開発サイクルの間にコンポーネントが変更された回数を追跡します。この値はすべてのリリース イメージで 0 となります。

RPM 命名規則	説明
例 : fex-2.0.0.0-9.2.1.lib32_n9000.rpm	
9.2.1	RPMのリリース番号またはディストリビューションバージョンを示します。NVR 形式に沿っています。機能 RPM は NX-OS リリースにのみ適用可能であるため、このフィールドには存在する NX-OS リリース バージョンのみが指定されます。
lib32_n9000	RPM のアーキテクチャ タイプを示します。

オプション RPM とその関連機能

オプション RPM をインストールすると、ネイティブの NX-OS 動作に影響を与えずに機能を有効化できます。また、オプション RPM は、**install deactivate** コマンドを使用してスイッチから削除できます。

EIGRP などのオプション RPM は、基本ソフトウェアの一部ではありません。これらの RPM は、必要に応じてスイッチに対して追加、アップグレード、削除が可能です。**dnf** または **install** スイッチからの CLI コマンドを使用します。

次のリストは、オプション RPM とその関連機能を示しています。

表 3: オプション RPM とその関連機能

パッケージ名	関連機能
アプリケーションのホスティング	feature app-hosting
BGP	feature bgp
BFD	feature bfd
Container-tracker	feature container-tracker
EIGRP	feature eigrp

パッケージ名	関連機能
Ext-Eth	• feature openflow • feature evb • feature imp • feature netflow • feature sla_sender • feature sla_responder • feature sla twamp-server • feature sflow
EXT_ETH_LOWMEM	• feature evb • feature netflow
FCoE	• feature-set fcoe • feature-set fcoe-npv
FEX	feature-set fex
FHRP	• feature hsrp • feature vrrpv3
HW TELEMETRY	feature hw telemetry
iCAM	feature icam
ISIS	feature isis
MPLS	• feature mpls segment-routing • feature mpls evpn
マルチキャスト	• feature pim • feature pim6 • feature msdp • feature ngmvpn
NIA	N/A
NXSDK	N/A

パッケージ名	関連機能
OSPF	• feature ospf • feature ospfv3
RIP	feature rip
SDAA	N/A
サービス	feature catena
SR	feature mpls segment-routing traffic-engineering
TELEMETRY	feature telemetry
仮想化	該当なし
VM トラッカー	機能 vmtracker
VXLAN	• feature nv overlay • feature fabric forwarding

NX-OS 機能 RPM インストールに関するガイドライン

Cisco NX-OS シリーズ スイッチに存在する次の RPM 管理用 NX-OS システム RPM リポジトリを参照してください。



(注) RPM をシステムのリポジトリに手動でコピーをすることは避けてください。代わりに、install または次のコマンドを使用してください： DNF コマンドにも表示されません。

表 4: スイッチに存在する **RPM** リポジトリ

リポジトリ名	リポジトリのパス	説明
groups-repo	/rpms	バンドルされている NX-OS イメージの一部です。NX-OS イメージの一部としてバンドルされているすべての RPM を保持するために使用されます。このリポジトリに格納されているすべての RPM は、基本 RPM と呼ばれます。

リポジトリ名	リポジトリのパス	説明
localdb	/bootflash/.rpmstore/patching/localrepo	RPM の保持に使用されます。ユーザが NX-OS 機能 RPM を install add コマンドの一部として追加すると、RPM がこの場所にコピーされ、リロード時にも持続します。ユーザは、リポジトリをクリーンアップする責任があります。 このリポジトリに RPM を追加するには、install add コマンドを使用します。 このリポジトリから RPM を削除するには、install remove コマンドを使用します。 DNF コマンドでも、リポジトリに追加できます。 Cisco Nexus 3000 シリーズ スイッチを除き、リポジトリの最大領域は 200 Mb です (Cisco Nexus 9000 シリーズ スイッチのパッチ リポジトリを含む)。Cisco Nexus 3000 シリーズ スイッチでは、リポジトリの最大サイズは 20 Mb です。
patching	/bootflash/.rpmstore/patching/patchrepo	RPM の保持に使用されます。ユーザが NX-OS パッチ RPM をスイッチに追加すると、パッチ RPM がこのリポジトリにコピーされます。
Third_Party	/bootflash/.rpmstore/thirdparty	ユーザがサードパーティ RPM を追加したときに、RPM の保持に使用されます。

値は、**groups-repo** および **localdb** リポジトリには、システムブート時またはアクティベーション時にインストールする必要がある NX-OS 機能 RPM が保持されます。DNF コマンドまたは **install** コマンドは、これらの RPM のインストールまたは削除に使用できます。

次のルールが、ブートまたはインストール時の機能 RPM のインストール手順に適用されます。

- 同じ NX-OS リリース番号の RPM のみをインストール用に選択する必要があります。

- ベース RPM は、localdb リポジトリに追加できません。 **localdb** リポジトリにイメージを追加します。

サードパーティ製RPMインストールのガイドライン

現在、シスコによって提供されていないサードパーティ製パッケージは、シスコによって署名されていない場合でも、デバイスにインストールできます。

リリース10.1 (x) 以降、シスコによって署名されていないサードパーティパッケージは、デバイスにインストールできません。ただし、これをバイパスしてソフトウェアをインストールする場合は、サードパーティ製ソフトウェアのインストールを有効にするようにデバイスを設定できます。設定は通常の設定として保持され、確認できます。 **running-config** コマンドを使用します。この設定に従って、既知のリスクがあるサードパーティ製ソフトウェアをインストールできます。

機能またはサードパーティ RPM の操作に **install CLI** を使用する

機能 RPM の操作に **install CLI** を使用するには、次のリファレンス表を参照してください。

表 5: 機能 **RPM** 操作の **install CLI** リファレンス

CLI	説明
install reset	この操作は、すべてのパッチ、保持されたコンフィグレーション、アップグレードされたパッケージ、-サードパーティのインストール済みパッケージ、未保存のコンフィグレーションを削除し、デフォルトのパッケージを使用してスイッチの以前のモード（フル/基本）をリロードします。 値は、 install reset コマンドも write erase 操作を実行します。次のメッセージがプロンプトに表示されます。 <pre>switch(config)# install reset</pre> <pre>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.</pre> <pre>Do you want to proceed with reset operation? (y/n)? [n]</pre>

CLI	説明
install reset nxos base	この操作は、すべてのパッチ、アップグレードされたパッケージ、保持された etc コンフィグレーション、-サードパーティのインストール済みパッケージ、スタートアップ設定 (write erase) を削除して NX-OS をベース モードでインストールして、デフォルトのパッケージを使用してスイッチをリロードします。
install reset nxos full	この操作は、すべてのパッチ、アップグレードされたパッケージ、保持された etc コンフィグレーション、-サードパーティのインストール済みパッケージ、スタートアップ設定 (write erase) を削除して NXOS をフルモードでインストールして、デフォルトのパッケージ (必須およびオプションの RPM による) を使用してスイッチをリロードします。
install add <>	それぞれのリポジトリに RPM ファイルを追加して、次のリポジトリを更新します。 patch/feature/third-party 。
install activate <rpm name>	リポジトリに存在する RPM をインストールします。
install commit <rpm name>	パッチ RPM に使用します。リロード時にパッチを保持します。
install deactivate <rpm name>	RPM をアンインストールします。 Cisco NX-OS リリース 10.1 (1) 以降では、このコマンドを使用して RPM を非アクティブ化すると、RPM の基本バージョンにダウングレードするオプションまたは RPM をアンインストールするオプションが表示されます。必要なオプションを選択すると、操作が続行されます。
install remove <rpm name>	リポジトリから RPM ファイルを削除してリポジトリを更新します。
sh install active	システムにインストールされている RPM のリストを表示します (base rootfs RPM を含む)。(機能/パッチ/サードパーティ)。
sh install inactive	リポジトリに保持されていてインストールされていない RPM のリストを表示します。

CLI	説明
sh install packages	インストールされているすべての RPM をリストします（rootfs RPM を含む）。
[no] system software allow third-party	Cisco NX-OS リリース 10.1 (1) 以降では、デフォルトでは、サードパーティ製 RPM をデバイスにインストールできません。このコマンドは、この制限をバイパスし、サードパーティ製ソフトウェアのインストールを有効にするようにデバイスを設定します。 次のコマンドは、サードパーティコンフィギュレーションを適用せずにサードパーティ RPM をアクティブにした場合のエラーメッセージを示しています。 <pre>switch(config)# install activate pbwMonitor-1.0-1.5.0.x86_64.rpm Install operation 193 failed because package is not signed by Cisco.Enable TPS installation using 'system software allow third-party' CLI at Tue Nov 17 04:23:10 2020</pre> 次のコマンドは、設定適用後のサードパーティ製 RPM インストールのアクティブ化を示しています。 <pre>switch(config)# system software allow third-party switch(config)# 2020 Nov 17 04:25:41 switch %\$ VDC-1 %\$ %USER-2-SYSTEM_MSG: <<%PATCH-INSTALLER-2-TPS_FEATURE_ENABLED>> User has enabled TPS installation - patch_installer switch(config)# install activate pbwMonitor-1.0-1.5.0.x86_64.rpm [#####] 100% Install operation 194 completed successfully at Tue Nov 17 04:25:58 2020</pre> 次のコマンドは、サードパーティコンフィギュレーションの無効化を示しています。 <pre>switch(config)# no system software allow third-party switch(config)# 2020 Nov 17 04:27:17 switch %\$ VDC-1 %\$ %USER-2-SYSTEM_MSG: <<%PATCH-INSTALLER-2-TPS_FEATURE_DISABLED>> User has disabled TPS installation - patch_installer</pre>



- (注) ISSU を使用している場合、または以前のバージョンから Cisco NX-OS リリース 10.1.1 リリースにアップグレードしている場合は、アップグレード後最初の 30 分以内にサードパーティの設定を手動で適用して、サードパーティの RPM をインストールする必要があります。

デジタル署名のサポートにインストール CLI を使用する

次の CLI コマンドを使用して、デジタル署名サポート用の install CLI を実行します。

手順の概要

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

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	switch#install add bootflash:<keyfile> gpg-key 例 : install add bootflash:RPM-GPG-KEY-puppetlabs gpg-key [#####] 100% Install operation 304 completed successfully at Thu Jun 19 16:40:28 2018	Cisco GPG (NU Privacy Guard) キーを使用して Cisco リリース RPM に署名します。公開 GPG キーは、次に存在します : /etc/pki/rpm-gpg/arm-Nexus9k-rel.gpg。異なるソースからその他の公開キーを追加するには、このセクションの手順を使用してください。
ステップ 2	switch#install verify package <package-name>	パッケージを検証します。
ステップ 3	または switch#install verify bootflash:<RPM file> 例 : switch# install verify bootflash:vxlan-2.0.0.0-9.2.1.lib32_n9000.rpm RSA signed switch#	RPM ファイルが署名済みか未署名であるかどうかを確認するには、ステップ 2 または 3 を使用します。

インストールされているすべての RPM のクエリ

インストール済みのすべての RPM をクエリするには、次のステップを実行します。

手順の概要

1. show install packages

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	show install packages 例 : <pre>switch# show install packages Boot Image: NXOS Image: bootflash:/nxos.9.2.1.bin ----- Installed Packages attr.x86_64 2.4.47-r0.0 installed Unsigned aufs-util.x86_64 3.14+git0+b59a2167a1-r0.0 installed Unsigned base-files.n9000 3.0.14-r89.0 installed Unsigned base-passwd.lib32_x86 3.5.29-r0.1.0 installed Unsigned bash.lib32_x86 4.3.30-r0.0 installed Unsigned bfd.lib32_n9000 2.0.0.0-9.2.1 installed Signed bgp.lib32_n9000 2.0.0.0-9.2.1 installed Signed binutils.x86_64 2.25.1-r0.0 installed Unsigned bridge-utils.x86_64 1.5-r0.0 installed Unsigned busybox.x86_64 1.23.2-r0.0 installed Unsigned busybox-udhcp.x86_64 1.23.2-r0.0 installed Unsigned bzip2.x86_64 1.0.6-r5.0 installed Unsigned ca-certificates.all 20150426-r0.0 installed Unsigned cgrouplite.x86_64 1.1-r0.0 installed Unsigned chkconfig.x86_64 1.3.58-r7.0 installed Unsigned container-tracker.lib32_n9000 2.0.0.0-9.2.1 installed Signed containerd-docker.x86_64 0.2.3+gitaa8187dbd3b7ad67d8e5e3a15115d3eef43a7ed1-r0.0 installed Unsigned core.lib32_n9000 2.0.0.0-9.2.1 installed Signed coreutils.lib32_x86 8.24-r0.0 installed Unsigned cpio.x86_64 2.12-r0.0 installed Unsigned cracklib.lib32_x86 2.9.5-r0.0 installed Unsigned cracklib.x86_64 2.9.5-r0.0 installed Unsigned createrepo.x86_64 0.4.11-r9.0 installed Unsigned cronie.x86_64 1.5.0-r0.0 installed Unsigned curl.lib32_x86 7.60.0-r0.0 installed Unsigned db.x86_64 6.0.30-r0.0 installed Unsigned dbus-1.lib32_x86 1.8.20-r0.0 installed Unsigned dhcp-client.x86_64 4.3.2-r0.0 installed Unsigned dhcp-server.x86_64 4.3.2-r0.0 installed Unsigned switch#</pre>	インストールされているすべての RPM をクエリします。

1 ステップ手順による RPM のインストール

RPM をインストールする CIL とアップグレードする CLI は同じです。1 ステップ手順を使用して RPM をインストールするには、次の手順を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	install add <rpm> activate 例 : <pre>switch# install add bootflash:chef.rpm activate Adding the patch (/chef.rpm) [#####] 100% Install operation 868 completed successfully at Tue May 8 11:20:10 2018 Activating the patch (/chef.rpm) [#####] 100% Install operation 869 completed successfully at Tue May 8 11:20:20 2018</pre>	RPM をインストールしてアクティブ化します。

例

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

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

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

2 ステップ手順による RPM のインストール

RPM をインストールする CIL とアップグレードする CLI は同じです。2 ステップ手順を使用して RPM をインストールするには、以下の手順を参照してください。

手順の概要

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

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	install add <rpm> 例 : <pre>switch# install add bootflash:vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm [#####] 100% Install operation 892 completed successfully at Thu Jun 7 13:56:38 2018 switch(config)# sh install inactive grep vxlan vxlan-2.0.1.0-9.2.1.lib32_n9000</pre>	RPM をインストールします。
ステップ 2	install activate <rpm> 例 :	RPM をアクティブ化します。

例

```
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#
```

1 ステップの RPM のアップグレード

RPM をインストールする CIL とアップグレードする CLI は同じです。RPM をアップグレードするには、以下の手順を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	install add <rpm>activate upgrade 例 : <pre>switch(config)# install add bootflash:bgp-2.0.2.0-9.2.1.lib32_n9000.rpm activate upgrade</pre> <pre>Adding the patch (/bgp-2.0.2.0-9.2.1.lib32_n9000.rpm) [#####] 100% Install operation 870 completed successfully at Tue May 8 11:22:30 2018</pre> <pre>Activating the patch (/bgp-2.0.2.0-9.2.1.lib32_n9000.rpm) [#####] 100% Install operation 871 completed successfully at Tue May 8 11:22:40 2018</pre>	RPM をインストールします。

例

```
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
```

RPM のダウングレード

ダウン グレード手順では、特別な CLI 属性を必要とします。1 ステップ手順を使用して RPM をダウングレードするには、次の手順を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	install add <rpm>activate downgrade 例 : <pre>switch(config)# install add bootflash:bgp-2.0.1.0-9.2.1.lib32_n9000.rpm activate downgrade</pre> <pre>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</pre> <pre>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>	RPM をダウン グレードします。

例

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

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

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

```

nxos-ssh-2.0.0.0-9.2.1.lib32_n9000
snmp-2.0.0.0-9.2.1.lib32_n9000
svi-2.0.0.0-9.2.1.lib32_n9000
tacacs-2.0.0.0-9.2.1.lib32_n9000
vtp-2.0.0.0-9.2.1.lib32_n9000
switch(config)#

```

RPM のインストール

RPM をアップグレードするには、以下の手順を参照してください。

手順の概要

1. **install deactivate** <rpm>

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	install deactivate <rpm> 例 : <pre> switch(config)# install deactivate bgp Base RPM found. Do you want to downgrade to base version(y/n) [n] y Downgrading to the base version [#####] 100% Install operation 190 completed successfully at Tue Nov 17 04:10:40 2020 </pre> 例 : <pre> switch(config)# install deactivate bgp Base RPM found. Do you want to downgrade to base version(y/n) [n] n </pre> <pre> WARNING!! This operation will remove 'bgp-3.0.0.0-9.4.1.lib32_n9000' related configuration from running-configuration on successful completion. Update startup-configuration accordingly. </pre> <pre> [#####] 100% Install operation 9 completed successfully at Tue Nov 17 05:05:59 2020 </pre>	groups-repo (/rpms) に RPM がある場合は、RPM の基本バージョンにダウングレードします。または、RPM をスイッチから完全にアンインストールします。 基本バージョンにダウングレードするには、次のように入力します： y 。RPM を完全にアンインストールするには、コマンドプロンプトで n と入力します。

RPM の削除

RPM を削除するには、次の手順を参照してください。

手順の概要

1. `install remove <rpm>`

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	<code>install remove <rpm></code> 例 : <pre>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</pre>	リポジトリから、RPM を削除します。

について DNF コマンドの対応関係について コマンド

詳細については、次の項を参照してください。DNF コマンドの対応関係について コマンドにも表示されません。



(注) DNF コマンドの対応関係について コマンドは、`ctrl+c` をサポートしていません。`install` コマンドは、`ctrl+c` をサポートしています。次の場合 DNF コマンドの対応関係について コマンドを `ctrl+c` で中断した場合、「`/isan/bin/patching_utils.py --unlock`」を使用して手動でクリーンアップする必要があります。

以下を使用したパッケージの操作 : DNF コマンド

以下のコマンドを使用してパッケージを操作するには、以下のセクションを参照してください。DNF コマンドを使用した NAT プールの作成およびグローバル IP アドレスの範囲の定義方法を示します。



(注) DNF コマンドは、ボックスの BASH シェルからのみアクセスできます。NXOS VSH ターミナルからはアクセスできません。



(注) `sudo` ユーザとして、スーパー ユーザ権限にアクセスできることを確認してください。

イメージのベースバージョン RPM を特定する

`ls/rpms` コマンドを使用して、イメージのベースバージョン RPM を特定します。ベース RPM バージョンは、システム イメージにアーカイブされた、事前インストール済みの RPM です。

#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                  mtz-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                  mtz-infra-2.0.0.0-9.2.1.lib32_n9000.rpm
services-2.0.0.0-9.2.1.lib32_n9000.rpm
```

インストール済み RPM のリストをチェックする

`dnf list installed` コマンドを使用して機能 RPM とサードパーティ RPM をクエリして、特定の RPM を `grep` 検索します。機能 RPM については、次の例を参照してください。

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

bfd.lib32_n9000	2.0.0.0-9.2.1	@groups-repo
core.lib32_n9000	2.0.0.0-9.2.1	installed
eth.lib32_n9000	2.0.0.0-9.2.1	installed
guestshell.lib32_n9000	2.0.0.0-9.2.1	@groups-repo
lacp.lib32_n9000	2.0.0.0-9.2.1	installed
linecard2.lib32_n9000	2.0.0.0-9.2.1	installed
lldp.lib32_n9000	2.0.0.0-9.2.1	installed
mcast.lib32_n9000	2.0.0.0-9.2.1	@groups-repo
mtz-device.lib32_n9000	2.0.0.0-9.2.1	installed

インストール済み RPM の詳細を取得する

```

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
...

```

インストール済み RPM の詳細を取得する

値は、**dnf info <rpmname>** コマンドは、インストール済み RPM の詳細情報リストを出力します。

dnf info vxlan

```

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

```

```

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

```

```

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

```

RPM のインストール

RPM をインストールすると、RPM がダウンロードされ、それぞれのプログラムがスイッチにコピーされます。次の例は、RPM をリモートサーバ（ネットワークで到達可能）からインストールする場合を示しています。


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

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

```

                                | 1.1 kB    00:00 ...
localdb
                                | 951 B     00:00 ...
localdb/primary
                                | 886 B     00:00 ...
localdb
                                |             1/1
patching
                                | 951 B     00:00 ...
thirdparty
                                | 951 B     00:00 ...
```

```
Setting up Install Process
vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm
```

```

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

```
Dependencies Resolved
```

Package	Repository	Arch	Version	Size
Installing:				
vxlan		lib32_n9000	2.0.1.0-9.2.1	
	/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!
```

次の例は、RPM をローカル ブートフラッシュからインストールする場合を示しています。

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

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

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

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

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

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

```
Setting up Install Process
```

```
Examining /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm: vxlan-2.0.1.0-9.2.1.lib32_n9000
```

```
Marking /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm as an update to
```

```
vxlan-2.0.0.0-9.2.1.lib32_n9000
```

```
Resolving Dependencies
```

```
--> Running transaction check
```

```
--> Package vxlan.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
```

```
--> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
```

```
--> Finished Dependency Resolution
```

```
Dependencies Resolved
```

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

```
Transaction Summary
```

```
Upgrade 1 Package
```

```
Total size: 6.4 M
```

```
Is this ok [y/N]: y
```

```
Downloading Packages:
```

```
Running Transaction Check
```

```
Running Transaction Test
```

```
Transaction Test Succeeded
```

```
Running Transaction
```

```
Updating : vxlan-2.0.1.0-9.2.1.lib32_n9000
```

```
1/2
```

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

```
pre-install for vxlan complete
```

```
starting post-install package version mgmt for vxlan
```

```
post-install for vxlan complete
```

```
Cleanup : vxlan-2.0.0.0-9.2.1.lib32_n9000
```

```
2/2
```

```
Updated:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1
```

```
Complete!
```

次の例は、RPM がリポジトリ内で利用可能な場合の RPM のインストールを示しています。

```
dnf install eigrp
```

RPM のアップグレード

次の例は、RPM をリモート サーバ（ネットワークで到達可能）からアップグレードする場合を示しています。

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

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

```

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

```
Setting up Upgrade Process
vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm
```

```

                                | 1.6 MB      00:00
Examining /var/tmp/yum-root-RaANgb/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm:
vxlan-2.0.1.0-9.2.1.lib32_n9000
Marking /var/tmp/yum-root-RaANgb/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm as an update to
vxlan-2.0.0.0-9.2.1.lib32_n9000
Resolving Dependencies
--> Running transaction check
--> Package vxlan.lib32_n9000 0:2.0.0.0-9.2.1 will be updated
--> Package vxlan.lib32_n9000 0:2.0.1.0-9.2.1 will be an update
--> Finished Dependency Resolution
```

```
Dependencies Resolved
```

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

```
Transaction Summary
```

```
Upgrade      1 Package
```

```
Total size: 6.4 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Check
```

```

Running Transaction Test
Transaction Test Succeeded
Running Transaction
** Found 1 pre-existing rpmdb problem(s), 'yum check' output follows:
busybox-1.23.2-r0.0.x86_64 has missing requires of busybox-syslog
  Updating      : vxlan-2.0.1.0-9.2.1.lib32_n9000                      1/2

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

Updated:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1

Complete!

```

次の例は、RPM をローカル ブートフラッシュからアップグレードする場合を示しています。

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

```

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

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

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

```

Transaction Summary

```

```

Upgrade      1 Package

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

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

                               2/2

Updated:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1

Complete!

```

次の例は、RPM がリポジトリ内で利用可能な場合の RPM のアップグレードを示しています。

```
dnf upgrade eigrp
```

RPM のダウングレード

次の例は、RPM をリモート サーバ（ネットワークで到達可能）からダウングレードする場合を示しています。

```

sudo dnf
downgrade vxlan-2.0.0.0-9.2.1.lib32_n9000

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

                               | 1.1 kB      00:00 ...
localdb

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

                               | 1.3 kB      00:00 ...
localdb

                               2/2
patching

                               | 951 B        00:00 ...
thirdparty

```

```

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

```

Dependencies Resolved

Package	Version	Arch	Repository
Downgrading:			
vxlan	2.0.0.0-9.2.1	lib32_n9000	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!

次の例は、RPM をローカル ブートフラッシュからダウングレードする場合を示しています。

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

次の例は、RPM がいずれかのリポジトリ内で利用可能な場合の RPM のダウングレードを示しています。

```
dnf downgrade eigrp
```

RPM の削除

RPM を削除すると、RPM がアンインストールされ、機能のコンフィグレーション CLI がすべて削除されます。 **dnf erase <rpm>** コマンドを使用して、RPM を削除します。

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

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

```
Dependencies Resolved
```

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

```
6.4 M
```

```
Transaction Summary
```

```
Remove      1 Package
```

```
Installed size: 6.4 M
```

```
Is this ok [y/N]: y
```

```
Downloading Packages:
```

```
Running Transaction Check
```

```
Running Transaction Test
```

```
Transaction Test Succeeded
```

```
Running Transaction
```

```
Erasing      : vxlan-2.0.1.0-9.2.1.lib32_n9000
```

```
1/1
```

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

```
pre-remove for vxlan complete
```

```
Removed:
```

```
vxlan.lib32_n9000 0:2.0.1.0-9.2.1
```

```
Complete!
```

以下をサポート DNF グループ

次の DNF グループのサポートは、パッケージ管理の一部です。これにより、管理者によるパッケージの管理が簡素化され、高度な柔軟性が提供されます。

groupinstall コマンドを使用する

管理者は、パッケージ (RPM) のリストを論理グループにグループ化して、さまざまな操作を実行することができます。DNF では、次の **group** コマンドがサポートされています。

- **groupinstall**
- **groupinfo**
- **groupremove**
- **groupupdate**

DNF グループは、L2、L3、ルーティング、および管理として幅広い分類が可能です。

groupinstall コマンドを使用する

Linux では、複数のパッケージを特定のグループにまとめることができます。個別のパッケージを **dnf** でインストールするのではなく、特定のグループをインストールすることにより、そのグループに属するすべての関連パッケージをインストールできます。たとえば、使用可能なすべてのグループを一覧表示するには、次を使用します：**dnf grouplist** コマンドに対して表示されます。

```
bash-4.4# dnf grouplist
Last metadata expiration check: 0:00:00 ago on Fri 08 Mar 2024 12:26:33 PM UTC.
] --- B/s | 0 B --:-- ETA

Available Groups:
  management
  routing
  L2
  L3
bash-4.4#
```

groupmembers コマンドを使用する

用途 **dnf groupinfo** コマンドを使用して、パッケージグループの説明と内容を表示します。このコマンドは、グループの機能メンバのリストを出力します。

```
bash-4.4# dnf groupinfo 12

Last metadata expiration check: 0:00:00 ago on Fri 08 Mar 2024 12:27:44 PM UTC.
] --- B/s | 0 B --:-- ETA

Group: L2
Mandatory Packages:
  lacp
  lldp
  svi
  vtp
bash-4.4#
```

groupinstall コマンドを使用する

このコマンドは、メンバ RPM のインストールとアップグレードの両方に使用します。メンバがインストールされていない場合は、使用可能な最も高いバージョンがインストールされます。メンバがすでにインストールされていてより高いバージョンの RPM が使用可能である場合、このコマンドでそのメンバがアップグレードされます。


```

bash-4.4# dnf groupinstall 13
Last metadata expiration check: 0:00:00 ago on Fri 08 Mar 2024 12:38:05 PM UTC.
] --- B/s | 0 B    --:-- ETA
Not a redundant system. Nothing todo
Dependencies resolved.
=====

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

Is this ok [y/N]: y
Complete!
Install operation 10 completed successfully at Fri Mar  8 12:38:08 2024.

[#####] 100%

```

groupupdate コマンドを使用する

dnf groupupdate コマンドを使用して、既存のインストール済みグループパッケージをすべて更新します。

```

bash-4.4# dnf groupupdate 13

Last metadata expiration check: 0:00:00 ago on Wed 13 Mar 2024 12:30:11 PM UTC.
] --- B/s | 0 B    --:-- ETA
Dependencies resolved.
=====
Group                                     Packages
=====
Marking packages as installed by the group:
@L3                                     bfd
=====

Package                                Arch      Version
Repository                               Size
=====
Installing group packages:
bfd                                lib32_64_n9000    2.0.0.0-10.4.3
groups-repo                                562 k
=====

Transaction Summary
=====
Install 1 Package

Total size: 562 k
Installed size: 2.3 M
Is this ok [y/N]: y
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :
                                1/1
  Running scriptlet: bfd-2.0.0.0-10.4.3.lib32_64_n9000
                                1/1
starting pre-install package version mgmt for bfd

```

grouperase コマンドを使用する

```

pre-install for bfd complete
Installing      : bfd-2.0.0.0-10.4.3.lib32_64_n9000
                                                         1/1

Running scriptlet: bfd-2.0.0.0-10.4.3.lib32_64_n9000
                                                         1/1

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

Installed:
  bfd.lib32_64_n9000 2.0.0.0-10.4.3

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

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

```

grouperase コマンドを使用する

dnf grouperase コマンドを使用して、グループ、またはグループのすべての RPM メンバーを削除します。

```

bash-4.4# dnf grouperase l3
Dependencies resolved.

```

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

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

Transaction Summary

Remove 1 Package

Freed space: 2.3 M

Is this ok [y/N]: y

Running transaction check

Transaction check succeeded.

Running transaction test

Transaction test succeeded.

Running transaction

```

Preparing      :
                                                         1/1

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

Running scriptlet: bfd-2.0.0.0-10.4.3.lib32_64_n9000
                                                         1/1

starting post-remove package version mgmt for bfd

```

```
post-remove for bfd complete

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

Removed:
  bfd.lib32_64_n9000 2.0.0.0-10.4.3

Complete!
Install operation 11 completed successfully at Fri Mar 8 12:38:41 2024.

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

リポジットリを特定する

このコマンドは、スイッチに存在するリポジトリに加え、これらのリポジトリに含まれる **RPM** の数をリストします。

```

bash-4.3# dnf repolist all

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

localdb                | 1.1 kB      00:00 ...

patching               | 951 B       00:00 ...

thirdparty             | 951 B       00:00 ...

repo id               | 951 B       00:00 ...

repo name

status

groups-repo          Groups-RPM Database
                        enabled: 37

localdb              Local RPM Database
                        enabled: 6

patching             Patch-RPM Database
                        enabled: 0

thirdparty           Thirdparty RPM Database
                        enabled: 0

open-nxos            open-nxos
                        disabled

repolist: 43

```

インストール済みの **DNF** バージョンを特定する方法を示します。バージョン

次に、インストール済みの DNF バージョンを特定する方法を示します。 version:

dnf --version

```

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

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

```

NX-OS CLI と DNF コマンドの対応関係 コマンド

次の表は、NX-OS CLI とそれに対応する DNF コマンドの対応関係 コマンドを使用した NAT プールの作成およびグローバル IP アドレスの範囲の定義方法を示します。

表 6: パッチ適用コマンドリファレンス

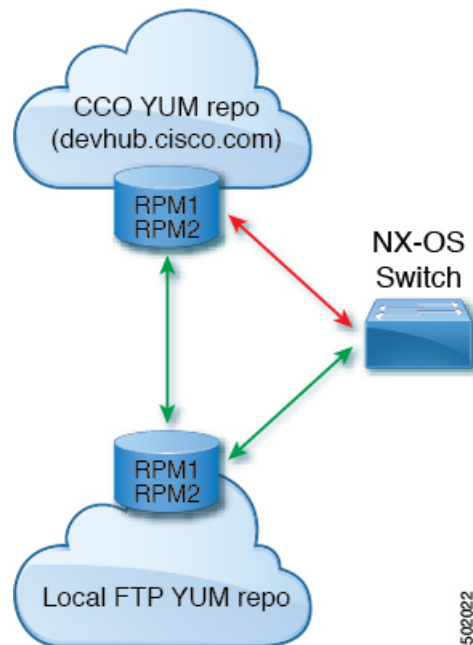
NX-OS CLI コマンド	DNF コマンドの対応関係 コマンド
show install inactive	dnf list --patch-only available
show install active	dnf list --patch-only installed
show install committed	dnf list --patch-only committed
show install packages	dnf list --patch-only
show install pkg-info	dnf info --patch-only
show install log	dnf history --show-patch-log ここで log_cmd は次のとおりです。 • opid : ある操作 ID に固有なログ • last : 最新の操作のログを示します。 • reverse : 逆の順序でログを表示します。 • detail : 詳細ログを表示します。 • from : 特定の操作 ID 以降のログを示します。
clear install log	dnf history --clear-patch-log= ここで clear_log_cmd は次のとおりです。 • all : ログを完全にクリアします。 • : この操作 ID 以降のログをクリアします。
install add	dnf install --add bootflash:/

NX-OS CLI コマンド	DNF コマンドの対応関係 コマンド
<code>install remove</code>	<code>dnf install --remove</code>
<code>install remove inactive</code>	<code>dnf install --remove all</code>
<code>install activate</code>	<code>dnf install --no-persist --nocommit</code> (注) デフォルトでは、すべてのパッケージがアクティブ化され、コミットされます。
<code>install deactivate</code>	<code>dnf erase --nocommit</code> (注) デフォルトでは、すべてのパッケージが非アクティブ化され、コミットされます。
<code>install commit</code>	<code>dnf install --commit</code>
<code>Install commit</code>	<code>dnf install --commit all</code>

FTP サーバの設定とローカル FTP YUM リポジトリのセットアップ

ローカル FTP YUM リポジトリをセットアップするには、次の図に示すように、初めに FTP サーバを作成して、ローカル FTP YUM リポジトリを作成し、FTP サーバに到達するように Cisco NX-OS スイッチを設定します。

図 2: FTP サーバの設定とローカル FTP YUM リポジトリのセットアップ



(注) Cisco NX-OS リリース 10.1(1) の場合は、<https://devhub.cisco.com/artifactory/open-nxos/10.1.1/> シスコのリポジトリにアクセスしてください。open-nxos リポジトリにイメージを追加します。

Red Hat Enterprise Linux 7 (RHEL7) 仮想マシン上に FTP サーバを作成する

Red Hat Enterprise Linux 7 (RHEL7) 仮想マシン上に FTP サーバを作成するには、次の手順を実行します。

手順の概要

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

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	dnf install vsftpd	Vsftpd (FTP サーバ) をインストールします。
ステップ 2	systemctl start vsftpd	FTP サーバを開始します。
ステップ 3	systemctl status vsftpd	サーバのステータスを確認します。
ステップ 4	firewall-cmd --zone=public --permanent --add-port=21/tcp	外部システムからの FTP サービスへのアクセスを許可し、ポート 21 を開きます。
ステップ 5	firewall-cmd --zone=public --permanent --add-service=ftp	FTP サービスを追加します。
ステップ 6	firewall-cmd --reload	サーバをリロードします。
ステップ 7	wget ftp:// <ip of FTP server> /test.txt	FTP サーバ内のファイル (test.txt など) をホストし、そのファイルの Wget を試みます。 (注) Note that /var/ftp/ は、FTP サーバのデフォルト ホームディレクトリです。

ローカル FTP YUM リポジトリを作成する

外部リポジトリ RPM と FTP サーバを同期し、ローカル FTP YUM リポジトリを作成するには、次の手順を実行します。

手順の概要

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

ローカル FTP YUM リポジトリを作成する

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	touch /etc/yum.repos.d/local.repo 例 : <pre>bash-4.3#touch /etc/yum.repos.d/local.repo</pre>	次の下にリポジトリ ファイルを作成します : /etc/yum.repos.d/ たとえば、 local.repo リポジトリを作成し、ベースURLを追加します。
ステップ 2	vim /etc/yum.repos.d/local.repo 例 : <pre>bash-4.3#vim /etc/yum.repos.d/local.repo</pre> <pre>[localrepo] name=localrepo baseurl= https://devhub.cisco.com/artifactory/open-nxos/7.0-3-I2-1/x86_64/ enabled=1 gpgcheck=0 sslverify=0</pre>	リポジトリ ファイルを編集し、localrepo の詳細をコピーします。 (注) ベース URL を必要なリポジトリ URL に変更します。
ステップ 3	cat /etc/yum.repos.d/local.repo 例 : <pre>bash-4.3#cat /etc/yum.repos.d/local.repo</pre> <pre>[localrepo] name=localrepo baseurl= https://devhub.cisco.com/artifactory/open-nxos/7.0-3-I2-1/x86_64/ enabled=1 gpgcheck=0 sslverify=0</pre>	ローカルリポジトリ データを確認して続行します。
ステップ 4	dnf repolist 例 : <pre>bash-4.3# dnf repolist Loaded plugins: fastestmirror, langpacks Loading mirror speeds from cached hostfile * base: mirror.dhakacom.com * extras: mirror.dhakacom.com * updates: mirror.dhakacom.com repo id repo name status base/7/x86_64 CentOS-7 - Base 9,911 extras/7/x86_64 CentOS-7 - Extras 313 localrepo localrepo 687 updates/7/x86_64 CentOS-7 - Updates 711 repolist: 11,622</pre>	リポジトリの到達可能性を確認します。
ステップ 5	nohup reposync -r <repo-name mentioned in the local.repo> -p <directory path to sync>& 例 :	すべてのパッケージを外部リポジトリからFTPサーバのホーム ディレクトリに同期します。

	コマンドまたはアクション	目的
	<pre>nohup reposync -r localrepo -p /var/ftp/ &</pre> このコマンドは、次の名前のディレクトリを作成します：local.repo 内部 /var/ftp/ 次からすべてのパッケージをダウンロードします：devhub.cisco.com としてディレクトリに割り当てます。	
ステップ 6	tail -f nouhup.out	同期のステータスを確認します。

FTP サーバに到達するようにスイッチを設定する

FTP サーバに到達するようにスイッチを設定するには、次の手順を実行します。

手順の概要

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

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	run bash sudo su	Sudo ユーザとしてログインします。
ステップ 2	ip netns exec management ping <ip_address>	コマンドを使用して、スイッチから FTP サーバーアドレスの到達可能性を確認します。 ping コマンドを使用します。
ステップ 3	touch /etc/yum/repos.d/ftp.repo 例： <pre>bash-4.3#touch /etc/yum/repos.d/ftp.repo</pre>	次の下にリポジトリ ファイルを作成します： /etc/yum/repos.d/ たとえば、 ftp.repo リポジトリにイメージを追加します。
ステップ 4	vim /etc/yum/repos.d/ftp.repo 例： <pre>bash-4.3#vim /etc/yum/repos.d/ftp.repo</pre> <pre>[ftp]</pre>	リポジトリファイルを編集し、ftp リポジトリの詳細をコピーします。 (注) ベース URL を必要な ftp サーバー IP に変更します。

	コマンドまたはアクション	目的
	<pre>name=ftp baseurl= ftp://198.51.100.1/localrepo/ enabled=1 gpgcheck=0 sslverify=0</pre>	
ステップ 5	cat /etc/yum/repos.d/ftp.repo 例 : <pre>bash-4.3# cat /etc/yum/repos.d/ftp.repo [ftp] name=ftp baseurl=ftp://198.51.100.1/localrepo/ enabled=1 gpgcheck=0 sslverify=0</pre>	URL として FTP サーバアドレスを使用して、スイッチ上にリポジトリ ファイルを作成します。
ステップ 6	ip netns exec management bash	Bash シェル プロンプトを使用します。
ステップ 7	dnf repolist 例 : <pre>bash-4.3# dnf repolist Loaded plugins: downloadonly, importpubkey, localrpmDB, patchaction, patching, : protect-packages groups-repo 1.1 kB 00:00 ... localdb 951 B 00:00 ... patching 951 B 00:00 ... thirdparty 951 B 00:00 ... thirdparty/primary 758 B 00:00 ... thirdparty 1/1 repo id repo name status groups-repo Groups-RPM Database 37 localdb Local RPM Database 0 patching Patch-RPM Database 0 thirdparty Thirdparty RPM Database 1 ftp ftp 686 repolist: 724</pre>	新しく作成されたリポジトリの到達可能性を確認します。
ステップ 8	dnf list available	新しいリポジトリで利用可能なパッケージをリストします。

インストール操作ユーザ ロールの作成

値は、**install** コマンドは、**admin** ロールのユーザーのみが使用できます。値は、**install** コマンドは、RBAC によりユーザーが利用できるようになります。ビジネスインサイトの ユーザ アカウントおよび **RBAC** の注意事項と制約事項 次の同じ項目を参照してください： *Cisco Nexus 3600 NX-OS Security Configuration Guide*。

Compacting Cisco NX-OS ソフトウェア イメージの圧縮



(注) このコマンドは、Cisco NX-OS の以下のリリースでは廃止されています：

Cisco NX-OS ソフトウェア画像の圧縮は、コピー要求を完了する前に画像ファイルのサイズを縮小します。送信元として SCP、HTTP、または HTTPS を使用し、宛先としてブートフラッシュまたは USB を使用します。次の例では SCP およびブートフラッシュを使用します。

```
switch# copy scp://user@scpserver.cisco.com//download/nxos64.10.1.1.bin  
bootflash:nxos64.10.1.1.bin compact vrf management use-kstack
```

```
user1@10.65.42.196's password:  
nxos64.10.1.1.bin 100% 1501MB 8.4MB/s 02:58  
Copy complete, now saving to disk (please wait)...  
Copy complete.
```

値は、**compact** キーワードは、ファイルをスーパーバイザモジュールにコピーする前に NX-OS イメージを圧縮します。



(注) ソフトウェア画像の圧縮は、SCP、HTTP、または HTTPS でのみサポートされています。その他のプロトコルで圧縮しようとする場合、システムは次のエラーを返します。

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



(注) 圧縮された画像は、LXC 起動モードではサポートされません。



(注) ソフトウェア画像の圧縮は、Cisco Nexus 9300 シリーズ プラットフォーム スイッチでのみサポートされます。

翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。