



## **Cisco Nexus 9000 シリーズ NX-OS ソフトウェア アップグレードおよびダウングレードガイド、リリース 10.3(x)**

初版：2022 年 8 月 19 日

最終更新：2023 年 1 月 31 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/](http://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](http://www.cisco.com/go/offices).

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.

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

© 2022 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 9000 シリーズ 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 viii

通信、サービス、およびその他の情報 ix

---

第 1 章

**新機能と変更情報 1**

新機能と変更情報 1

---

第 2 章

**Cisco Nexus 9000 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 イメージについて 3

ISSU について 5

Cisco NX-OS ソフトウェア アップグレードの前提条件 6

Cisco NX-OS ソフトウェア ダウングレードの前提条件 7

Cisco NX-OS ソフトウェア アップグレードの注意事項 8

ISSU プラットフォームのサポート 19

ISSU プラットフォームのサポート 25

Cisco NX-OS ソフトウェア ダウングレードの注意事項 30

アップグレードパス 32

アップグレードパッチ手順 32

拡張 ISSU の設定 42

Cisco NX-OS ソフトウェアのアップグレード 44

vPC のアップグレードプロセス 50

プライマリ スイッチでの vPC トポロジのアップグレードプロセス	50
セカンダリ スイッチでの vPC トポロジのアップグレードプロセス	51
サイトをマルチサイトに追加する。	52
Cisco NX-OS のアップグレード履歴	54

## 第 3 章

**Cisco NX-OS ソフトウェアのオプション 57**

Cisco NX-OS ソフトウェアのオプション	57
モジュラ パッケージの使用	59
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	60
RPM に関する情報	60
RPM の形式	60
オプション RPM とその関連機能	61
NX-OS 機能 RPM インストールに関するガイドライン	63
サードパーティ製 RPM インストールのガイドライン	65
機能またはサードパーティ RPM の操作に install CLI を使用する	65
デジタル署名のサポートにインストール CLI を使用する	68
インストールされているすべての RPM のクエリ	68
1 ステップ手順による RPM のインストール	69
2 ステップ手順による RPM のインストール	70
1 ステップの RPM のアップグレード	71
RPM のダウングレード	73
RPM のインストール	74
RPM の削除	74
YUM コマンドに関する情報	75
YUM コマンドを使用したパッケージの操作	75
イメージのベース バージョン RPM を特定する	75
インストール済み RPM のリストをチェックする	76
インストール済み RPM の詳細を取得する	77
RPM のインストール	77
RPM のアップグレード	80
RPM のダウングレード	82

RPM の削除	83
YUM グループのサポート	84
リポジトリを特定する	91
インストール済み YUM バージョンを特定する	92
NX-OS CLI と YUM コマンドの対応関係	92
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	94
Red Hat Enterprise Linux 7 (RHEL7) 仮想マシン上に FTP サーバを作成する	94
ローカル FTP YUM リポジトリを作成する	95
FTP サーバに到達するようにスイッチを設定する	96
インストール操作ユーザ ロールの作成	98
Compacting Cisco NX-OS ソフトウェア イメージの圧縮	98

---

 第 4 章

Cisco NX-OS から ACI ブート モードへの変換および ACI ブート モードから Cisco NX-OS への再変換	99
ACI ブート モードへの変換	99
リプレースメント スタンバイ スーパーバイザから ACI ブート モードへの変換	102
Cisco NX-OS への再変換	103
ACI シェルで SCP を使用したブートフラッシュへの NX-OS イメージのロード	106

---

 第 5 章

vPC トポロジでのスイッチの移行	109
vPC フォークリフト アップグレード	109
Nexus 9000 -RシリーズスイッチのvPCアップグレードおよびダウングレード手順	109





## 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 9000 シリーズ 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](#) (viii 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
<b>bold</b>	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、ユーザが値を指定する引数です。
[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、角かっこで囲み、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。
variable	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ないでください。引用符を使用すると、その引用符も含めて string と見なされます。

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 screen 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 9000 シリーズ スイッチの関連資料

Cisco Nexus 9000 シリーズ 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/en/US/products/ps13386/tsd\\_products\\_support\\_series\\_home.html](http://www.cisco.com/en/US/products/ps13386/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。
- 重要な技術によりビジネスに必要な影響を与えるには、[Cisco Services](#) 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、[Cisco Support](#) にアクセスしてください。
- 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 Marketplace](#) にアクセスしてください。
- 一般的なネットワーキング、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 バグ検索ツール

[Cisco Bug Search Tool](#) (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

## 新機能と変更情報

この章では、Cisco Nexus 9000 シリーズ NX-OS ソフトウェア アップグレードおよびダウングレードガイド、リリース 10.3(x)に記載されている新しい機能と変更された機能に関するリリース固有の情報について説明します。

- [新機能と変更情報 \(1 ページ\)](#)

## 新機能と変更情報

次の表は、Cisco Nexus 9000 シリーズ NX-OS リリース 10.3(x) ソフトウェア アップグレードおよびダウングレードガイドに記載されている新機能および変更機能をまとめたものです。それぞれの説明が記載されている箇所も併記されています。

表 1: NX-OS リリース 10.3(x)の新機能および変更された機能

機能	説明	変更が行われたリリース	参照先
ISSU のガイドライン	10.3(1)F リリースへのアップグレードのガイドライン	10.3(1)F	<a href="#">Cisco NX-OS ソフトウェア アップグレードの注意事項 (8 ページ)</a>





## 第 2 章

# Cisco Nexus 9000 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について \(3 ページ\)](#)
- [ISSU について \(5 ページ\)](#)
- [Cisco NX-OS ソフトウェア アップグレードの前提条件 \(6 ページ\)](#)
- [Cisco NX-OS ソフトウェア ダウングレードの前提条件 \(7 ページ\)](#)
- [Cisco NX-OS ソフトウェア アップグレードの注意事項 \(8 ページ\)](#)
- [ISSU プラットフォームのサポート \(25 ページ\)](#)
- [Cisco NX-OS ソフトウェア ダウングレードの注意事項 \(30 ページ\)](#)
- [アップグレードパス \(32 ページ\)](#)
- [アップグレードパッチ手順 \(32 ページ\)](#)
- [拡張 ISSU の設定 \(42 ページ\)](#)
- [Cisco NX-OS ソフトウェアのアップグレード \(44 ページ\)](#)
- [vPC のアップグレードプロセス \(50 ページ\)](#)
- [サイトをマルチサイトに追加する。 \(52 ページ\)](#)
- [Cisco NX-OS のアップグレード履歴 \(54 ページ\)](#)

## ソフトウェアイメージについて

各デバイスには、Cisco NX-OS ソフトウェアが事前インストールされています。Cisco NX-OS ソフトウェアは、1 つの NX-OS ソフトウェア イメージで構成されています。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。

Cisco NX-OS リリース 10.1(1)、10.1(2)、および 10.2(1)F には、32 ビット イメージと 64 ビット イメージがあります。

- 32 ビットの Cisco NX-OS イメージファイルのファイル名は、「nxos」で始まります（例：nxos.10.1.1.bin）。
- 64 ビットの Cisco NX-OS イメージファイルのファイル名は、「nxos64」で始まります（例：nxos64.10.1.1.bin）。

Cisco NX-OS リリース 10.2(2)F 以降、すべての Cisco Nexus プラットフォームは 64 ビットイメージで動作し、2 つの 64 ビットイメージがあります。

- 64 ビットの Cisco NX-OS イメージファイルには、「nxos64-cs」で始まるイメージファイル名（例：nxos64-cs.10.2.2.F.bin）があります。このイメージは、Cisco Nexus 9000 -EX、-FX、-GX、GX2 シリーズのモジュラースイッチおよび Nexus 9000 シリーズの固定スイッチでサポートされています。
- 64 ビットの Cisco NX-OS イメージファイルには、「nxos64-msll」で始まるイメージファイル名があります（例：nxos64-msll.10.2.2.F.bin）。このイメージは、Cisco Nexus 9000 -R および -R2 シリーズモジュラースイッチ、Cisco Nexus 3600 シリーズ固定スイッチ、および Cisco Nexus 3500-XL スイッチでサポートされています。

それぞれのプラットフォームでの 32 ビットまたは 64 ビットのイメージのサポートについては、次のドキュメントを参照してください。

- [Cisco Nexus 9000 シリーズ NX-OS リリース ノート、リリース 10.2\(1\)F](#)
- [Cisco Nexus 9000 シリーズ NX-OS リリース ノート、リリース 10.1\(2\)](#)
- [Cisco Nexus 9000 シリーズ NX-OS リリース ノート、リリース 10.1\(1\)](#)

Cisco Nexus 9000 シリーズスイッチは、中断を伴うソフトウェアのアップグレードとダウングレードをデフォルトでサポートしています。



- 
- (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（たとえば、n9000-dk10.1.1.）**CSCab00001.gbin** など）。SMU の詳細については、『Cisco Nexus 9000 シリーズ NX-OS システム管理設定ガイド』を参照してください。
- 



- 
- (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。EPLD イメージとアップグレードプロセスの詳細については、『Cisco Nexus 9000 シリーズ FPGA/EPLD アップグレードリリース ノート』を参照してください。
-

# ISSU について

in-service software upgrade (ISSU) は、スイッチがトラフィックの転送を続けながら、デバイスのソフトウェアをアップグレードすることができます。ISSU を使用すると、ソフトウェアのアップグレードによるダウンタイムを短縮するかゼロにすることができます。一部のスイッチでは、ISSU を実行できます。（サポートされているプラットフォームの完全なリストについては、[ISSU プラットフォームのサポート \(19 ページ\)](#) を参照してください。）

デフォルトのソフトウェアアップグレードプロセスは中断を伴います。したがって、ISSU はコマンドラインインターフェース (CLI) を使用して有効にする必要があります。詳細は本書の設定のセクションを参照してください。無停止オプションを使用すると、無停止アップグレードを確実に実行できます。ゲストシェルは ISSU の実行中は無効になり、アップグレード後に再びアクティブ化されます。

拡張 ISSU では、Cisco Nexus 9000 シリーズ スイッチの一部のサポートを追加しました。

次の ISSU シナリオはサポートされません。

- シングル スーパーバイザ構成のトップオブラック (ToR) スイッチでの標準 ISSU の実行
- シングル スーパーバイザ構成のトップオブラック (ToR) スイッチでの拡張 ISSU の実行

## シングル スーパーバイザ構成のトップオブラック (ToR) スイッチでの標準 ISSU の実行

ToR Cisco Nexus 9300 プラットフォーム スイッチおよび Cisco Nexus 3100 シリーズ スイッチは、シングル スーパーバイザ構成のスタンドアロン スイッチです。Cisco Nexus 9000 および 3100 シリーズ スイッチで ISSU を実行すると、スーパーバイザの CPU がリセットされ、新しいソフトウェア バージョンがロードされます。CPU が Cisco NX-OS ソフトウェアの更新バージョンをロードした後、システムはコントロールプレーンを以前の既知の設定とランタイム状態に復元し、データ プレーンと同期します。これで ISSU プロセスが完了します。

データ プレーン トラフィックは ISSU プロセスの実行中に中断されません。つまり、データ プレーンは、コントロールプレーンのアップグレード中にパケットを転送し、Cisco Nexus 9000 および 3100 シリーズ スイッチに接続されているすべてのサーバではトラフィックの中断が一切見られません。ISSU プロセス実行中のコントロールプレーンのダウンタイムは、約 120 秒未満です。

## シングル スーパーバイザ構成のトップオブラック (ToR) スイッチでの拡張 ISSU の実行



(注) Cisco NX-OS リリース 10.1(x) への拡張 ISSU はサポートされていません。これは、含まれているカーネル更新を有効化するために基盤となるカーネルのリロードが必要となるためです。システムに次のメッセージが表示されます。

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

実際に、システムは拡張 ISSU の代わりに、非中断 ISSU を実行します。

Cisco NX-OS ソフトウェアは、通常、ハードウェアで直接実行されます。ただし、シングルスーパーバイザ ToR 上の拡張 ISSU またはコンテナベース ISSU の構成は、スーパーバイザ モジュールおよびラインカードの仮想インスタンスを作成することによって実現されます。拡張 ISSU では、ソフトウェアはスーパーバイザおよびラインカード用の独立した Linux コンテナ (LXC) の内部で実行されます。3 番目のコンテナが ISSU 手順の一部として作成され、これがスタンバイ スーパーバイザとして稼働します。

仮想インスタンス (または Linux コンテナ) は、エミュレーションされたイーサネット接続を使用して相互に通信します。通常の状態では、vSup1 (アクティブなロール内の仮想 SUP コンテナ) と vLC (仮想ラインカードコンテナ) の 2 つの Linux コンテナのみがインスタンス化されます。拡張 ISSU では、スイッチに 16G のメモリが必要です。

拡張 ISSU (LXC) モードでの起動を有効にするには、**[no] boot mode lxc** コマンドを使用します。このコマンドは、コンフィギュレーションモードで実行されます。詳細については、次の設定例を参照してください。

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



(注) 拡張 ISSU を初めて有効にするときには、最初にスイッチをリロードする必要があります。

拡張 ISSU によるソフトウェア アップグレードの実行中は、スーパーバイザ コントロールプレーンは最小限のスイッチオーバーダウタイム中断を伴って稼働状態を維持し、ネットワークのフォワーディングステートはアップグレード中に正確に維持されます。スーパーバイザが最初にアップグレードされ、次にラインカードがアップグレードされます。

データプレーン トラフィックは ISSU プロセスの間に中断されません。コントロールプレーンのダウタイムは 6 秒未満です。



(注) 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。

ISSU および高可用性については、『[Cisco Nexus 9000 シリーズ NX-OS 高可用性および冗長性ガイド](#)』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。

- すべてのリリースの ISSU 互換性については、『[Cisco NX-OS ISSU サポート マトリックス](#)』を参照してください。



- 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** を使用します 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。
- ファイル転送プロトコル (TFTP、FTP、SFTP、SCP など) を介して NX-OS ソフトウェア イメージを Nexus スイッチに転送するには、NX-OS ソフトウェア イメージが保存するリモートファイルサーバに Nexus スイッチが接続できることを確認します。サブネット間のルータからルータへのトラフィックがない場合、ensure that the Nexus スイッチとリモートファイルサーバは同じサブネットワークにあることを確認します。リモートサーバへの接続を確認するには、選択したファイル転送プロトコルを使用してテストファイルを転送するか、リモートファイルサーバが ICMP エコー要求パケットに応答するように設定されている場合は ping コマンドを使用します。ping の使用例 リモートファイルサーバ 192.0.2.100 への接続を確認するコマンドを次に示します。

```
switch# ping 192.0.2.100 vrf management
PING 192.0.2.100 (192.0.2.100): 56 data bytes
64 bytes from 192.0.2.100: icmp_seq=0 ttl=239 time=106.647 ms
64 bytes from 192.0.2.100: icmp_seq=1 ttl=239 time=76.807 ms
64 bytes from 192.0.2.100: icmp_seq=2 ttl=239 time=76.593 ms
64 bytes from 192.0.2.100: icmp_seq=3 ttl=239 time=81.679 ms
64 bytes from 192.0.2.100: icmp_seq=4 ttl=239 time=76.5 ms

--- 192.0.2.100 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 76.5/83.645/106.647 ms
```

設定セッションの詳細については、『Cisco Nexus 9000 シリーズ NX-OS システム管理設定』を参照してください。

## 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。

## Cisco NX-OS ソフトウェア アップグレードの注意事項

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。

- Cisco Nexus リリース 10.1(2)、10.2(1)F、および 10.2(2)F、で実行されているデバイスでは、L2 サブインターフェイスが設定されている場合、ND-ISSU はサポートされません。
- Cisco NX-OS リリース 10.2(2)F 以降、Cisco Nexus 9504 および 9508 プラットフォーム スイッチ、および Cisco Nexus 9508-R、R2、および RX ラインカードは、Cisco NX-OS 64 ビット イメージをサポートします。以前のリリースから 10.2(2)F 64 ビット NX-OS イメージへの中断を伴うアップグレードがサポートされています。Cisco NX-OS 32 ビット イメージは、これらのプラットフォーム スイッチではサポートされなくなりました。
- Cisco NX-OS リリース 10.2(2)F 以降、FCoE/FC NPV は N9K-C9336C-FX2-E プラットフォーム スイッチでサポートされています。

FCoE (ファイバチャネルオーバーイーサネット) /FC (ファイバチャネル) NPV (Nポート仮想化) を使用した ISSU は、一部の Cisco Nexus 9000 スイッチでサポートされています。ISSU は、スイッチがトラフィックの転送を続けながら、デバイスのソフトウェアをアップグレードすることができます。一部の Cisco Nexus 9000 スイッチでは、In-Service Software Upgrade (ISSU、無停止アップグレード) を実行できます。デフォルトのソフトウェア アップグレード プロセスは中断を伴います。無停止オプションを使用すると、無停止アップグレードを確実に実行できます。

ファイバチャネル N ポート仮想化 (NPV) は、異なるファブリック アップリンク上の VXLAN と共存できますが、Cisco Nexus 93180YC-FX、N9K-C9336C-FX2-E、および N9k-C93360YC-FX2 スイッチの同じまたは異なる前面パネル ポート上にあります。

- Cisco NX-OS リリース 10.2(2)F では、FEX ND-ISSU セットアップが正常に行われるように **BGP graceful-restart restart time** コマンドを再調整する必要があります。これは、FEX アップグレードごとに 1 つずつ実行する必要があります。

次の例は、中断のない FEX アップグレードごとに **bgp-graceful restart-time** を再調整するのにかかる時間を示しています。

```
In the Non-disruptive upgrade with FEX, each FEX will upgrade taking about 90 secondss
(1.5 minutess) sequentially (one-by-one and not a parallel upgrade).
Total non-disruptive upgrade time for all FEX = No. of fex * time taken per fex
For 10 FEX = 10 * 90
= 900 seconds or 15 minutes
```

- 以前のリリースから Cisco NX-OS リリース 10.2(2)F に対して ND ISSU を実行する場合、MPLS ストリップ、GRE ストリップ、および基礎となる ACL 構成は ISSU 互換ではありません。

以前のリリースから Cisco NX-OS リリース 10.2(2)F への ND ISSU の後に、ポスト GRE ストリップ **dot1q トンネル VLAN\_tag** が欠落している可能性があります。この問題を回避するには、GRE ストリップ対応インターフェイスの L2 インターフェイスからポート ACL を削除して追加します。

- すべてのリリースの ISSU 互換性については、『ISSU サポート マトリックス』を参照してください。
- Cisco NX-OS リリース 10.1(1) 以降のリリース/ソフトウェア イメージの 64 ビット イメージへの無停止アップグレードはサポートされていません。アップグレードは中断を伴います。
- Cisco NX-OS リリース 10.1 (1) 以降、64 ビット イメージへの中断的アップグレードまたは 64 ビット イメージから 32 ビット イメージへのダウングレード中に、機能 ITD が有効になっている場合は、『Cisco Nexus 9000 Series NX-OS Intelligent Traffic Director Configuration Guide, Release 10.1(x)』の「Guidelines and Limitations for ITD」の ITD のガイドラインと制限事項を参照してください（ASCII リロードを使用してアップグレードまたはダウングレードを行う場合）。
- 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。install allno-reload この状態で設定を保存すると、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。
- LXC ブートモードのスイッチの場合、Cisco NX-OS リリース 10.1 (1) 以前のリリースから Cisco NX-OS リリース 10.2 (x) にイメージをアップグレードすると、アップグレードが中断されます。
- Cisco NX-OS リリース 10.2(3)F 以降、LXC モードで非破壊的なスイッチの場合、install コマンドに新しいオプション skip-kernel-upgrade が追加されました。
- 以下は、LXC モードで ND ISSU を実行できる 2 つの方法です。
  - LXC モードの ND ISSU - EOR に類似したスイッチオーバーベースの ISSU。2 番目の SUP が新しいコンテナで起動され、切り替えが行われます。2 番目の SUP が新たにアクティブになります。カーネルはアップグレードされません。
  - フォールバック ND LXC ISSU：これは、上記のスイッチオーバーベースの ISSU を実行できない場合にのみ実行されます（SRG カーネルに互換性がないか、メモリが少ない）。カーネルがアップグレードされます。
  - skip-kernel-upgrade オプションは、LXC モードで ND ISSU を強制する：スイッチオーバーベースの ISSU（実行中の場合でも）とターゲットカーネルに互換性はありません。
- Cisco NX-OS リリース 9.3(3) から Cisco NX-OS リリース 9.3(6) にアップグレードするとき、Cisco NX-OS リリース 9.3(3) から TRM 対応 VRF の設定を保持しない場合や、アップグレード後に新しい VRF を作成する場合、機能 ip multicast multipath s-g-hash next-hop-based が有効な際は ngmvpn CLI の自動生成は発生しません。TRM 対応 VRF ごとに CLI を手動で有効にする必要があります。設定手順については、『Cisco Nexus 9000 シリーズ NX-OS VXLAN 設定ガイド、リリース 10.1(x)』を参照してください。
- Cisco Nexus 9000 デバイスを Cisco NX-OS リリース 10.1(x) にアップグレードするときに、QSFP ポートが手動ブレイクアウト コマンドで設定され、QSA を使用している場合、インターフェイスイーサネット 1/50/1 の設定はサポートされず、削除する必要があります。

設定を復元するには、デバイスのインターフェイスイーサネット 1/50 を手動で設定する必要があります。

- 拡張 ISSU : Cisco NX-OS リリース 10.1(x) への拡張 ISSU はサポートされていません。これは、含まれているカーネル更新を有効化するために基盤となるカーネルのリロードが必要となるためです。02-04-2019 06:38 システムに次のメッセージが表示されます。

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

実際に、システムは拡張 ISSU の代わりに、非中断 ISSU を実行します。

- Cisco NX-OS リリース 9.2 (4) 以前のリリースから Cisco NX-OS リリース 9.3 (4) 以降にアップグレードする場合、実行コンフィギュレーションに追加の TCAM 設定行が含まれます。これらの余分な行はアップグレードおよび設定に影響しないため、無視できます。
- Cisco NX-OS リリース 9.3 (1) または 9.3 (2) から Cisco NX-OS リリース 9.3 (3) 以降に ISSU を実行する場合は、<ssh port> が規定のポート範囲内にある。ポート範囲が正しくない場合は、syslog メッセージの推奨事項に従ってください。詳細については、『Cisco Nexus 9000 シリーズ NX-OS IP SLA 設定ガイド、リリース 10.1(x)』を参照してください。
- Cisco NX-OS リリース 9.2(2) 以前のリリースから Cisco NX-OS リリース 10.1(x) にアップグレードする場合は、入力 RACL TCAM リージョンが 50% 以上使用されていないことを確認する必要があります。そうでない場合、アトミック更新機能はアップグレード後に有効になり、TCAM 割り当てが 50% を超える RACL を持つインターフェイスはダウンしたままになります。
- Cisco NX-OS リリース 10.1(1) 以降、ISSU は Cisco Nexus 93360YC-FX2 の FC/FCoE スイッチモードでサポートされます。FC/FCoE スイッチモードとサポートされるハードウェアの詳細については、『Cisco Nexus 9000 シリーズ NX-OS SAN 切り替え設定ガイド、リリース 10.1(x)』を参照してください。
- Cisco NX-OS リリース 10.1 (1) 以降では、Cisco Nexus 93180YC-FX および 93360YC-FX2 スイッチの FC / FCoE スイッチモードで Enhanced ISSU がサポートされています。FC/FCoE スイッチモードとサポートされるハードウェアの詳細については、『Cisco Nexus 9000 シリーズ NX-OS SAN 切り替え設定ガイド、リリース 10.1(x)』を参照してください。
- Cisco NX-OS リリース 10.1 (1) 以降では、Cisco Nexus 93180YC-FX および 93360YC-FX2 スイッチの FC / FCoE NPV モードで Enhanced ISSU がサポートされています。FC/FCoE NPV モードとサポートされるハードウェアの詳細については、『Cisco Nexus 9000 シリーズ NX-OS FC -NPV FCoE NPV 設定ガイド、リリース 10.1(x)』を参照してください。
- 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。
- Cisco Nexus 3000 シリーズの圧縮イメージはハードウェアに依存し、CCO から圧縮またはダウンロードされた同じデバイスでのみ使用できます。Nexus 9000 シリーズで Nexus 3000 シリーズの圧縮イメージを使用しないでください。
- 次の制限は、7.0(3)I5 から 10.1(x) または 9.2(3) から 10.1(x) へのソフトウェアアップグレードに適用されます。

VLAN と SVI の両方で同じ NetFlow 設定を使用している場合は、アップグレード前に VLAN 設定から NetFlow フローモニタを削除する必要があります。アップグレードしたら、新しいフローモニタを作成し、VLAN 設定に追加して、NetFlow を再設定します。これらの手順を実行しないと、エラーメッセージが表示され、アップグレードされたソフトウェアの VLAN NetFlow 設定を変更できなくなります。

- Cisco NX-OS リリース 7.0(3)I4(8)、7.0(3)I5(3)、および 7.0(3)I6(1) から Cisco NX-OS リリース 10.1(x) にアップグレードすると、中断を伴うアップグレードが発生します。Cisco NX-OS リリース 7.0(3)I4(8)、7.0(3)I5(3) または 7.0(3)I6(1) から 10.1(x) への中断を伴うアップグレード中に、スタンバイ SUP への画像の同期が失敗した場合、手動で画像をスタンバイ SUP にコピーし、中断を伴うアップグレードを実行する必要があります。
- 7.0 (x) より前のリリースから Cisco NX-OS リリース 10.1 (x) に直接アップグレードすると、アップグレードが中断されます。中断のないアップグレードでは、Cisco NX-OS リリース 9.x への中間アップグレードが必要です。アップグレードの中間ホップとして、Cisco NX-OS Release 9.3 (x) の最新リリースにアップグレードすることを推奨します。サポートされるアップグレードパスの詳細については、『[ISSU Support Matrix](#)』を参照してください。
- Cisco NX-OS リリース 7.0(3)I6(1) または 7.0(3)I7(1) から Cisco NX-OS リリース 10.1(x) にアップグレードするとき (Cisco Nexus 9000 シリーズ スイッチが vPC を実行していて、レイヤ 2 vPC を介した IOS ベースのスイッチに接続されている場合)、IOS 側のレイヤ 2 ポート チャネルがエラー無効になる可能性があります。回避策は、IOS スイッチで `spanning-tree etherchannel guard misconfig` コマンドを無効にしてから、アップグレードプロセスを開始することです。

両方の Cisco Nexus 9000 シリーズ スイッチをアップグレードしたら、コマンドを再度有効にできます。

- **install all** コマンドを使用して Cisco NX-OS リリース 7.0(3)I5(2) から Cisco NX-OS リリース 10.1(x) にアップグレードする場合、CSCve24965 により BIOS はアップグレードされません。Cisco NX-OS リリース 10.1(x) へのアップグレードが完了したら、**install all** コマンドを再度使用して BIOS のアップグレードを完了します (該当する場合)。
- Cisco NX-OS リリース 7.0(3)I2(2b) からリリース 10.1(x) への **install all** コマンドを使用して実行されたアップグレードにより、VLAN を既存の FEX HIF トランク ポートに追加することができなくなります。これから回復するには、すべての FEX がオンラインになり、HIF が動作するようになった後、次の手順を実行する必要があります。
  1. プロンプトで `copy run bootflash:fex_config_restore.cfg` コマンドを入力します。
  2. プロンプトで `copy bootflash:fex_config_restore.cfg running-config echo-commands` コマンドを入力します。
- Cisco NX-OS リリース 7.0(3)I6(1) および以前、ASCII リプレイの実行、または FEX HIF 設定で `copy file run` コマンドを実行するには、FEX の復帰後に手動で FEX の設定を再適用する必要があります。

- 7.0(3)I2(x) 以前から Cisco NX-OS リリース 10.1(x) にアップグレードし、EVPN VXLAN 設定を実行している場合は、7.0(3)I4(x) または 7.0(3)I5(x) または 7.0(3)I6(x) への中間アップグレードが必要です。
- インターフェイスで FHS を有効にする前に、Cisco Nexus 9300 および 9500 プラットフォーム スイッチで ifacl TCAM リージョンをカービングすることを推奨します。以前のリリースで ifacl TCAM リージョンを作成した場合は、Cisco NX-OS リリース 10.1 (x) にアップグレードした後にシステムをリロードする必要があります。システムをアップロードすると、FHS TCAM リージョン ifacl に必要な一致修飾子が作成されます。
- FHS を有効にする前に、Cisco Nexus 9200 および 9300-EX プラットフォーム スイッチで ing-redirect TCAM リージョンをカービングすることを推奨します。以前のリリースで ing-redirect TCAM リージョンを切り分けた場合は、Cisco NX-OS リリース 10.1(x) にアップグレードした後にシステムをリロードする必要があります。システムをアップロードすると、FHS TCAM リージョンである ing-redirect に必要な一致修飾子が作成されます。
- Cisco NX-OS リリース 9.3(1)、9.3(2) または 9.3(3) から、Embedded Event Manager (EEM) 設定が実行設定に保存されている上位リリースにアップグレードすると、DME エラーが表示されることがあります。エラーは、**show consistency-checker dme running-config enhanced** コマンド、特にイベント マネージャ コマンドの出力にあります。このエラーが発生した場合は、ISSU の完了後にすべての EEM アプレット設定を削除してから、EEM 設定を再適用します。
- ISSU を使用して Cisco NX-OS リリース 9.3(5) にアップグレードする以前のリリースバージョンでは、次のロギングレベルコマンドが設定されている場合、それらはアップグレードされたバージョンにないため、再設定する必要があります。
  - **logging level evmc value**
  - **logging level mvsh value**
  - **logging level fs-daemon value**
- ISSU を使用して Cisco NX-OS リリース 9.3(6) にアップグレードする以前のリリースバージョンでは、次のロギングレベルコマンドが設定されている場合、それらはアップグレードされたバージョンにないため、再設定する必要があります。
  - **logging level evmc value**
  - **logging level mvsh value**
- **copy running-config save-config** および **reload** コマンドを入力せずに予約済み VLAN を変更した場合、ISSU を実行しようとするエラーが発生します。
- **install all** コマンドは、構成の互換性チェックと BIOS のアップグレードを自動的に実行するため、ソフトウェア アップグレードとダウングレードに推奨される方法です。一方、ブート変数を変更してデバイスをリロードすると、これらのチェックと BIOS アップグレードがバイパスされるため、これは推奨されません。
- Cisco NX-OS リリース 7.0(3)I1(2)、リリース 7.0(3)I1(3)、またはリリース 7.0(3)I1(3a) からのアップグレードには、Cisco Nexus 9500 プラットフォーム スイッチのみのパッチをイン

ストールする必要があります。アップグレードパッチの詳細については、「パッチのアップグレード手順」を参照してください。

- ISSU は Cisco NX-OS リリース 7.0(3)I4(1) からそれ以降の画像へのみ実行できます。
- ISSU の実行中、VRRP および VRRPv3 は次のメッセージを表示します。

- VRRPv3 が有効の場合：

```
2015 Dec 29 20:41:44 MDP-N9K-6 %$ VDC-1 %$ %USER-0-SYSTEM_MSG: ISSU ERROR: Service
"vrrpv3" has sent the following message: Feature vrrpv3 is configured. User can
change
vrrpv3 timers to 120 seconds or fine tune these timers based on upgrade time on
all Vrrp
Peers to avoid Vrrp State transitions. - sysmgr
```

- VRRP が有効な場合：

```
2015 Dec 29 20:45:10 MDP-N9K-6 %$ VDC-1 %$ %USER-0-SYSTEM_MSG: ISSU ERROR: Service
"vrrp-
eng" has sent the following message: Feature vrrp is configured. User can change
vrrp
timers to 120 seconds or fine tune these timers based on upgrade time on all
Vrrp Peers to
avoid Vrrp State transitions. - sysmgr
```

- ゲスト シェルは、ISSU 実行中に無効化され、アップグレード後に再度アクティブ化されます。ゲスト シェルで実行されているすべてのアプリケーションが影響を受けます。
- ITD プローブを設定している場合は、Cisco NX-OS リリース 10.1(x) にアップグレードする前に、ITD サービスを無効にする必要があります (**shutdown** コマンドを使用)。アップグレード後、ITD プローブの IP SLA を有効にする **feature sla sender** コマンドを入力してから、**no shutdown** コマンドを入力して ITD サービスを再度有効にします。(サービスをシャットダウンせずにアップグレードする場合は、アップグレード後に **feature sla sender** コマンドを入力できます)。
- 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。使用しているシャーシの『[Hardware Installation Guide](#)』を参照してください。
- スタンバイ スーパーバイザ モジュールではなく、アクティブ スーパーバイザ モジュールでインストールを実行します。
- **install all** コマンドは、構成の互換性チェックと BIOS のアップグレードを自動的に実行するため、ソフトウェアアップグレードに推奨される方法です。一方、ブート変数を変更してデバイスをリロードすると、これらのチェックと BIOS アップグレードがバイパスされるため、これは推奨されません。



(注) -R ライン カードを装備した Cisco Nexus 9500 プラットフォーム スイッチでは、設定を保存してデバイスをリロードし、Cisco NX-OS リリース 7.0(3)F3(5) から 9.3(1) にアップグレードする必要があります。Cisco NX-OS リリース 9.2(2) または 9.2(3) にアップグレードするには、**install all** コマンドを使用することを推奨します。

- ソフトウェアイメージの MD5、SHA256、または SHA512 チェックサムを確認することにより、アップグレードを実行する前に、不完全または破損した NX-OS ソフトウェア イメージを検出できます。ソフトウェア イメージの MD5 チェックサムを確認するには、**show file bootflash: <IMAGE-NAME>md5sum** コマンドを実行し、結果の値を [シスコのソフトウェアダウンロード Web サイト](#) で公開されているソフトウェアイメージの MD5 チェックサムと比較します。ソフトウェアイメージの SHA512 チェックサムを確認するには、次のコマンドを実行します。**show file bootflash: <IMAGE-NAME>sha512sum** コマンドを実行し、結果の値を [シスコのソフトウェアダウンロード Web サイト](#) で公開されているソフトウェアイメージの SHA512 チェックサムと比較します。
- Cisco Nexus 94xx、95xx、96xx ラインカードから Cisco Nexus 9732C-EX ラインカードおよびそのファブリック モジュールにアップグレードする場合は、ラインカードとファブリック モジュールを挿入する前に Cisco NX-OS ソフトウェアをアップグレードします。そうしないと、ラインカードで診断エラーが発生し、TCAM スペースが割り当てられなくなる可能性があります。**write\_erase** コマンドの後に **reload** コマンドを使用する必要があります。
- 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、『Cisco Nexus 9000 シリーズ NX-OS セキュリティ設定ガイド、リリース 10.1(x)』の「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。
- POAP の安全性を確保するために、DHCP スヌーピングが有効であることを確認し、ファイアウォールルールを設定して意図しない、または悪意のある DHCP サーバをブロックしてください。詳細については、『Cisco Nexus 9000 シリーズ基本設定ガイド、リリース 10.1(x)』を参照してください。
- 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、実行コンフィギュレーション 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、『Cisco Nexus 9000 シリーズ NX-OS システム管理設定ガイド、リリース 10.1(x)』を参照してください。
- 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。
- OpenFlow および LACP 高速タイマー レート設定は、ISSU ではサポートされません。



- ゲスト シェルは、ISSU 実行中に無効化され、アップグレード後に再度アクティブ化されます。
- ISSU は、BGP ピアのデフォルトのホールド タイマーのみをサポートします。
- Cisco Nexus 9300 シリーズ スイッチの ISSU の間、ISSU の対象のノードがアクティブであると、すべてのファーストホップ冗長プロトコル (FHRP) は他のピアをアクティブにします。
- VPC の両方のピアが同じモード (通常モードまたは拡張モード) であることを確認してから、無停止アップグレードを実行してください。



---

(注) 拡張 ISSU モード (ブートモード `lxc`) が設定されたスイッチと非拡張 ISSU モードスイッチ間の vPC ピアリングはサポートされていません。

---

- ISSU では、最初の vPC デバイス上のソフトウェアリロードプロセスで、vPC 通信チャンネルを介した CFS メッセージングを使用して、その vPC ピア デバイスをロックします。1 度に 1 つのデバイスだけアップグレードできます。最初のデバイスは、そのアップグレードが完了したら、そのピアデバイスのロックを解除します。次に、2 つ目のデバイスが、最初のデバイスが行ったのと同じように最初のデバイスをロックして、アップグレードプロセスを実行します。アップグレード中は、2 つの vPC デバイスが一時的に異なるリリースの Cisco NX-OS を実行することになりますが、その下位互換性サポートにより、システムは正常に機能します。
- OnePK が有効である場合、ISSU はサポートされません。 `show feature | include onep` コマンドを実行してこの機能が無効化されていることを確認してから、ISSU または拡張 ISSU を実行できます。
- 通常、ISSU では次のアップグレードがサポートされています。
  - メジャー リリースから関連するすべてのメンテナンス リリースへ
  - 最後の 2 つのメンテナンス リリースから次の 2 つのメジャー リリースへ
  - 以前のメンテナンス リリースから次の 2 つのメジャー リリースへ



---

(注) 中断のあるアップグレードまたは中断のない ISSU の実行が可能な特定のアップグレード元リリースのリストについては、お使いの特定のリリースの『[Cisco Nexus 9000 シリーズ NX-OS リリースノート](#)』を参照してください。

---

- Cisco Nexus 9300 プラットフォーム スイッチおよび Cisco Nexus 3164Q スイッチで ISSU を実行した後に、vPC ピアで MTS\_OPC\_CLISH メッセージが表示される可能性があります。

MTS\_OPC\_CLISHは、show コマンド出力の終わりを指定するためにバックエンドコンポーネントから VSH に送信される最後の MTS コードです。

show コマンドを実行して大量の出力が生成され、3分以上セッションが持続する場合、次の警告メッセージがコンソールに表示される場合があります。回避策として、**terminal length 0** コマンドまたは **show <command> | no-more** オプションを使用してターミナル長を 0 に設定できます。

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

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

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

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

```
(60.60.60.2/32, 225.0.0.1/32), uptime: 00:09:31, ip(0) mrrib(1) pim(0)
RPF-Source: 60.60.60.2 [20/110]
Data Created: Yes
VPC Flags
```

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

この問題により機能への影響やトラフィックの損失は生じません。show コマンドで出力がすべて表示されるか、ユーザが Ctrl+C を入力するか、セッションが終了すると、MTS メッセージの表示がすべて終了します。

- まれに、スイッチが正常に稼働しているときに Device not found ログがコンソールに表示される場合があります。この問題は、スイッチが古い ASIC バージョンを見つけようとして、さらに PCI プローブ障害のエラーメッセージがコードで有効になっているために発生します。この問題により機能への影響やトラフィックの損失は生じません。
- Cisco NX-OS リリース 7.0(3)I3(1)以降で EPLD が存在しない場合は、ISSU はサポートされません。

- ISSUは、install all nxosを使用したEPLDイメージのアップグレードをサポートします。  
<nxos-image> epld<epld-image>破壊的なシステム (NX-OS) のアップグレード中のコマンド。
- Cisco NX-OS 10.1(x) リリースでサポートされるプラットフォームに、簡素化されたNX-OS 番号付け形式が使用されています。古いリリース形式の Cisco NX-OS リリース 7.0(3)I7(4) より前のリリースからのソフトウェア アップグレードをサポートするために、**install all** の操作中はインストーラの機能によって I9(x) ラベルがサフィックスとして実際のリリースに付与されます。このラベルは、Cisco NX-OS リリース 7.0(3)I7(4) より前のリリースから 10.1(x) へのインストール操作中にイメージの一部として表示されますが、これは無視できません。次の例を参照してください。

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

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

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

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

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

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

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

Compatibility check is done:
Module  bootable  Impact          Install-type  Reason
-----  -
1       yes         disruptive      reset         Incompatible image for ISSU

Images will be upgraded according to following table:
Module  Image          Running-Version(pri:alt)  New-Version
Upg-Required
-----  -
1       nxos          7.0(3)I7(3)              9.3(1)I9(1)
yes
1       bios          v07.61(04/06/2017):v07.61(04/06/2017)  v05.33(09/08/2018)
yes

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

- uRPf が設定されたスイッチ上の Cisco NX-OS リリース 9.3(5)、標準、中断のない ISSU 以降では、次でサポートされます。
  - Cisco Nexus 9300-EX プラットフォーム スイッチ
  - Cisco Nexus 9300-FX/FX2 プラットフォーム スイッチ

- Cisco Nexus 9300-GX プラットフォーム スイッチ



(注) Cisco NX-OS リリース 9.3(5) 以前では、上記のスイッチのいずれかが uRPF で設定されている場合、標準、中断のない ISSU はサポートされていませんでした。

- **boot poap enable** が設定されている場合、ISSUはブロックされます。
- Cisco NX-OS リリース 7.0 (3) I6 (1) から上位バージョンへの無停止ISSUを実行すると、設定されているVLANの数に基づいてトラフィック損失が発生する可能性があります。トラフィック損失を回避するために、ルーティングプロトコルのグレースフルリスタートタイマーの値を大きくすることを推奨します。グレースフルリスタートタイマーの推奨値は600秒です。この値は、設定の規模に応じてさらに増減できます。
- Cisco NX-OS リリース 10.1 (1) 以降、Fs\_daemonは5000を超えるファイルを持つデバイスでのsnmpwalkをサポートしません。5000を超えるファイルがあるデバイスでsnmpwalkを実行する場合、エラーresourceUnavailable（これはエージェント内のメモリ不足障害である可能性が高い）が予想される動作です。
- Cisco NX-OS リリース 10.1 (2) 以降、CoPP は N9K-X9624D-R2 および N9K-C9508-FM-R2 プラットフォーム スイッチでサポートされ。
- Cisco NX-OS リリース 10.1(2) 以降、RACL は N9K-X9624D-R2 および N9K-C9508-FM-R2 プラットフォーム スイッチでサポートされ。
- Cisco NX-OS リリース 9.3 (5) 、 9.3 (6) 、 9.3 (7) 、 10.1 (1) 、 または 10.1 (2) から Cisco NX-OS リリース 10.2 (1) 以降のリリースへのISSUの実行中ブロックされます。
- 遅延設定がトラックリストのブルール/重みに存在する場合、ISSUはブロックされます。
- ISSU中にIPv6NDがタイムアウトすると、ISSUの後にIPv6BFDセッションがフラップする可能性があります。
- Cisco NX-OS リリース 10.2(3)F 以降、すべての Cisco Nexus 9300-X TOR の VPC ファブリック ピ어링で非中断 ISSU がサポートされます。標準と拡張の両方の無停止アップグレードがサポートされています。障害が発生していないときにISSUを開始またはトリガーする必要があることに注意してください。障害の例としては、VPCレグの1つがダウンしていることが挙げられます。
- 推奨されるルーティングプロトコルのグレースフルリスタートタイマーは600秒で、nve送信元インターフェイスのホールドダウン時間は400秒です。
- スイッチモードテストベッドでNDネイティブISSUを呼び出す場合は、EまたはFモードのVFCインターフェイスで**disable-fka**と設定することをお勧めします。そうでない場合、中断が発生する可能性があります。
- VRFスケールがある場合、各VRFの非中断型ISSUでは、グレースフルリスタートタイマーを300秒に構成する必要があります。

- 任意のリリースから nxos64-cs.10.3(1)F 以降のリリースにアップグレードする必要があるプラットフォームの場合、nxos.9.3.10.bin または nxos64-cs.10.2(3)F 以降のリリースを暫定ホップとして使用します。この制限は、中断および非中断アップグレードの両方で適用されます。nxos64-msll.10.3(1)F にはこの制限がありません。
- サポートされていないイメージを Cisco Nexus 9800 プラットフォーム スイッチにロードすると、スイッチがスタックします。その場合、リセットできるのは電源の再投入のみです。
- 9.3(10)または10.2(3)Fより前のバージョンからの中断を伴うアップグレードは、CSCwb63451 が原因で失敗する場合があります。10.3(1)F 以降にアップグレードする前に、まず9.3(10)または10.2(3)Fにアップグレードする必要があります。
- vPC コンプレックスまたは LAN シナリオの L2 スイッチで ISSU を実行している間、L2 スイッチはコントロールプレーンのダウンタイム中にレポート/クエリを転送できないため、IGMP グループ タイムアウトをより高い値で構成する必要があります。L2 スヌーピング クエリア間隔も、L3 クエリア間隔と一致する必要があります。

## ISSU プラットフォームのサポート

次の表に、標準および拡張 ISSU をサポートするプラットフォームと、サポートが導入されたときのリリースを示します。



- (注) 拡張 ISSU : Cisco NX-OS リリース 10.1(x) への拡張 ISSU はサポートされていません。これは、含まれているカーネル更新を有効化するために基盤となるカーネルのリロードが必要となるためです。02-04-2019 06:38 システムに次のメッセージが表示されます。

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

実際に、システムは拡張 ISSU の代わりに、非中断 ISSU を実行します。

### Cisco Nexus 9200 プラットフォーム スイッチの ISSU

ISSU タイプ	リリース/サポートされるプラットフォーム	中断のないISSUでサポートされない機能
標準	Cisco NX-OS リリース 7.0(3)I6(1) 以降 : Cisco Nexus 92300YC  Cisco NX-OS リリース 9.3(3) 以降 : Cisco Nexus 92348GC-X	次の機能が設定された Cisco Nexus 9200 プラットフォーム スイッチでは、どちらの ISSU タイプも中断を伴います。 <ul style="list-style-type: none"> <li>• セグメント ルーティング</li> </ul>
拡張	Cisco Nexus 92300YC	<ul style="list-style-type: none"> <li>• Tetration</li> </ul>

## Cisco Nexus 9300 プラットフォーム スイッチの ISSU

ISSU タイプ	リリース/サポートされるプラットフォーム	中断のない ISSU でサポートされない機能
標準	<p>Cisco NX-OS リリース 9.3(3) 以降 :</p> <p>Cisco Nexus 9332C</p> <p>Cisco Nexus 9364C</p> <p>(注) Cisco Nexus 9300 プラットフォームスイッチの ISSU は、スイッチがスパニングツリールートの場合サポートされます。スイッチがこの条件を満たしているかどうかを確認するには、<b>show spanning-tree issu-impact</b> コマンドを使用できます。</p>	<p>次の機能が設定された Cisco Nexus 9300 プラットフォームスイッチでは、どちらの ISSU タイプも中断を伴います。</p> <ul style="list-style-type: none"> <li>• デュアルホーム FEX</li> <li>• セグメントルーティング</li> <li>• VXLAN</li> </ul>
拡張	<p>Cisco NX-OS リリース 9.3(5) 以降 :</p> <p>Cisco Nexus 9332C</p> <p>Cisco Nexus 9364C</p> <p>(注) Cisco Nexus 9300 プラットフォームスイッチの ISSU は、スイッチがスパニングツリールートの場合サポートされます。スイッチがこの条件を満たしているかどうかを確認するには、<b>show spanning-tree issu-impact</b> コマンドを使用できます。</p>	

## Cisco Nexus 9300-X プラットフォーム スイッチの ISSU

ISSU タイプ	リリース/サポートされるプラットフォーム	中断のないISSUでサポートされない機能
標準	Cisco NX-OS リリース 10.2(3)F 以降、VPC ファブリック ピ어링は Cisco Nexus 9300-X TOR でサポートされています。	Cisco NX-OS リリース 10.2(3)F 以降、VPC ファブリック ピ어링の非中断型 ISSU 中には、次の VXLAN/VPC 機能はサポートされません。 <ul style="list-style-type: none"> <li>• TRM</li> <li>• VXLAN IPv6 アンダーレイ</li> <li>• VNF の比例マルチパス</li> <li>• VXLAN フラッドアンドラーニング</li> <li>• HSRP および VRRP</li> <li>• VXLAN Cloudsec</li> <li>• VXLAN から SR へのハンドオフとすべてのハンドオフ機能</li> <li>• マルチサイト</li> </ul>
Enhanced	Cisco NX-OS リリース	

## Cisco Nexus 9300-EX プラットフォーム スイッチの ISSU

ISSU タイプ	リリース/サポートされるプラットフォーム	中断のないISSUでサポートされない機能
標準	Cisco NX-OS リリース 7.0(3)I6(1) 以降： Cisco Nexus 93108TC-EX Cisco Nexus 93180YC-EX	次の機能が設定された Cisco Nexus 9300-EX プラットフォーム スイッチでは、どちらの ISSU タイプも中断を伴います。
拡張	Cisco NX-OS リリース 7.0(3)I7(3) 以降： Cisco Nexus 93108TC-EX Cisco Nexus 93180YC-EX	<ul style="list-style-type: none"> <li>• セグメント ルーティング</li> <li>• Tetration</li> </ul> <p>(注) Cisco NX-OS リリース10.2 (1) 以降では、両方の ISSUタイプが、ストレートスルー FEXおよびデュアルホーム接続FEXで設定されたCisco Nexus 9300-EXプラットフォームスイッチで中断されません。</p>



## Cisco Nexus 9300-FX プラットフォーム スイッチの ISSU

ISSU タイプ	リリース/サポートされるプラットフォーム	中断のないISSUでサポートされない機能
標準	Cisco NX-OS リリース 9.3(1) および 9.3(2) : なし Cisco NX-OS リリース 9.3(3) 以降 : Cisco Nexus 9336C-FX2 Cisco Nexus 93240YC-FX2 Cisco Nexus 93240YC-FX2Z Cisco Nexus 9348GC-FXP Cisco Nexus 93108TC-FX Cisco Nexus 93180YC-FX	標準ISSUは、次の機能が設定された Cisco Nexus 9300-FX プラットフォーム スイッチに影響を与えます。 <ul style="list-style-type: none"> <li>• セグメントルーティング</li> <li>• TRM 機能</li> </ul> (注) Cisco NX-OS リリース10.2 (1) 以降、標準ISSUは、ストレートスルーFEXおよびデュアルホーム接続FEXで設定されたCisco Nexus 9300-FXプラットフォームスイッチで中断がありません。

ISSU タイプ	リリース/サポートされるプラットフォーム	中断のないISSUでサポートされない機能
拡張	<p>Cisco NX-OS リリース 9.3(1)、9.3(2)、および 9.3(3) : なし</p> <p>Cisco NX-OS リリース 9.3(5) 以降 :</p> <p>Cisco Nexus 9336C-FX2</p> <p>Cisco Nexus 93240YC-FX2</p> <p>Cisco Nexus 93216TC-FX2</p> <p>Cisco Nexus 93360YC-FX2</p> <p>Cisco Nexus 93240YC-FX2Z</p> <p>Cisco Nexus 9348GC-FXP</p> <p>Cisco Nexus 93108TC-FX</p> <p>Cisco Nexus 93180YC-FX</p> <p>Cisco NX-OSリリース10.1 (1) 以降、Enhanced ISSUはFC / FCoE機能を備えた次のプラットフォームでサポートされます。</p> <p>Cisco Nexus 93360YC-FX2</p> <p>Cisco Nexus 93180YC-FX</p>	<p>拡張ISSUは、次の機能が設定された Cisco Nexus 9300-FX プラットフォーム スイッチに影響を与えます。12-12-2020 17:2</p> <ul style="list-style-type: none"> <li>セグメント ルーティング</li> <li>TRM 機能</li> </ul> <p>(注) Cisco NX-OSリリース9.3 (x) では、FC / FCoE機能を搭載したCisco Nexus 93360YC-FX2およびCisco Nexus 93180YC-FXでの拡張ISSUは中断を伴います。</p> <p>(注) Cisco NX-OSリリース10.2 (1) 以降、Enhanced ISSUは、ストレートスルーFEXおよびデュアルホーム接続FEXで設定されたCisco Nexus 9300-FXプラットフォームスイッチで中断が発生しません。</p>

## Cisco Nexus 9300-GX プラットフォーム スイッチの ISSU

ISSU タイプ	リリース/サポートされるプラットフォーム	中断のないISSUでサポートされない機能
標準	<p>Cisco NX-OS リリース 10.1(1) 以降 :</p> <p>Cisco Nexus 9364C-GX</p> <p>Cisco Nexus 9316D-GX</p> <p>Cisco Nexus 93600CD-GX</p>	<ul style="list-style-type: none"> <li>TRM 機能</li> <li>セグメント ルーティング</li> </ul>

ISSU タイプ	リリース/サポートされるプラットフォーム	中断のないISSUでサポートされない機能
Enhanced	Cisco NX-OS リリース 10.1(1) 以降： Cisco Nexus 9364C-GX Cisco Nexus 9316D-GX Cisco Nexus 93600CD-GX	<ul style="list-style-type: none"> <li>• TRM 機能</li> <li>• セグメント ルーティング</li> </ul>

## ISSU プラットフォームのサポート

次の表に、標準および拡張 ISSU をサポートするプラットフォームと、サポートが導入されたときのリリースを示します。



- (注) 拡張 ISSU : Cisco NX-OS リリース 10.1(x) への拡張 ISSU はサポートされていません。これは、含まれているカーネル更新を有効化するために基盤となるカーネルのリロードが必要となるためです。02-04-2019 06:38 システムに次のメッセージが表示されます。

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

実際に、システムは拡張 ISSU の代わりに、非中断 ISSU を実行します。

### Cisco Nexus 9200 プラットフォームスイッチの ISSU

ISSU タイプ	リリース/サポートされるプラットフォーム	中断のないISSUでサポートされない機能
標準	Cisco NX-OS リリース 7.0(3)I6(1) 以降： Cisco Nexus 92300YC  Cisco NX-OS リリース 9.3(3) 以降： Cisco Nexus 92348GC-X	次の機能が設定された Cisco Nexus 9200 プラットフォームスイッチでは、どちらの ISSU タイプも中断を伴います。 <ul style="list-style-type: none"> <li>• セグメント ルーティング</li> </ul>
拡張	Cisco Nexus 92300YC	<ul style="list-style-type: none"> <li>• Tetration</li> </ul>

## Cisco Nexus 9300 プラットフォーム スイッチの ISSU

ISSU タイプ	リリース/サポートされるプラットフォーム	中断のない ISSU でサポートされない機能
標準	<p>Cisco NX-OS リリース 9.3(3) 以降 :</p> <p>Cisco Nexus 9332C</p> <p>Cisco Nexus 9364C</p> <p>(注) Cisco Nexus 9300 プラットフォームスイッチの ISSU は、スイッチがスパニングツリールートの場合サポートされます。スイッチがこの条件を満たしているかどうかを確認するには、<b>show spanning-tree issu-impact</b> コマンドを使用できます。</p>	<p>次の機能が設定された Cisco Nexus 9300 プラットフォームスイッチでは、どちらの ISSU タイプも中断を伴います。</p> <ul style="list-style-type: none"> <li>• デュアルホーム FEX</li> <li>• セグメントルーティング</li> <li>• VXLAN</li> </ul>
拡張	<p>Cisco NX-OS リリース 9.3(5) 以降 :</p> <p>Cisco Nexus 9332C</p> <p>Cisco Nexus 9364C</p> <p>(注) Cisco Nexus 9300 プラットフォームスイッチの ISSU は、スイッチがスパニングツリールートの場合サポートされます。スイッチがこの条件を満たしているかどうかを確認するには、<b>show spanning-tree issu-impact</b> コマンドを使用できます。</p>	

## Cisco Nexus 9300-EX プラットフォーム スイッチの ISSU

ISSU タイプ	リリース/サポートされるプラットフォーム	中断のないISSUでサポートされない機能
標準	Cisco NX-OS リリース 7.0(3)I6(1) 以降： Cisco Nexus 93108TC-EX Cisco Nexus 93180YC-EX	次の機能が設定された Cisco Nexus 9300-EX プラットフォーム スイッチでは、どちらの ISSU タイプも中断を伴います。
拡張	Cisco NX-OS リリース 7.0(3)I7(3) 以降： Cisco Nexus 93108TC-EX Cisco Nexus 93180YC-EX	<ul style="list-style-type: none"> <li>• セグメントルーティング</li> <li>• Tetration</li> </ul> <p>(注) Cisco NX-OS リリース10.2 (1) 以降では、両方の ISSUタイプが、ストレートスルー FEXおよびデュアルホーム接続FEXで設定されたCisco Nexus 9300-EXプラットフォームスイッチで中断されません。</p>

## Cisco Nexus 9300-FX プラットフォーム スイッチの ISSU

ISSU タイプ	リリース/サポートされるプラットフォーム	中断のないISSUでサポートされない機能
標準	<p>Cisco NX-OS リリース 9.3(1) および 9.3(2) : なし</p> <p>Cisco NX-OS リリース 9.3(3) 以降 :</p> <p>Cisco Nexus 9336C-FX2</p> <p>Cisco Nexus 93240YC-FX2</p> <p>Cisco Nexus 93240YC-FX2Z</p> <p>Cisco Nexus 9348GC-FXP</p> <p>Cisco Nexus 93108TC-FX</p> <p>Cisco Nexus 93180YC-FX</p> <p>Cisco NX-OS リリース 10.2(1)F 以降 :</p> <p>Cisco Nexus 93180YC-FX3</p> <p>Cisco Nexus 93180YC-FX3S</p>	<p>標準ISSUは、次の機能が設定された Cisco Nexus 9300-FX プラットフォーム スイッチに影響を与えます。</p> <ul style="list-style-type: none"> <li>• セグメント ルーティング</li> <li>• TRM 機能</li> </ul> <p>(注) Cisco NX-OS リリース10.2 (1) 以降、標準ISSUは、ストレートスルー FEXおよびデュアルホーム接続FEXで設定されたCisco Nexus 9300-FXプラットフォームスイッチで中断がありません。</p>

ISSU タイプ	リリース/サポートされるプラットフォーム	中断のないISSUでサポートされない機能
拡張	<p>Cisco NX-OS リリース 9.3(1)、9.3(2)、および 9.3(3) : なし</p> <p>Cisco NX-OS リリース 9.3(5) 以降 :</p> <p>Cisco Nexus 9336C-FX2</p> <p>Cisco Nexus 93240YC-FX2</p> <p>Cisco Nexus 93216TC-FX2</p> <p>Cisco Nexus 93360YC-FX2</p> <p>Cisco Nexus 93240YC-FX2Z</p> <p>Cisco Nexus 9348GC-FXP</p> <p>Cisco Nexus 93108TC-FX</p> <p>Cisco Nexus 93180YC-FX</p> <p>Cisco NX-OSリリース10.1 (1) 以降、Enhanced ISSUはFC / FCoE機能を備えた次のプラットフォームでサポートされます。</p> <p>Cisco Nexus 93360YC-FX2</p> <p>Cisco Nexus 93180YC-FX</p> <p>Cisco NX-OSリリース10.2(1)F 以降、Enhanced ISSU は次のプラットフォームでサポートされます。</p> <p>Cisco Nexus 93180YC-FX3</p> <p>Cisco Nexus 93180YC-FX3S</p> <p>Cisco NX-OSリリース10.2(2)F 以降、Enhanced ISSU は FC/FCoE 機能を備えた次のプラットフォームでサポートされます。</p> <p>N9K-C9336C-FX2-E</p>	<p>拡張ISSUは、次の機能が設定された Cisco Nexus 9300-FX プラットフォーム スイッチに影響を与えます。12-12-2020 17:2</p> <ul style="list-style-type: none"> <li>セグメント ルーティング</li> <li>TRM 機能</li> </ul> <p>(注) Cisco NX-OS リリース9.3 (x) では、FC/FCoE機能を搭載したCisco Nexus 93360YC-FX2およびCisco Nexus 93180YC-FXでの拡張ISSUは中断を伴います。</p> <p>(注) Cisco NX-OS リリース10.2 (1) 以降、Enhanced ISSU は、ストレートスルーFEXおよびデュアルホーム接続FEXで設定されたCisco Nexus 9300-FXプラットフォームスイッチで中断が発生しません。</p>

## Cisco Nexus 9300-GX プラットフォーム スイッチの ISSU

ISSU タイプ	リリース/サポートされるプラットフォーム	中断のないISSUでサポートされない機能
標準	<p>Cisco NX-OS リリース 10.1(1) 以降 :</p> <p>Cisco Nexus 9364C-GX</p> <p>Cisco Nexus 9316D-GX</p> <p>Cisco Nexus 93600CD-GX</p>	<ul style="list-style-type: none"> <li>TRM 機能</li> <li>セグメント ルーティング</li> </ul>

ISSU タイプ	リリース/サポートされるプラットフォーム	中断のないISSUでサポートされない機能
Enhanced	<p>Cisco NX-OS リリース 10.2(2)F 以降、Enhanced ISSU は Cisco Nexus 9300-GX2B プラットフォーム スイッチでサポートされます。</p> <p>Cisco NX-OS リリース 10.1(1) 以降：  Cisco Nexus 9364C-GX  Cisco Nexus 9316D-GX  Cisco Nexus 93600CD-GX</p>	<ul style="list-style-type: none"> <li>• TRM 機能</li> <li>• セグメント ルーティング</li> </ul>

## 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。

- Cisco Nexus 9000 シリーズ スイッチをダウングレードする唯一の方法は、`install all` コマンドを使用することです。ブート変数の変更、設定の保存、およびスイッチのリロードの各操作は、スイッチのダウングレードではサポートされていません。

Cisco NX-OS リリース 9.3(x) から以前のリリースにダウングレードする必要がある場合は、ゲスト シェルを無効にします。

- FCoE (ファイバチャネル オーバー イーサネット) NPV (N ポート仮想化) 設定で Cisco NX-OS リリース 9.3(x) からリリース 7.0(3)I4(1) への ISSU ダウングレードを実行すると、コア ファイルとともにポート チャネルがクラッシュします。

```
[##### ] 38%2016 Apr 18 20:52:35 n93-ns1 %$ VDC-1 %$ %SYSMGR-2-
SERVICE_CRASHED: Service "port-channel" (PID 14976) hasn't caught signal 11 (core
will
be saved)
```

- ISSU (中断なし) ダウングレードはサポートされていません。
- Cisco NX-OS リリース 9.3(x) から以前のリリースにダウングレードする場合、`statistic per-entry` コマンドが有効になっていて、RACL として適用されている ACL では、ダウングレード前に実行設定から `statistics per-entry` コマンドを削除する必要があります。そうしないと、この ACL が RACL として適用されるインターフェイスは、ダウングレード後にエラー無効になります。
- -FX または -FX + EX ライン カードを使用する Cisco Nexus 9500 シリーズ スイッチを、Cisco NX-OS リリース 10.1(x) から以前のリリース (9.2(x) または 7.x) にダウングレードする前に、NetFlow (ing-netflow) に適用する場合は、次のコマンドを使用してゼロ (0) に切り分ける必要があります。

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



9.2(x) のデフォルトが 0 の一方、9.3(1) 以降のデフォルトの `ing-netflow TCAM` リージョンが 512 のため、設定変更が必要です。

- Cisco NX-OS リリース 10.1(x) から 9.3(x) 以前のリリースにダウングレードする場合は、ラベル共有機能がない場合に、入力機能の ACL TCAM 使用率が割り当てられた TCAM スペースを超えないようにしてください。ラベル共有は、Cisco NX-OS リリース 9.3(x) の新機能です。そうしないと、TCAM に適合しなかった RACL を持つインターフェイスは、ダウングレード後に無効になります。
- ソフトウェア ダウングレードは、`install all` コマンドを使用して実行する必要があります。ブート変数の変更、設定の保存、およびスイッチのリロードの各操作は、スイッチのダウングレードではサポートされていません。
- 次の制限は、Trust Anchor Module (TAM) をサポートする Cisco Nexus プラットフォーム スイッチに適用されます。

Cisco NX-OS リリース 9.3(3) 以降から以前のバージョンにダウングレードする場合、TACACS グローバル キーは復元できません。TAM は 9.3(3) でバージョン 7 に更新されましたが、以前の NX-OS バージョンでは TAM バージョン 3 が使用されていました。

- iCAM は、リリース 9.2(x) またはリリース 9.3(x) → 7.0(3)I7(1) にダウングレードする前に無効にする必要があります。iCAM が有効になっている場合は、リリース 9.3(1) → リリース 9.2(4) のみを実行できます。
- Cisco NX-OS リリース 9.3(3) 以降では、SRAPP 用の新しい設定コマンドがあります (MPLS および SRTE のサブモード オプション付き)。リリース 9.3(3) (またはそれ以降) を実行しているスイッチの SRAPP 設定は、スイッチが以前のリリースにダウングレードされている場合は存在しません。
- 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。使用しているシャーシの『[ハードウェア設置ガイド](#)』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 (設定されている場合) を削除する必要があります。詳細については、『[Cisco Nexus 9000 シリーズ NX-OS システム管理設定ガイド](#)、リリース 10.1(x)』を参照してください。
- 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 リリース 10.2 (1) 以降から以前のリリースにダウングレードする場合、**install all** コマンドはブロックされます。
- 遅延設定がトラックリストのブール/重みに存在する場合、**install all** コマンドはブロックされます。
- Cisco NX-OS リリース 10.2(3)F からインターフェイスに **epbr L2** が適用された Cisco NX-OS リリース 10.2(2)F への ISSD を実行しているときに、ISSD を実行する前にインターフェイスからポリシーを削除して、トラックの重複の問題を回避します。
- Cisco NX-OS リリース 10.2(3)F 以降、**lldp chassis-id switch** コマンドを構成している場合は、ISSD を実行する前にコマンドを無効にする必要があります。
- 10.2(3)F 以降、アクセスポートへの ePBR ポリシーの適用はサポートされていますが、この構成でのダウングレードは推奨されません。

## アップグレードパス

すべてのリリースの ISSU 互換性とアップグレードパスに関する詳細については、『[Cisco NX-OS ISSU サポートマトリックス](#)』を参照してください。

## アップグレードパッチ手順

Cisco Nexus 9500 シリーズスイッチでのみ、Cisco NX-OS リリース 7.0(3)I1(2), 7.0(3)I1(3)、または 7.0(3)I1(3a) から他の Cisco NX-OS OS リリースへのソフトウェアアップグレードでは、**install all** コマンドを使用してアップグレードする前に 2 つのパッチをインストールする必要があります。これらのパッチは、それぞれのリリースで使用でき、以下のリンクを使用してダウンロードできます。



**注意** この手順に従わないと、アップグレード後にスイッチを回復するためにコンソールアクセスが必要になる場合があります。



(注) これらのパッチはアップグレード専用です。アップグレード後、パッチは自動的に削除されません。パッチのインストール後にアップグレードしない場合は、非アクティブにしないでください。パッチを非アクティブ化すると、**bios\_daemon** がクラッシュする可能性があります。

[Cisco NX-OS リリース 7.0\(3\)I1\(2\) アップグレードパッチ](#)

[Cisco NX-OS リリース 7.0\(3\)I1\(3\) アップグレードパッチ](#)

[Cisco NX-OS リリース 7.0\(3\)I1\(3a\) アップグレードパッチ](#)

`install all` コマンドを使用してアップグレードする前にこれらのパッチをインストールするには、次の手順に従います。次に、NX-OS ソフトウェアパッチと 7.0(3)I1(2) から 7.0(3)I7(1) へのアップグレードの例を示します。

1. **install add bootflash: {patch-file.bin}** コマンドを使用して、両方のパッチを追加します。

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

2. **install activate {patch-file.bin}** コマンドを使用して、両方のパッチをアクティブにします。

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

3. **install commit {patch-file.bin}** コマンドを使用して、両方のパッチをコミットします。

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

4. **install all** コマンドを使用して、目的のターゲット リリースに NX-OS ソフトウェアをアップグレードします。

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

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

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

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

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

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

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

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

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

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

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

```

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

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

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

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

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

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

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

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

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

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

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

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

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

```

Compatibility check is done:

Module	bootable	Impact	Install-type	Reason
1	yes	disruptive	reset	Incompatible image
6	yes	disruptive	reset	Incompatible image
8	yes	disruptive	reset	Incompatible image
9	yes	disruptive	reset	Incompatible image
10	yes	disruptive	reset	Incompatible image
11	yes	disruptive	reset	Incompatible image
14	yes	disruptive	reset	Incompatible image
15	yes	disruptive	reset	Incompatible image
16	yes	disruptive	reset	Incompatible image
21	yes	disruptive	reset	Incompatible image
22	yes	disruptive	reset	Incompatible image
23	yes	disruptive	reset	Incompatible image
24	yes	disruptive	reset	Incompatible image
25	yes	disruptive	reset	Incompatible image
26	yes	disruptive	reset	Incompatible image
27	yes	disruptive	reset	Incompatible image
28	yes	disruptive	reset	Incompatible image
29	yes	disruptive	reset	Incompatible image
30	yes	disruptive	reset	Incompatible image

Images will be upgraded according to following table:

Module	Image	Running-Version (pri:alt)	New-Version	Upg-Required
--------	-------	---------------------------	-------------	--------------

	1	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)
yes	1	bios	v01.42(00):v01.42(00)	v01.48(00)
yes	6	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)
yes	6	bios	v01.48(00):v01.48(00)	v01.48(00)
no	8	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)
yes	8	bios	v01.48(00):v01.29(00)	v01.48(00)
no	9	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)
yes	9	bios	v01.48(00):v01.35(00)	v01.48(00)
no	10	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)
yes	10	bios	v01.48(00):v01.42(00)	v01.48(00)
no	11	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)
yes	11	bios	v01.48(00):v01.52(00)	v01.48(00)
no	14	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)
yes	14	bios	v01.48(00):v01.48(00)	v01.48(00)
no	15	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)
yes	15	bios	v01.48(00):v01.40(00)	v01.48(00)
no	16	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)
yes	16	bios	v01.48(00):v01.42(00)	v01.48(00)
no	21	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)
yes	21	bios	v01.48(00):v01.42(00)	v01.48(00)
no	22	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)
yes	22	bios	v01.48(00):v01.40(00)	v01.48(00)
no	23	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)
yes	23	bios	v01.48(00):v01.40(00)	v01.48(00)
no	24	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)
yes	24	bios	v01.48(00):v01.40(00)	v01.48(00)
no	25	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)
yes	25	bios	v01.48(00):v01.40(00)	v01.48(00)
no	26	lcn9k	7.0(3)I1(2)	7.0(3)I7(1)
yes	26	bios	v01.48(00):v01.40(00)	v01.48(00)
no	27	nxos	7.0(3)I1(2)	7.0(3)I7(1)
yes				

```

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

```

```

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

```

```

Install is in progress, please wait.

```

```

Performing runtime checks.
[#####] 100% -- SUCCESS

```

```

Syncing image bootflash:/nxos.7.0.3.I7.1.bin to standby.
[#####] 100% -- SUCCESS

```

```

Setting boot variables.
[#####] 100% -- SUCCESS

```

```

Performing configuration copy.
[#####] 100% -- SUCCESS

```

```

Module 1: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#####] 100% -- SUCCESS

```

```

Module 6: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#####] 100% -- SUCCESS

```

```

Module 8: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#####] 100% -- SUCCESS

```

```

Module 9: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#####] 100% -- SUCCESS

```

```

Module 10: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#####] 100% -- SUCCESS

```

```

Module 11: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#####] 100% -- SUCCESS

```

```

Module 14: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#####] 100% -- SUCCESS

```

```

Module 15: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#####] 100% -- SUCCESS

```

```
Module 16: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#####] 100% -- SUCCESS

Module 21: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#####] 100% -- SUCCESS

Module 22: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#####] 100% -- SUCCESS

Module 23: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#####] 100% -- SUCCESS

Module 24: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#####] 100% -- SUCCESS

Module 25: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#####] 100% -- SUCCESS

Module 26: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#####] 100% -- SUCCESS

Module 27: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#####] 100% -- SUCCESS

Module 28: Refreshing compact flash and upgrading bios/loader/bootrom.
Warning: please do not remove or power off the module at this time.
[#####] 100% -- SUCCESS

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

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

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

CISCO SWITCH Ver 8.26

CISCO SWITCH Ver 8.26
Memory Size (Bytes): 0x0000000080000000 + 0x0000000038000000
Relocated to memory
Time: 6/3/2016 4:41:8
Detected CISCO IOFPGA
Booting from Primary Bios
Code Signing Results: 0x0
Using Upgrade FPGA
FPGA Revision      : 0x27
FPGA ID            : 0x1168153
FPGA Date          : 0x20160111
Reset Cause Register: 0x22
```

```
Boot Ctrl Register : 0x60ff
EventLog Register1 : 0x2000000
EventLog Register2 : 0xfbe77fff
Version 2.16.1240. Copyright (C) 2013 American Megatrends, Inc.
Board type 1
IOFPGA @ 0xe8000000
SLOT_ID @ 0x1b
Standalone chassis
check_bootmode: grub: Continue grub
Trying to read config file /boot/grub/menu.lst.local from (hd0,4)
  Filesystem type is ext2fs, partition type 0x83

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

Image Signature verification was Successful.

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

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

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

..done Thu Mar 3 04:42:11 UTC 2016
Creating /dev/mcelog
Starting mcelog daemon
```



```
Overwriting dme stub lib
Replaced dme stub lib
INIT: Entering runlevel: 3
Running S93thirdparty-script...

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

User Access Verification
switchlogin:
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
of Module 1
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
of Module 6
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
of Module 8
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
of Module 9
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
of Module 10
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
of Module 11
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
of Module 14
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
of Module 15
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
of Module 16
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
of Module 21
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
of Module 22
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
of Module 23
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
of Module 24
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
of Module 25
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
of Module 26
2016 Mar  3 04:43:52 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PRESENT: Detected the presence
```

```

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

```

```
(Serial number ABC1211DE34)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED:
All Ejectors closed for module 1. Ejector based shutdown enabled
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 1 detected
(Serial number ABC1217DEFG) Module-Type 32p 40G Ethernet Module Model N9K-X9432PQ
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 1 powered up
(Serial number ABC1217DEFG)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED:
All Ejectors closed for module 9. Ejector based shutdown enabled
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 9 detected
(Serial number ABC1236D4E5) Module-Type 48x1/10G-T 4x40G Ethernet Module Model
N9K-X9564TX
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 9 powered up
(Serial number ABC1236D4E5)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED:
All Ejectors closed for module 10. Ejector based shutdown enabled
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 10 detected
(Serial number ABC1217EFGH) Module-Type 32p 40G Ethernet Module Model N9K-X9432PQ
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 10 powered up
(Serial number ABC1217EFGH)
2016 Mar 3 04:44:01 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED:
All Ejectors closed for module 11. Ejector based shutdown enabled
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 11 detected
(Serial number ABC123DEF4) Module-Type 36p 40G Ethernet Module Model N9K-X9536PQ
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 11 powered up
(Serial number ABC123DEF4)
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED:
All Ejectors closed for module 15. Ejector based shutdown enabled
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 15 detected
(Serial number ABC1212DEFG) Module-Type 36p 40G Ethernet Module Model N9K-X9536PQ
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 15 powered up
(Serial number ABC1212DEFG)
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED:
All Ejectors closed for module 16. Ejector based shutdown enabled
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 16 detected
(Serial number ABCD1235DEFG) Module-Type 48x1/10G SFP+ 4x40G Ethernet Module Model
N9K-X9464PX
2016 Mar 3 04:44:02 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 16 powered up
(Serial number ABCD1235DEFG)
2016 Mar 3 04:44:08 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED:
All Ejectors closed for module 14. Ejector based shutdown enabled
2016 Mar 3 04:44:08 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 14 detected
(Serial number ABC9876DE5F) Module-Type 8p 100G Ethernet Module Model N9K-X9408PC-CFP2
2016 Mar 3 04:44:08 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 14 powered up
(Serial number ABC9876DE5F)
2016 Mar 3 04:44:09 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED:
All Ejectors closed for module 6. Ejector based shutdown enabled
2016 Mar 3 04:44:09 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 6 detected
(Serial number ABC9876DE3F) Module-Type 8p 100G Ethernet Module Model N9K-X9408PC-CFP2
2016 Mar 3 04:44:09 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 6 powered up
(Serial number ABC9876DE3F)
2016 Mar 3 04:44:10 switch%$ VDC-1 %$ %PLATFORM-2-MODULE_EJECTOR_POLICY_ENABLED:
All Ejectors closed for module 8. Ejector based shutdown enabled
2016 Mar 3 04:44:10 switch%$ VDC-1 %$ %PLATFORM-2-MOD_DETECT: Module 8 detected
(Serial number ABC3456D7E8) Module-Type 48x1/10G-T 4x40G Ethernet Module Model
N9K-X9564TX
2016 Mar 3 04:44:10 switch%$ VDC-1 %$ %PLATFORM-2-MOD_PWRUP: Module 8 powered up
(Serial number ABC3456D7E8)
2016 Mar 3 04:44:56 switch%$ VDC-1 %$ %USBHSD-STANDBY-2-MOUNT: logflash: online
2016 Mar 3 04:47:31 switch%$ VDC-1 %$ %ASCII-CFG-2-CONF_CONTROL: System ready
2016 Mar 3 04:47:51 switch%$ VDC-1 %$ %VMAN-2-ACTIVATION_STATE: Successfully activated
virtual service 'guestshell+'
2016 Mar 3 04:47:51 switch%$ VDC-1 %$ %VMAN-2-GUESTSHELL_ENABLED: The guest shell
has been enabled. The command 'guestshell' may be used to access it, 'guestshell
```

```

destroy' to remove it.

User Access Verification

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

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

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

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

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

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

  plugin
  Core Plugin, Ethernet Plugin

  Active Package(s):
  switch#

```

## 拡張 ISSU の設定

拡張 (LXC) ISSU を有効または無効にできます。



- (注) • Cisco NX-OS リリース 10.1(x) への拡張 ISSU はサポートされていません。これは、含まれているカーネル更新を有効化するために基盤となるカーネルのリロードが必要となるためです。システムに次のメッセージが表示されます。

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

実際に、システムは拡張 ISSU の代わりに、非中断 ISSU を実行します。

- Cisco N9K-C9332D-GX2B、N9K-C9348D-GX2A、および N9K-C9364D-GX2A プラットフォームスイッチの場合、拡張 (LXC) ISSU がデフォルトモードであるため、このモードを有効または無効にすることはできません。また、これらのスイッチでは、**show module** コマンドの出力に仮想スーパーバイザ モジュールが表示されます。

### 始める前に

LXC モードを有効にする前に、インストールしたライセンスのライセンス ファイルに 27000 という文字列が含まれていないことを確認します。

### 手順の概要

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

### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>configure terminal</b> 例： switch# configure terminal switch(config#)	グローバル設定モードを開始します。
ステップ 2	<b>[no] boot mode lxc</b> 例： switch(config)# boot mode lxc Using LXC boot mode 例： switch(config)# no boot mode lxc Using normal native boot mode	拡張 (LXC) ISSU を有効または無効にします。  (注) 中断のない拡張 ISSU を実行するには、最初に LXC モードでスイッチを起動する必要があります。
ステップ 3	(任意) <b>show boot mode</b> 例：	拡張 (LXC) ISSU の有効/無効を示します。

	コマンドまたはアクション	目的
	<pre>switch(config)# show boot mode LXC boot mode is enabled</pre> <p>例 :</p> <pre>switch(config)# show boot mode LXC boot mode is disabled</pre>	
ステップ 4	<p><b>copy running-config startup-config</b></p> <p>例 :</p> <pre>switch(config)# copy running-config startup-config</pre>	実行中の設定をスタートアップ コンフィギュレーションに保存します。
ステップ 5	<p><b>reload</b></p> <p>例 :</p> <pre>switch(config)# reload This command will reboot the system. (y/n)? [n] Y loader&gt;</pre>	デバイスがリロードされます。プロンプトが表示されたら、 <b>Y</b> キーを押してリブートを確定します。

#### 次のタスク

「Cisco NX-OS ソフトウェアのアップグレードセクション」の手順に従います。標準または拡張 ISSU を実行する場合は、必ず **non-disruptive** オプションを選択してください。

## Cisco NX-OS ソフトウェアのアップグレード

Cisco NX-OS 10.2(x) リリースにアップグレードするには、次の手順を使用します。



- (注) Cisco NX-OS リリース 10.1(1) 以降では、Cisco Nexus -GX シリーズプラットフォームは 64 ビットの Cisco NX-OS イメージファイルを使用します。このイメージファイルの名前は「nxos64」で始まります (例 : nxos64.10.1.1.bin)。ソフトウェア拡張性をサポートする 64 ビットソフトウェア イメージは、Cisco Nexus C9316D-GX、C93600CD-GX、C9364C-GX スイッチで使用できます。非 GX シリーズプラットフォームは、32 ビットの Cisco NX-OS イメージファイルを使用します。このファイルのイメージファイル名は「nxos」で始まります (例 : nxos.10.1.1.bin)。



- (注) -R ラインカードを装備した Cisco Nexus 9500 プラットフォーム スイッチでは、設定を保存してデバイスをリロードし、Cisco NX-OS リリース 7.0(3)F3(5) から 10.1(1) にアップグレードする必要があります。Cisco NX-OS リリース 9.2(2) またはそれ以降にアップグレードするには、**install all** コマンドを使用することを推奨します。



- (注) アップグレード中にエラーメッセージが表示された場合、アップグレードは失敗し、その理由が表示されます。考えられる原因と解決法については、『Cisco Nexus 9000 シリーズ NX-OS トラブルシューティングガイド、リリース 10.1(x)』を参照してください。

### 始める前に

Cisco NX-OS リリース 10.1(1) に対して中断のない ISSU を実行する前に、Cisco Nexus 3132Q-V プラットフォーム スイッチの BGP グレースフル リスタート タイマーを 180 秒に設定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。『Cisco Nexus 9000 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. 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. アクティブ 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作ります。
5. 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. スタンバイ 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作ります。
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. ソフトウェア イメージの MD5、SHA256、または SHA512 チェックサムを確認することにより、アップグレードを実行する前に、不完全または破損した NX-OS ソフトウェア イメージを検出できます。ソフトウェア イメージの MD5 チェックサムを確認するには、**show file bootflash: <IMAGE-NAME>md5sum** コマンドを実行し、結果の値を [シスコのソフトウェア ダウンロード Web サイト](#) で公開されているソフトウェア イメージの MD5 チェックサムと比較します。ソフトウェア イメージの SHA512 チェックサムを確認するには、次のコマンドを実行します。 **show file bootflash: <IMAGE-NAME>sha512sum** コマンドを実行し、結果の値を [シスコのソフトウェア ダウンロード Web サイト](#) で公開されているソフトウェア イメージの SHA512 チェックサムと比較します。
10. ソフトウェア イメージの MD5、SHA256、または SHA512 チェックサムを確認することにより、アップグレードを実行する前に、不完全または破損した NX-OS ソフトウェア イメージを検出できます。ソフトウェア イメージの MD5 チェックサムを確認するには、**show file bootflash: <IMAGE-NAME>md5sum** コマンドを実行し、結果の値を [シスコのソフトウェア ダウンロード Web サイト](#) で公開されているソフトウェア イメージの MD5 チェックサムと比較します。ソフトウェア イメージの SHA512 チェックサムを確認するには、次のコマンドを実行します。 **show file bootflash: <IMAGE-NAME>sha512sum** コマ

ンドを実行し、結果の値をシスコのソフトウェアダウンロード Web サイトで公開されているソフトウェアイメージの SHA512 チェックサムと比較します。

11. 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。
12. 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。
13. 必要に応じて、`install all nxos`を使用してEPLDイメージをアップグレードします。  
<nxos-image> epld<epld-image>コマンド。
14. `install all nxos bootflash:filename [no-reload | non-disruptive | non-interruptive | serial]` コマンドを使用して Cisco NX-OS ソフトウェアをアップグレードします。
15. (任意) アップグレードプロセス全体を表示します。
16. (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。
17. (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。『Cisco NX-OS ライセンシングガイド』を参照してください。

## 手順の詳細

**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。『Cisco Nexus 9000 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** 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:
16384   Oct 30 17:05:32 2020  lost+found/
1964291584   Dec 08 19:44:33 2020  nxos.10.1.1.bin
...
Usage for bootflash://sup-local
 4825743360 bytes used
16312102912 bytes free
21137846272 bytes total
```

(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** アクティブ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作ります。

```
switch# delete bootflash:nxos.9.2.1.bin
```

**ステップ 5** 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。

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



```
4825743360 bytes used
16312102912 bytes free
21137846272 bytes total
```

- ステップ 6** スタンバイ 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作ります。

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

- 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/nxos64.10.2.1.F.bin
bootflash:nxos64.10.2.1.F.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/nxos64.10.2.1.F.bin
bootflash:nxos64.10.2.1.F.bin compact vrf management use-kstack
```

```
user1@10.65.42.196's password:
nxos64.10.2.1.F.bin 100% 1887MB 6.6MB/s 04:47
Copy complete, now saving to disk (please wait)...
Copy complete.
```

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、**compact** キーワードは NX-OS 画像を圧縮します。

- (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。

- ステップ 9** ソフトウェア イメージの MD5、SHA256、または SHA512 チェックサムを確認することにより、アップグレードを実行する前に、不完全または破損した NX-OS ソフトウェア イメージを検出できます。ソフトウェア イメージの MD5 チェックサムを確認するには、**show file bootflash: <IMAGE-NAME>md5sum** コマンドを実行し、結果の値を [シスコのソフトウェア ダウンロード Web サイト](#) で公開されているソフトウェア イメージの MD5 チェックサムと比較します。ソフトウェア イメージの SHA512 チェックサムを確認するには、次のコマンドを実行します。**show file bootflash: <IMAGE-NAME>sha512sum** コマンドを実行し、結果の値を [シスコのソフトウェア ダウンロード Web サイト](#) で公開されているソフトウェア イメージの SHA512 チェックサムと比較します。

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

```
switch# show file bootflash:nxos.10.1.1.bin sha512sum
7f25coe57ca137a79211fb3835338aae64acf9b021b75oec5d4156e873b4274ca4f98e9a74fe4c8961f5ace99ed65f3826650599369f84ab07265d7c5d61b57f
```

- ステップ 10** ソフトウェア イメージの MD5、SHA256、または SHA512 チェックサムを確認することにより、アップグレードを実行する前に、不完全または破損した NX-OS ソフトウェア イメージを検出できます。ソフトウェア イメージの MD5 チェックサムを確認するには、**show file bootflash: <IMAGE-NAME>md5sum** コマンドを実行し、結果の値をシスコのソフトウェアダウンロード Web サイトで公開されているソフトウェア イメージの MD5 チェックサムと比較します。ソフトウェア イメージの SHA512 チェックサムを確認するには、次のコマンドを実行します。**show file bootflash: <IMAGE-NAME>sha512sum** コマンドを実行し、結果の値をシスコのソフトウェアダウンロード Web サイトで公開されているソフトウェア イメージの SHA512 チェックサムと比較します。

```
switch# show file bootflash:nxos64.10.2.1.F.bin md5sum
c49660952215822afd30bb7958a0765a

switch# show file bootflash:nxos64.10.2.1.F.bin sha256sum
2a64efbb381fabbb52054af74cf3efda1691772a49a70ddd35550431cadecf8e

switch# show file bootflash:nxos64.10.2.1.F.bin sha512sum
3bf6a771aa4a192a8e1383e348b26db483356a9774d74ba39edbf7718248483b3391942d8103de8104deea8fda212266e70bd736220cff34943bd8e359432975
```

- ステップ 11** 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:nxos64.10.2.1.F.bin
```

互換性チェック中、[理由 (Reason)] フィールドに次の ISSU 関連のメッセージが表示されることがあります。

Reason フィールドのメッセージ	説明
ISSU の非互換イメージ	アップグレードしようとする Cisco NX-OS イメージは ISSU をサポートしていません。
デフォルト アップグレードはヒットレスではありません	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。ISSU を実行するには、 <b>non-disruptive</b> オプションを設定する必要があります。

- ステップ 12** 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 13** 必要に応じて、install all nxosを使用してEPLDイメージをアップグレードします。<nxos-image> epld<epld-image>コマンド。

次に、install all nxosの出力例を示します。<nxos-image> epld<epld-image> command :

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

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

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

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

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

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

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

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

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

switch# install all nxos nxos.10.1.1.IJD9.0.59.bin epld n9000-epld.10.2.1.F.img

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

Verifying image bootflash:/nxos.10.1.1.IJD9.0.59.bin for boot variable "nxos".
[#####] 100% -- SUCCESS

Verifying EPLD image bootflash:/ n9000-epld.10.2.1.F.img.
[#####] 100% -- SUCCESS

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

Preparing "nxos" version info using image bootflash:/nxos.10.1.1.IJD9.0.59.bin.
[#####] 100% -- SUCCESS

Preparing "bios" version info using image bootflash:/nxos.10.1.1.IJD9.0.59.bin.
[#####] 100% -- SUCCESS

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

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

**ステップ 14** `install all nxos bootflash:filename [no-reload | non-disruptive | non-interruptive | serial]` コマンドを使用して Cisco NX-OS ソフトウェアをアップグレードします。

```
switch# install all nxos bootflash:nxos64.10.2.1.F.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) `with` オプションを使用すると、デバイスを再配置する前に保存された設定を使用できません。 **install all no-reload** この状態で設定を保存すると、新しいバージョンの NX-OS でデバイスをリロードすると、誤った起動設定になる可能性があります。
- **non-disruptive** : In-Service Software Upgrade (ISSU) を実行してデータ トラフィックの中断を防ぎます。(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)。

- **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。
- **serial** : Cisco Nexus 9500 シリーズ スイッチの I/O モジュールを 1 つずつアップグレードします (デフォルトでは、I/O モジュールは並行してアップグレードされ、全体的なアップグレード時間が短縮されます。具体的な手順としては、ラインカードとファブリック モジュールの前半、ラインカードとファブリック モジュールの後半、最初のシステム コントローラ、2 番目のシステム コントローラという順序で I/O モジュールが並行してアップグレードされます)。

(注) ファイル名を指定しないで **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 NX-OS 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、実行中の画像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 15 (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 16 (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 17 (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。『Cisco NX-OS ライセンシング ガイド』を参照してください。

## vPC のアップグレード プロセス

### プライマリ スイッチでの vPC トポロジのアップグレード プロセス

次のリストは、プライマリまたはオペレーショナル プライマリ vPC ロールのいずれかを有する vPC トポロジのスイッチ上のアップグレード プロセスを要約したものです。vPC 以外のトポロジでのスイッチのアップグレードと異なるステップは太字で示しています。



(注) vPC トポロジでは、2 台のピア スイッチを個別にアップグレードする必要があります。一方のピア スイッチでアップグレードを実行しても、vPC ピア スイッチは自動的に更新されません。

1. **vPC** プライマリ スイッチで **install all** コマンドを発行すると、インストールのアップグレードが開始されます。
2. 互換性チェックにより、アップグレードの影響が表示されます。
3. アップグレードの影響に基づき、インストールが続行するか、停止します。

4. 両方の vPC ピア スイッチで設定がロックされます。
5. 現在の状態が保存されます。
6. システムが新しいイメージをアンロードして実行します。
7. システム ソフトウェアとアプリケーションの再起動が正常に行われます。
8. 新しいイメージでインストーラが再開されます。
9. これでインストールが完了しました。

インストールが完了すると、vPC プライマリ スイッチがアップグレードされます。



- (注) vPC プライマリ スイッチがアップグレードされたバージョンを実行し、vPC セカンダリ スイッチが元のバージョンのソフトウェアを実行します。

## セカンダリ スイッチでの vPC トポロジのアップグレード プロセス

次のリストは、セカンダリまたはオペレーショナルセカンダリ vPC ロールのいずれかを有する vPC トポロジのスイッチ上のアップグレードプロセスを要約したものです。vPC 以外のトポロジでのスイッチのアップグレードと異なるステップは太字で示しています。

1. **vPC セカンダリ スイッチで `install all` コマンドを発行すると、インストールのアップグレードが開始されます。**
2. 互換性チェックにより、アップグレードの影響が表示されます。
3. アップグレードの影響に基づき、インストールが続行するか、停止します。
4. 現在の状態が保存されます。
5. システムが新しいイメージをアンロードして実行します。
6. システム ソフトウェアとアプリケーションの再起動が正常に行われます。
7. 新しいイメージでインストーラが再開されます。
8. **プライマリ スイッチとセカンダリ スイッチでの設定のロックが解除されます。**
9. これでインストールが完了しました。

■ サイトをマルチサイトに追加する。

## サイトをマルチサイトに追加する。



(注) ダウングレード中にエラーメッセージが表示された場合、ダウングレードは失敗し、その理由が表示されます。考えられる原因と解決法については、『Cisco Nexus 9000 シリーズ NX-OS トラブルシューティング ガイド、リリース 10.2(x)』を参照してください。

### 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。『Cisco Nexus 9000 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

**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。『Cisco Nexus 9000 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 サーバにダウンロードします。  
<http://software.cisco.com/download/navigator.html>

(注) アクティブまたはスタンバイ スーパーバイザ モジュール bootflash: にさらに多くのスペースが必要な場合、**delete** を使用します コマンドを使用して、不要なファイルを削除します。

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

(注) **install all** を入力する場合、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
```

## Cisco NX-OS のアップグレード履歴

Cisco Nexus 9000 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 9000 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

**show upgrade history** コマンドを入力して、Cisco NX-OS のアップグレード履歴を表示します。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。**show upgrade history** コマンドの出力例を次に示します。

```
switch# show upgrade history
TYPE                VERSION  DATE                STATUS
NXOS EPLD           n9000-   26 Apr 2020 11:37:16  EPLD Upgrade completed
                   epld.9.3.4.img
NXOS EPLD           n9000-   26 Apr 2020 11:32:41  EPLD Upgrade started
                   epld.9.3.4.img
NXOS system image   9.3(5)   24 Mar 2020 20:09:10  Installation End
NXOS system image   9.3(5)   24 Mar 2020 20:05:29  Installation started
NXOS SMU            9.3(5)   03 Mar 2020 23:34:15  Patch activation ended for
                   nxos.libnbproxyccli_patch-n9k_
                   ALL-1.0.0-9.3.5.lib32_n9000.rpm
NXOS SMU            9.3(5)   03 Mar 2020 23:34:03  Patch activation started for
                   nxos.libnbproxyccli_patch-n9k_
                   ALL-1.0.0-9.3.5.lib32_n9000.rpm
```

Cisco NX-OS リリース 10.2(3)F 以降、Cisco Nexus 9000 スイッチは、ログインの詳細（ユーザー名/セッション ID）を表示する新しい cli **"show upgrade history details"** をサポートします。

**show upgrade history details** コマンドを入力して、Cisco NX-OS アップグレード履歴を表示します。出力には、アップグレード履歴とともに、スイッチの [ログイン (LOGIN)] 列の下にユーザーログインの詳細（ユーザー名/セッション ID）が表示されます。**show upgrade history details** コマンドの出力例を次に示します。

```
switch# sh upgrade history details
TYPE                VERSION  DATE                LOGIN
                   STATUS
NXOS system image   10.2(3)  21 Jan 2022 10:01:06  admin/10.30.216.212
                   Installation End
NXOS system image   10.2(3)  21 Jan 2022 10:00:53  admin/10.30.216.212
                   Installation started
```



NXOS system image 10.2(3)  
Installation End

21 Jan 2022 01:03:52

admin/10.30.216.212





## 第 3 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 \(57 ページ\)](#)
- [モジュラ パッケージの使用 \(59 ページ\)](#)
- [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 \(60 ページ\)](#)
- [RPM に関する情報 \(60 ページ\)](#)
- [YUM コマンドに関する情報 \(75 ページ\)](#)
- [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 \(94 ページ\)](#)
- [インストール操作ユーザ ロールの作成 \(98 ページ\)](#)
- [Compacting Cisco NX-OS ソフトウェア イメージの圧縮 \(98 ページ\)](#)

## 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。

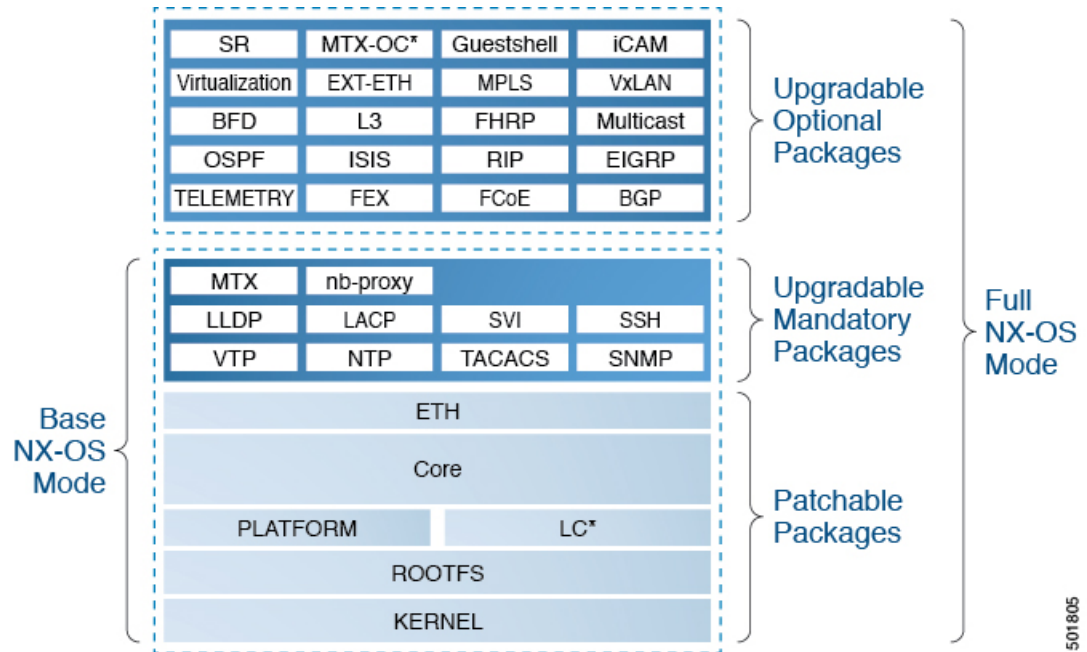
モジュール式の Cisco NX-OS ソフトウェアを使用する利点は次のとおりです。

- 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。

# 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のみ使用する必要があります。

## RPM と YUM

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。

YUM (Yellowdog Updater, Modified) 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。YUM 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。YUM は、システムにインストールされたパッケージを把握するだけでなく、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。

- VSH プロンプトで **install reset nxos base** オプションを使用して、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。
- VSH プロンプトで **install reset nxos full** オプションを使用して、NX-OS イメージをフル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、NX-OS の `install` コマンドを使用して、または YUM 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。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。

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)
- **version** (<X.y.x.b> 形式) : <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	説明
例 : <code>fex-2.0.0.0-9.2.1.lib32_n9000.rpm</code>	
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。
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、`yum` または `install CLI` コマンドを使用して、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。

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
BGP	feature bgp
BFD	feature bfd
Container-tracker	feature container-tracker
EIGRP	feature eigrp
Ext-Eth	<ul style="list-style-type: none"> <li>• feature openflow</li> <li>• feature evb</li> <li>• feature imp</li> <li>• feature netflow</li> <li>• feature sla_sender</li> <li>• feature sla_responder</li> <li>• feature sla twamp-server</li> <li>• feature sflow</li> </ul>
FCoE	<ul style="list-style-type: none"> <li>• feature-set fcoe</li> <li>• feature-set fcoe-npv</li> </ul>
FEX	feature-set fex
FHRP	<ul style="list-style-type: none"> <li>• feature hsrp</li> <li>• feature vrrpv3</li> </ul>
iCAM	feature icam
ISIS	feature isis
MPLS	<ul style="list-style-type: none"> <li>• feature mpls segment-routing</li> <li>• feature mpls evpn</li> </ul>
マルチキャスト	<ul style="list-style-type: none"> <li>• feature pim</li> <li>• feature pim6</li> <li>• feature msdp</li> <li>• feature ngmvpn</li> </ul>
OSPF	<ul style="list-style-type: none"> <li>• feature ospf</li> <li>• feature ospfv3</li> </ul>



パッケージ名	関連機能
RIP	feature rip
サービス	feature catena
SR	feature mpls segment-routing traffic-engineering
TELEMETRY	feature telemetry
仮想化	該当なし
VXLAN	<ul style="list-style-type: none"> <li>feature nv overlay</li> <li>feature fabric forwarding</li> </ul>

## 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 YUM コマンドを使用してください。

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	<p>RPM の保持に使用されます。ユーザが NX-OS 機能 RPM を <b>install add</b> 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、リロード時に常駐します。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。</p> <p>このリポジトリに RPM を追加するには、<b>install add</b> コマンドを使用します。</p> <p>このリポジトリから RPM を削除するには、<b>install remove</b> コマンドを使用します。</p> <p>YUM コマンドを使用しても、リポジトリに追加できます。</p> <p>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。</p>
patching	/bootflash/.rpmstore/patching/patchrepo	<p>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。</p>
Third_Party	/bootflash/.rpmstore/thirdparty	<p>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。</p>

**groups-repo** と **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。YUM 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を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。

## 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、コマンドを使用して確認できます。**running-config** 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せん。

## 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
<b>install reset</b>	<p>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。</p> <p><b>install reset</b> コマンドも <b>write erase</b> 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。</p> <pre>switch(config)# install reset</pre> <hr/> <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> <hr/> <pre>Do you want to proceed with reset operation? (y/n)? [n]</pre>

CLI	説明
<code>install reset nxos base</code>	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 ( <code>write erase</code> ) 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。
<code>install reset nxos full</code>	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 ( <code>write erase</code> ) 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。
<code>install add &lt;&gt;</code>	それぞれのリポジトリに RPM ファイルを追加して、リポジトリ ( <b>patch/feature/third-party</b> ) を更新します。
<code>install activate &lt;rpm name&gt;</code>	リポジトリに存在する RPM をインストールします。
<code>install commit &lt;rpm name&gt;</code>	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。
<code>install deactivate &lt;rpm name&gt;</code>	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。
<code>install remove &lt;rpm name&gt;</code>	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。
<code>sh install active</code>	ベース <code>rootfs</code> RPM 以外の、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 (機能/パッチ/サードパーティ)。
<code>sh install inactive</code>	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
<b>sh install packages</b>	Rootfs RPM を含む、インストールされているすべての RPM をリストします。
<b>[no] system software allow third-party</b>	<p>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。</p> <p>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。</p> <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> <p>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。</p> <pre>switch(config)# system software allow third-party switch(config)# 2020 Nov 17 04:25:41 switch %\$ VDC-1 %\$ %USER-2-SYSTEM_MSG: &lt;&lt;%PATCH-INSTALLER-2-TPS_FEATURE_ENABLED&gt;&gt; User has enabled TPS installation - patch_installer  switch(config)# install activate pbwMonitor-1.0-1.5.0.x86_64.rpm [#####] 100% Install operation 194 completed successfully at Tue Nov 17 04:25:58 2020</pre> <p>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。</p> <pre>switch(config)# no system software allow third-party switch(config)# 2020 Nov 17 04:27:17 switch %\$ VDC-1 %\$ %USER-2-SYSTEM_MSG: &lt;&lt;%PATCH-INSTALLER-2-TPS_FEATURE_DISABLED&gt;&gt; User has disabled TPS installation - patch_installer</pre>



- (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

	コマンドまたはアクション	目的
ステップ 1	<pre>switch#install add bootflash:&lt;keyfile&gt; gpg-key</pre> <p>例 :</p> <pre>install add bootflash:RPM-GPG-KEY-puppetlabs gpg-key [#####] 100% Install operation 304 completed successfully at Thu Jun 19 16:40:28 2018</pre>	Cisco 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、 <code>/etc/pki/rpm-gpg/arm-Nexus9k-rel.gpg</code> に存在します。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	<pre>switch#install verify package &lt;package-name&gt;</pre>	パッケージを検証します。
ステップ 3	<p>または <code>switch#install verify bootflash:&lt;RPM file&gt;</code></p> <p>例 :</p> <pre>switch# install verify bootflash:vxlan-2.0.0.0-9.2.1.lib32_n9000.rpm  RSA signed switch#</pre>	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

	コマンドまたはアクション	目的
ステップ 1	<p><b>show install packages</b></p> <p>例 :</p> <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	<b>install add &lt;rpm&gt;activate</b> 例 :  <pre>switch# <b>install add bootflash:chef.rpm</b> <b>activate</b> 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

	コマンドまたはアクション	目的
ステップ 1	<b>install add</b> <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	<b>install activate</b> <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。

## 1 ステップの RPM のアップグレード

## 手順

	コマンドまたはアクション	目的
ステップ 1	<p><b>install add &lt;rpm&gt;activate upgrade</b></p> <p>例 :</p> <pre>switch(config)# <b>install add</b> <b>bootflash:bgp-2.0.2.0-9.2.1.lib32_n9000.rpm</b> <b>activate upgrade</b></pre> <p>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</p> <p>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</p>	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	<pre>install add &lt;rpm&gt;activate downgrade</pre> <p>例 :</p> <pre>switch(config)# install add bootflash:bgp-2.0.1.0-9.2.1.lib32_n9000.rpm activate downgrade</pre> <p>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</p> <p>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</p>	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

	コマンドまたはアクション	目的
ステップ 1	<p><code>install deactivate &lt;rpm&gt;</code></p> <p>例 :</p> <pre>switch(config)# install deactivate bgp Base RPM found. Do you want to downgrade to base version(y/n) [n] y Downgrading to the base version [#####] 100% Install operation 190 completed successfully at Tue Nov 17 04:10:40 2020</pre> <p>例 :</p> <pre>switch(config)# install deactivate bgp Base RPM found. Do you want to downgrade to base version(y/n) [n] n</pre> <hr/> <pre>WARNING!! This operation will remove 'bgp-3.0.0.0-9.4.1.lib32_n9000' related configuration from running-configuration on successful completion. Update startup-configuration accordingly.</pre> <hr/> <pre>[#####] 100% Install operation 9 completed successfully at Tue Nov 17 05:05:59 2020</pre>	<p>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。</p> <p>基本バージョンにダウングレードするには、<b>y</b> と入力します。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 <b>n</b> と入力します。</p>

## 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

	コマンドまたはアクション	目的
ステップ 1	<b>install remove &lt;rpm&gt;</b> 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。

## YUM コマンドに関する情報

YUM コマンドの詳細については、以降のセクションを参照してください。



- (注) YUM コマンドは `ctrl+c` をサポートしていません。インストール コマンドは `ctrl+c` をサポートしています。YUM コマンドが `ctrl+c` を使用して中断された場合は、「`/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。

## YUM コマンドを使用したパッケージの操作

YUM 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。



- (注) YUM 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。

## インストール済み RPM のリストをチェックする

```
#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
```

## インストール済み RPM のリストをチェックする

**yum 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# 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
...

```

## インストール済み RPM の詳細を取得する

**yum 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。

### yum info vxlan

```

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

```

```

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

```

```

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

```

## 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# 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

```

```

localdb                | 1.1 kB    00:00 ...
localdb/primary       | 951 B     00:00 ...
localdb                | 886 B     00:00 ...

```

1/1

```

patching
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	/vxlan-2.0.1.0-9.2.1.lib32_n9000	lib32_n9000	2.0.1.0-9.2.1	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 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

```

```

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

```



```

thirdparty          | 951 B      00:00 ...

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

```

Dependencies Resolved

Package 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。

```
yum 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# 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!
```

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 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
```

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

```
Setting up Upgrade Process
```

```
Examining /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm: vxlan-2.0.1.0-9.2.1.lib32_n9000
Marking /bootflash/vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm as an update to
vxlan-2.0.0.0-9.2.1.lib32_n9000
```

```
Resolving Dependencies
```

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

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

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

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

```
Dependencies Resolved
```

Package Version	Arch Size	Repository
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 のアップグレードを示しています。

```
yum 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 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

```

```

localdb                | 1.1 kB    00:00 ...

```

```

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

```

```

localdb                | 1.3 kB    00:00 ...

```

2/2

```
patching
```

```

thirdparty             | 951 B     00:00 ...

```

```

| 951 B     00:00 ...

```

```
Resolving Dependencies
```

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

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

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

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

```
Dependencies Resolved
```

---

Package

Arch

```

                                Version                               Size                               Repository
-----
Downgrading:
vxlan                               lib32_n9000
                                2.0.0.0-9.2.1                               1.6 M                               groups-repo

Transaction Summary
-----
Downgrade      1 Package

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

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

                                                    2/2

Removed:
  vxlan.lib32_n9000 0:2.0.1.0-9.2.1

Installed:
  vxlan.lib32_n9000 0:2.0.0.0-9.2.1

Complete!

```

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。

```
yum 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。

```
yum 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。RPM を削除するには、**yum erase <rpm>** コマンドを使用します。

```
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	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!

## YUM グループのサポート

YUM 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。

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。YUM では、次の `group` コマンドがサポートされています。

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

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

### grouplist 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。yum でパッケージを個別に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して、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、使用可能なグループをすべてリストするには **yum grouplist** コマンドを使用します。

```
bash-4.2# sudo yum grouplist

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

localdb                               | 1.1 kB    00:00 ...
patching                               | 951 B    00:00 ...
thirdparty                             | 951 B    00:00 ...
groups-repo/group                      | 951 B    00:00 ...

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

bash-4.3$
```

### groupmembers コマンドを使用する

あるパッケージグループの説明と内容を表示するには、**yum groupinfo** コマンドを使用します。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-4.2# sudo yum groupinfo l2

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

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

## groupinstall コマンドを使用する

```

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

Group: L2
Mandatory Packages:
  lacp
  lldp
  svi
  vtp

```

## 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-4.2# sudo yum groupinstall routing

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 ...
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	Repository	Version	Size
Installing:				
bgp	lib32_n9000	groups-repo	2.0.0.0-9.2.1	2.4 M
eigrp	lib32_n9000	groups-repo	2.0.0.0-9.2.1	428 k
isis	lib32_n9000		2.0.0.0-9.2.1	



```

rip                lib32_n9000                groups-repo                1.2 M
                                                           groups-repo                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!

```

## groupupdate コマンドを使用する

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するには、**yum groupupdate** コマンドを使用します。

## groupupdate コマンドを使用する

```
bash-4.3# yum groupupdate routing
```

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

```

localdb                | 1.1 kB    00:00 ...
localdb/primary        | 951 B     00:00 ...
localdb                | 1.9 kB    00:00 ...
patching               6/6
thirdparty             | 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	Repository	Size	Version
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!
```

## grouperase コマンドを使用する

グループ、またはグループのすべての RPM メンバを削除するには、**yum grouperase** コマンドを使用します。

```
bash-4.3$ sudo yum grouperase routing

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

localdb | 1.1 kB 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	Repository	Version	Size
Removing:				
bgp	lib32_n9000	@groups-repo	2.0.0.0-9.2.1	11 M
eigrp	lib32_n9000	@groups-repo	2.0.0.0-9.2.1	2.0 M
isis	lib32_n9000	@groups-repo	2.0.0.0-9.2.1	5.7 M
ospf	lib32_n9000	@groups-repo	2.0.0.0-9.2.1	15 M
rip	lib32_n9000	@groups-repo	2.0.0.0-9.2.1	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!

```

## リポジトリを特定する

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# yum 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 ...

```

## インストール済み YUM バージョンを特定する

```

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

```

## インストール済み YUM バージョンを特定する

次に、インストール済み YUM バージョンをリストする例を示します。

```
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

```

## NX-OS CLI と YUM コマンドの対応関係

次の表は、NX-OS CLI とそれに対応する YUM コマンドを示しています。

表 6: パッチ適用コマンドリファレンス

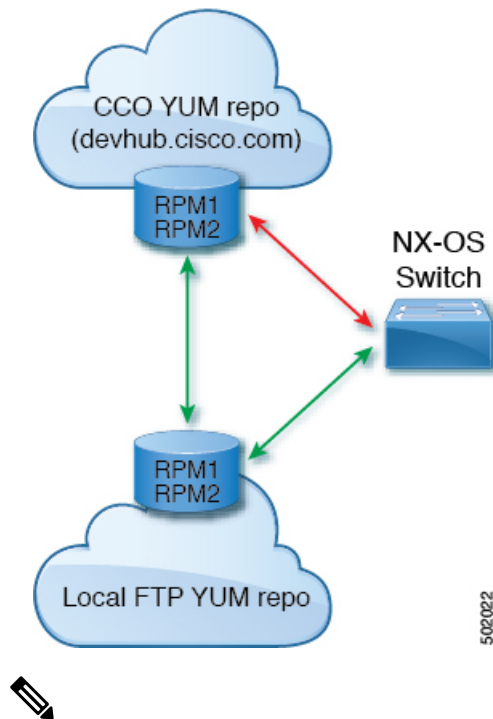
NX-OS CLI コマンド	YUM コマンド
<b>show install inactive</b>	<b>yum list --patch-only available</b>
<b>show install active</b>	<b>yum list --patch-only installed</b>
<b>show install committed</b>	<b>yum list --patch-only committed</b>
<b>show install packages</b>	<b>yum list --patch-only</b>
<b>show install pkg-info</b>	<b>yum info --patch-only</b>

NX-OS CLI コマンド	YUM コマンド
<b>show install log</b>	<b>yum history --show-patch-log</b> ここで log_cmd は次のとおりです。 <ul style="list-style-type: none"> <li>• opid : ある操作 ID に固有なログ</li> <li>• last : 最新の操作のログを示します。</li> <li>• reverse : 逆の順序でログを表示します。</li> <li>• detail : 詳細ログを表示します。</li> <li>• from : 特定の操作 ID 以降のログを示します。</li> </ul>
<b>clear install log</b>	<b>yum history --clear-patch-log=</b> ここで clear_log_cmd は次のとおりです。 <ul style="list-style-type: none"> <li>• all : ログを完全にクリアします。</li> <li>• : この操作 ID 以降のログをクリアします。</li> </ul>
<b>install add</b>	<b>yum install --add bootflash:/</b>
<b>install remove</b>	<b>yum install --remove</b>
<b>install remove inactive</b>	<b>yum install --remove all</b>
<b>install activate</b>	<b>yum install --no-persist --nocommit</b> (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。
<b>install deactivate</b>	<b>yum erase --nocommit</b> (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。
<b>install commit</b>	<b>yum install --commit</b>
<b>Install commit</b>	<b>yum install --commit all</b>

## 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) については、Ciscoリポジトリを参照してください。  
<https://devhub.cisco.com/artifactory/open-nxos/10.1.1/open-nxos>

## 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. `yum 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

	コマンドまたはアクション	目的
ステップ 1	<code>yum install vsftpd</code>	Vsftpd (FTP サーバ) をインストールします。
ステップ 2	<code>systemctl start vsftpd</code>	FTP サーバを開始します。
ステップ 3	<code>systemctl status vsftpd</code>	サーバのステータスを確認します。
ステップ 4	<code>firewall-cmd --zone=public --permanent --add-port=21/tcp</code>	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	<code>firewall-cmd --zone=public --permanent --add-service=ftp</code>	FTP サービスを追加します。
ステップ 6	<code>firewall-cmd --reload</code>	サーバをリロードします。
ステップ 7	<code>wget ftp:// &lt;ip of FTP server&gt; /test.txt</code>	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) <code>/var/ftp/</code> 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. `cat /etc/yum.repos.d/local.repo`
2. `bash 4.3 #yum repolist`
3. `nohup reposync -r <repo-name mentioned in the local.repo> -p <directory path to sync>&`
4. `tail -f nohup.out`

## FTP サーバに到達するようにスイッチを設定する

## 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>cat /etc/yum.repos.d/local.repo</b> 例 : <pre>bash-4.3#cat /etc/yum.repos.d/local.repo  [localrepo] name=localrepo baseurl= https://devhub.cisco.com/artifactory/open-nxos/7.0-3-I2-1/x86_64/ enabled=1 gpgcheck=0 sslverify=0</pre>	/etc/yum.repos.d/ 下にリポジトリ ファイルを作成します。たとえば、 <b>local.repo</b> リポジトリを作成してベース URL を追加します。
ステップ 2	<b>bash 4.3 #yum repolist</b> 例 : <pre>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</pre>	リポジトリの到達可能性を確認します。
ステップ 3	<b>nohup reposync -r &lt;repo-name mentioned in the local.repo&gt; -p &lt;directory path to sync&gt;&amp;</b> 例 : <pre>nohup reposync -r localrepo -p /var/ftp/ &amp;</pre> <p>このコマンドは、<b>/var/ftp/</b> 内部に名前 <b>local.repo</b> でディレクトリを作成し、すべてのパッケージを <b>devhub.cisco.com</b> からこのディレクトリにダウンロードします。</p>	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。
ステップ 4	<b>tail -f nouhup.out</b>	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. **cat /etc/yum/repos.d/ftp.repo**

4. **ip netns exec management bash**
5. **yum repolist**
6. **yum list available**

## 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<b>run bash sudo su</b>	Sudo ユーザとしてログインします。
ステップ 2	<b>ip netns exec management ping &lt;ip_address&gt;</b>	スイッチから <b>ping</b> コマンドを使用して、FTP サーバアドレスの到達可能性を確認します。
ステップ 3	<b>cat /etc/yum/repos.d/ftp.repo</b>  例： <pre>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</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 4	<b>ip netns exec management bash</b>	Bash シェル プロンプトを使用します。
ステップ 5	<b>yum repolist</b>  例： <pre>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</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 6	<b>yum list available</b>	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

**install** コマンドは、**admin** ロールのユーザのみが使用できます。**install** コマンドは、RBAC によりユーザが利用できるようになります。該当する RBAC 設定ガイドラインを参照してください。

## Compacting Cisco NX-OS ソフトウェア イメージの圧縮

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.
```

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、**compact** キーワードは NX-OS 画像を圧縮します。



(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。



## 第 4 章

# Cisco NX-OS から ACI ブートモードへの変換および ACI ブートモードから Cisco NX-OS への再変換

この章では、Cisco NX-OS 9000 シリーズ スイッチを Cisco NX-OS からシスコ アプリケーション セントリック インフラストラクチャ (ACI) のブートモードに変換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。

- [ACI ブートモードへの変換 \(99 ページ\)](#)
- [リプレースメント スタンバイ スーパーバイザから ACI ブートモードへの変換 \(102 ページ\)](#)
- [Cisco NX-OS への再変換 \(103 ページ\)](#)

## ACI ブートモードへの変換

任意の Cisco Nexus 9000 シリーズ スイッチを Cisco NX-OS から ACI ブートモードに変換できます。



- (注) この手順は、Cisco Nexus 9000 シリーズ スイッチを ACI ブートモードに変換する場合に使用してください。それ以前の Cisco NX-OS リリースから ACI ブートモードに変換する場合は、『[Nexus 9000 Standalone \(NXOS\) to ACI Conversion document](#)』の手順に従ってください。

### 始める前に

『[Release Notes for Cisco Nexus 9000 Series ACI-Mode Switches](#)』の「Supported Hardware」のセクションに従って、スイッチのハードウェアが ACI ブートモードでサポートされているかどうかを確認します。たとえば、ラインカードは Cisco NX-OS と ACI ブートモードの間で互換性がありません。

サポートされていないモジュールは取り外すか電源をオフにします (**poweroff module module** コマンドを使用します)。これを行わない場合、サポートされていないモジュールの電源がオ

フになるまでソフトウェアがリカバリ/再試行機能を使用するため、変換プロセスの遅延になる可能性があります。

デュアル スーパーバイザ システムでは、`show module` コマンドを使用して、スタンバイ スーパーバイザ モジュールが `ha-standby` 状態であることを確認します。

Application Policy Infrastructure Controller (APIC) でリリース 1.0(2j) 以降が実行されていることを確認します。

ACI イメージが 11.0(2x) 移行のリリースであることを確認します。

`show install all impact epld epld-image-name` コマンドを使用して、スイッチで EPLD イメージのアップグレードが不要であることを確認します。何らかのアップグレードが必要な場合は、『[Cisco Nexus 9000 Series FPGA/EPLD Upgrade Release Notes](#)』の手順に従ってください。

## 手順の概要

1. スイッチが最新のリリースを実行しているか確認します。
2. 以下の手順に従って、ACI イメージを APIC からコピーします。
3. 以下の手順に従って、ACI イメージを起動します。
4. デバイスの証明書をインストールする必要があるかどうかを確認します。

## 手順の詳細

**ステップ 1** スイッチが最新のリリースを実行しているか確認します。

例：

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

Cisco NX-OS のファイル名は、「`nxos`」で始まります。

**ステップ 2** 以下の手順に従って、ACI イメージを APIC からコピーします。

- a) スイッチの `mgmt0` インターフェイスに IP アドレスを設定し、このインターフェイスと APIC の間で接続できるようにします。
- b) スイッチで SPC サービスを有効化します。

例：

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

- c) APIC CLI から、SCP を使用して、APIC のファームウェア イメージをスイッチ上のアクティブ スーパーバイザ モジュールにコピーします。

例：

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

- d) デュアル スーパーバイザ システムでは、ACI イメージをスタンバイ スーパーバイザ モジュールにコピーします。

例 :

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

**ステップ 3** 以下の手順に従って、ACI イメージを起動します。

- a) Cisco NX-OS から起動しないようにスイッチを設定します。

例 :

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

- b) 設定を保存します。

例 :

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

(注) ACI イメージを起動する前に `copy running-config startup-config` コマンドを実行する必要があります。このコマンドは、`boot aci` コマンドを入力した後は実行しないでください。

- c) ACI イメージでアクティブおよびスタンバイ スーパーバイザ モジュールを起動します。

例 :

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

注意 `copy running-config startup-config` コマンドを `boot aci` コマンドの後に入力しないでください。このコマンドを入力すると、スイッチは `loader>` プロンプトに移動します。

- d) MD5 チェックサムを表示して、ファイルの整合性を確認します。

例 :

```
switch(config)# show file bootflash:aci-image-name md5sum
```

- e) スイッチをリロードします。

例 :

```
switch(config)# reload
```

- f) 管理者としてスイッチにログインします。

例 :

```
Login: admin
```

**ステップ 4** デバイスの証明書をインストールする必要があるかどうかを確認します。

例 :

```
admin@apic1:aci> openssl asn1parse -in /securedata/ssl/server.crt
```

コマンド出力で PRINTABLESTRING を検索します。「シスコが製造元の CA」がリストされている場合は、正しい証明書がインストールされます。他のものがリストされている場合は、デバイスに正しい証明書を生成しインストールする方法について TAC にお問い合わせください。

(注) 2014 年 5 月より前に出荷された Cisco Nexus 9000 シリーズ スイッチでは、証明書のインストールが必要になることがあります。

このコマンドを実行するには、TACにお問い合わせください。

### 次のタスク

スイッチを ACI モードに設定して運用するには、ACI および APIC のドキュメントを参照してください：<http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html>

## リプレースメント スタンバイ スーパーバイザから ACI ブート モードへの変換

デュアルスーパーバイザシステムのスタンバイスーパーバイザモジュールを交換する必要がある場合、リプレースメントスタンバイスーパーバイザで使用するためには、ACI イメージのコピーおよび起動が必要になります。

### 始める前に

ACI イメージを USB ドライブにコピーします。

**ステップ 1** スイッチをリロードします。

例：

```
switch# reload
```

**ステップ 2** 初期のブートシーケンス中にブレイクシーケンス (Ctrl+C キーまたは Ctrl+] キー) を入力して、loader> プロンプトにアクセスします。

例：

```
Ctrl-C  
loader>
```

**ステップ 3** ACI イメージを含む USB ドライブをスタンバイスーパーバイザ USB スロットに接続します。

**ステップ 4** ACI イメージを起動します。

例：



```
loader> boot usb#:aci-image-name
```

(注) USB ドライブが 2 つある場合、**dir** コマンドを入力して、どのドライブに ACI イメージが含まれているか確認してください。次に、**boot** コマンドで **usb1** または **usb2** を指定してください。

**ステップ 5** 管理者としてスイッチにログインします。

```
Login: admin
```

**ステップ 6** ACI イメージを USB ドライブからスイッチにコピーします。

例 :

```
switch# copy usb#:aci-image-name bootflash:aci-image-name
```

## Cisco NX-OS への再変換

Cisco Nexus 9000 シリーズ スイッチを ACI ブート モードから Cisco NX-OS に再変換できます。

### 手順の概要

1. スイッチをリロードします。
2. 初期のブートシーケンス中にブレイク シーケンス (Ctrl+C キーまたは Ctrl+] キー) を入力して、loader> プロンプトにアクセスします。
3. switch(boot)# プロンプトで、ブート プロセスが停止するように設定します。
4. Cisco NX-OS イメージでアクティブ スーパーバイザ モジュールを起動します。
5. スイッチのファイルシステムのパーティション分割をデフォルト設定に戻します。ブートフラッシュ ファイルシステムが Cisco NX-OS パーティションにリセットされ、Cisco NX-OS イメージが削除されます。
6. Cisco NX-OS イメージ ファイルのアップロードを完了します。
7. Cisco NX-OS イメージをブートフラッシュに再コピーして、適切なブート変数を設定し、次回のリロード時にシステムが Cisco NX-OS イメージをリロードするようにします。
8. システム コントローラが立ち上がるまで待機します。約 15 ~ 20 分かかることがあります。
9. アクティブ スーパーバイザ モジュールとシステム コントローラがアクティブ状態であることを確認します。
10. デュアル スーパーバイザ システムでは、スタンバイ スーパーバイザでステップ 3 ~ 6 を実行します。
11. スイッチにログインし、Cisco NX-OS ソフトウェアが稼働していることを確認します。

## 手順の詳細

ステップ1 スイッチをリロードします。

例：

```
switch# reload
```

ステップ2 初期のブートシーケンス中にブレイクシーケンス (Ctrl+C キーまたは Ctrl+J キー) を入力して、loader> プロンプトにアクセスします。

例：

```
Ctrl-C  
loader>
```

ステップ3 switch(boot)# プロンプトで、ブートプロセスが停止するように設定します。

例：

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

ステップ4 Cisco NX-OS イメージでアクティブスーパーバイザモジュールを起動します。

例：

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

(注) ブート変数に記載されている Cisco NX-OS イメージがブートフラッシュに存在しない場合、システムはブートシーケンス中にローダプロンプトにフォールバックします。ローダプロンプトからスイッチを回復するには、ブートフラッシュに存在する別のイメージを使用してシステムを起動するか、**tftboot** を実行するか、または USB デバイスを介してブートします。

(注) 一部の Cisco NX-OS リリースおよび Cisco Nexus 9000 シリーズスイッチでは、次のエラーメッセージが表示されます。

```
!!Fatal error!!  
Can't reserve space for RPM repo  
Please free up bootflash space and reboot
```

このエラーメッセージが表示される場合は、ステップ1からやり直します。ステップ3の後で、**cmdlineinit\_system** コマンドを入力して、ステップ4に進みます。スイッチは通常の Cisco NX-OS プロンプトで起動し、switch(boot)# プロンプトをスキップします。

ステップ5 スイッチのファイルシステムのパーティション分割をデフォルト設定に戻します。ブートフラッシュファイルシステムが Cisco NX-OS パーティションにリセットされ、Cisco NX-OS イメージが削除されます。

例：

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

ステップ6 Cisco NX-OS イメージファイルのアップロードを完了します。

例 :

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

(注) 一部の Cisco Nexus 9000 シリーズ スイッチでは、デバイスは通常の Cisco NX-OS プロンプト (switch#) でロードされず、代わりに「bash-4.2#」として起動します。この場合は、デバイスの電源を再投入し、ローダに移動し、TFTPまたはUSBのいずれかの方法を使用してNX-OS イメージを起動する必要があります。

- TFTP 方式の場合は、最初に **set ip** IP アドレスのサブネットマスクと **set gw** ゲートウェイ アドレス コマンドを使用して、デバイスに IP アドレスとゲートウェイを割り当てます。これは、上記の手順で **init system** コマンドを実行すると、デバイス上で使用可能なすべての設定が消去されるため必要です。

例

```
loader> set ip 1.1.1.2 255.255.255.255.0  
loader>set gw 1.1.1.1
```

その後、**tftp** コマンドを使用して、イメージをロードします。

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

- USB 方式の場合、スイッチに USB をマウントし、ローダ上で **dir** コマンドを実行して、ブートフラッシュフォルダと USB デバイスの内容を確認します。

例

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

その後、次のコマンドを使用して、NX-OS イメージを起動します。

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

Cisco NX-OS イメージを起動すると、デバイスは Cisco NX-OS スイッチとしてロードされ、残りの手順を続行できます。

**ステップ 7** Cisco NX-OS イメージをブートフラッシュに再コピーして、適切なブート変数を設定し、次回のリロード時にシステムが Cisco NX-OS イメージをリロードするようにします。

例 :

TFTP の例 :

```
switch# copy tftp://tftp-server-ip/nxos-image-name bootflash:  
switch# configure terminal  
switch(config)# boot nxos bootflash:nxos-image-name  
switch(config)# copy running-config startup-config  
switch(config)# end
```

USB の例 :

```
switch# copy usb1:nxos-image-name bootflash:  
switch# configure terminal  
switch(config)# boot nxos bootflash:nxos-image-name
```

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

**ステップ 8** システム コントローラが立ち上がるまで待機します。約 15 ～ 20 分かかることがあります。

ACI と Cisco NX-OS でファイル システムが異なる場合、ACI から Cisco NX-OS への変換中に、再フォーマットの変更を一度だけ実施する必要があります。Cisco NX-OS イメージによる後続のリロードは高速になります。

**ステップ 9** アクティブスーパーバイザモジュールとシステムコントローラがアクティブ状態であることを確認します。

例：

```
switch# show module
```

**ステップ 10** デュアルスーパーバイザシステムでは、スタンバイスーパーバイザでステップ 3 ～ 6 を実行します。

**ステップ 11** スイッチにログインし、Cisco NX-OS ソフトウェアが稼働していることを確認します。

## ACI シェルで SCP を使用したブートフラッシュへの NX-OS イメージのロード

ACI モードのスイッチがあり、それを NX-OS モードに変換する必要があるが、TFTP ブートを実行できず、USB オプションが使用できない場合に、このタスクを使用します。次の手順では、ACI モードでスイッチを起動し、管理ポートを設定し、ソフトウェア画像をブートフラッシュパーティションにコピーする方法について説明します。

リーフスイッチは、ファブリック検出状態の ACI モードで起動します。

**ステップ 1** ユーザ名は `admin` で、パスワードなしでログインします。次のコマンドプロンプトが表示されます。

```
#
```

**ステップ 2** `configure terminal`

例：

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

**ステップ 3** `interface mgmt 0`

例：

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

**ステップ 4** `ip address ipv4-address { [/length ] | [ subnet-mask ] }`

例：

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

**ステップ 5 no shutdown**

例 :

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

**ステップ 6 exit**

例 :

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

**ステップ 7 vrf context management**

例 :

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

**ステップ 8 ip route ipv4-address { [ /length ] | [ subnet-mask ] } default-gw-ipv4-address { [ /length ] | [ subnet-mask ] }**

例 :

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

**ステップ 9 end**

例 :

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

**ステップ 10 cd /bootflash**

例 :

```
# cd /bootflash
#
```

**ステップ 11 scp username @ scp-server-ip-address : nxos-image**

例 :

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

**ステップ 12** スイッチをリロードし、ローダー プロンプトに戻り、以前の「Cisco NX-OSへの変換」の手順に従って NX-OS 画像をロードします。新しくコピーされたソフトウェア画像がブートフラッシュに表示されます。

**例**

```
# configure terminal
(config)# interface mgmt 0
(config-if)# ip address 10.1.1.20/24
(config-if)# no shutdown
(config-if)# exit
(config)# vrf context management
```

## ACI シェルで SCP を使用したブートフラッシュへの NX-OS イメージのロード

```
(config-vrf)# ip route 0.0.0.0/0 10.1.1.30/24
(config-vrf)#end
# cd /bootflash
# scp user1@10.1.1.25:n9000-dk9.7.0.3.I1.1.bin
```



## 第 5 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 \(109 ページ\)](#)
- [Nexus 9000-R シリーズスイッチのvPC アップグレードおよびダウングレード手順 \(109 ページ\)](#)

## vPC フォークリフト アップグレード

vPC トポロジでは、Cisco Nexus 9000 シリーズスイッチのペアから Cisco Nexus 9000 シリーズスイッチの異なるペアに移行できます。たとえば、Cisco Nexus 9508 vPC ピア ノードのペアから、Cisco Nexus 9516 スイッチのペアに移行できます。詳細については、『[Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide](#)』の「vPC Forklift Upgrade Scenario」のセクションを参照してください。

## Nexus 9000-R シリーズスイッチのvPC アップグレードおよびダウングレード手順

vPC トポロジでは、通常 2 台のピアスイッチを個別にアップグレードする必要があります。一方のピアスイッチでアップグレードを実行しても、vPC ピアスイッチは自動的に更新されません。

ただし、Cisco NX-OS リリース 7.0(3)F3(3c) および 7.0(3)F3(4) は、vPC ピアスイッチの Cisco NX-OS リリース 9.2(x) とは互換性がありません。両方の vPC ピアを同時に Cisco NX-OS リリース 9.2(x) にアップグレードして、1 つのスイッチが 7.0(3)F3(x) リリースを実行し、もう一方のスイッチが 9.2(x) を実行している状況を避ける必要があります。スイッチを Cisco NX-OS リリース 7.0(3)F3(4) からアップグレードする場合は、次の手順を使用してアップグレード中のトラフィックの影響を最小限に抑えることも可能です。



(注) この手順は、BroadcomまたはCloudscaleベースのスイッチでは使用できません。

1. スイッチ A と B は、Cisco NX-OS リリースを実行しています。スイッチ A は、プライマリスイッチで、スイッチ B はセカンダリスイッチです。両方のスイッチで **copy r s** コマンドを使用します。

```
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. プライマリスイッチでピアリンク (PL) を停止します。セカンダリスイッチは、そのvPC レッグを停止します。

```
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. セカンダリ スイッチ上の vPC ドメインで、vPC 自動リカバリを設定します。 **vpc upgrade** を有効化します (**exec** コマンド)。

```

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

```

- セカンダリ スイッチでレイヤ 3 のルートが学習された後に、新しいリリース イメージでプライマリ スイッチをリロードします。セカンダリ スイッチがプライマリの役割を引き継ぎ、約 5 秒後にその vPC レッグを稼働します。

```

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. プライマリ スイッチが復帰したら、そのピア リンクが稼働します。

```
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
```

ダウングレードの場合は、両方のスイッチを同時にリロードします。





## 索引

### B

boot aci bootflash [101](#)

### C

copp profile [14](#)

copy [47, 53, 103](#)

copy bootflash: [101](#)

### D

delete bootflash [46, 53](#)

dir bootflash [46, 52](#)

### F

features scp-server [100](#)

### G

guestshell destroy [31](#)

### I

init system [104](#)

install all nxos bootflash [49, 53](#)

### L

load-nxos [105](#)

### N

no boot nxos [101](#)

### P

ping [7](#)

poweroff module [53, 99](#)

### R

reload [101-102](#)

### S

show configuration session summary [7](#)

show incompatibility-all nxos bootflash [53](#)

show install all impact nxos bootflash [48, 53](#)

show install all status [50, 53](#)

show version [50, 54, 100](#)

show file bootflash [48, 101](#)

show incompatibility nxos bootflash: [7](#)

show install all impact epld [100](#)

show module [100, 106](#)

### せ

セットアップ [14](#)

### ふ

ブート [103](#)



## 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。