



Cisco Nexus 9000 シリーズ NX-OS リリース 9.2(x) ソフトウェア アップグレード/ダウングレードガイド

初版：2018年7月18日

最終更新：2019年1月29日

シスコシステムズ合同会社

〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー

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

お問い合わせ先：シスコ コンタクトセンター

0120-092-255（フリーコール、携帯・PHS含む）

電話受付時間：平日 10:00～12:00、13:00～17:00

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

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

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

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

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

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

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

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

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

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

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. 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)

© 2018–2019 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

新機能および変更された機能に関する情報 1

新機能および変更された機能に関する情報 1

第 2 章

Cisco Nexus 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 4

Cisco NX-OS ソフトウェア アップグレードの前提条件 8

Cisco NX-OS ソフトウェア ダウングレードの前提条件 9

Cisco NX-OS ソフトウェア アップグレードの注意事項 9

Cisco NX-OS ソフトウェア ダウングレードの注意事項 15

拡張 ISSU の設定 16

Cisco NX-OS ソフトウェアのアップグレード 17

以前のソフトウェア リリースへのダウングレード 20

第 3 章

Cisco NX-OS ソフトウェアのオプション 23

Cisco NX-OS ソフトウェアのオプション 23

モジュラ パッケージの使用 25

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	26
RPM に関する情報	26
RPM の形式	26
オプション RPM とその関連機能	27
NX-OS 機能 RPM インストールに関するガイドライン	29
機能 RPM の操作に install CLI を使用する	31
デジタル署名のサポートにインストール CLI を使用する	32
インストールされているすべての RPM のクエリ	33
1 ステップ手順による RPM のインストール	34
2 ステップ手順による RPM のインストール	35
1 ステップの RPM のアップグレード	36
RPM のダウングレード	37
RPM の削除	38
YUM コマンドに関する情報	39
YUM コマンドを使用したパッケージの操作	39
イメージのベース バージョン RPM を特定する	39
インストール済み RPM のリストをチェックする	40
インストール済み RPM の詳細を取得する	41
RPM のインストール	41
RPM のアップグレード	43
RPM のダウングレード	46
RPM の削除	47
YUM グループのサポート	48
リポジトリを特定する	55
インストール済み YUM バージョンを特定する	56
NX-OS CLI と YUM コマンドの対応関係	56
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	58
Red Hat Enterprise Linux 7 (RHEL7) 仮想マシン上に FTP サーバを作成する	58
ローカル FTP YUM リポジトリを作成する	59
FTP サーバに到達するようにスイッチを設定する	60
インストール操作ユーザ ロールの作成	61

第 4 章	高速リロードを使用した Cisco NX-OS ソフトウェアのアップグレード	63
	高速リロードについて	63
	イベントの高速リロード シーケンス	64
	高速リロードの前提条件	64
	高速リロードに関する注意事項と制約事項	64
	高速リロードの実行と Cisco NX-OS ソフトウェアのアップグレード	66
	高速リロードによる設定の保存	67
	その他の参考資料	68
	関連資料	68

第 5 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	69
	ACI ブート モードへの変換	69
	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	72
	Cisco NX-OS への再変換	73

第 6 章	vPC トポロジでのスイッチの移行	77
	vPC フォークリフトアップグレード	77
	vPC のアップグレードとダウングレードの手順	77



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
bold	太字の文字は、表示どおりにユーザが入力するコマンドおよびキーワードです。
<i>italic</i>	イタリック体の文字は、ユーザが値を指定する引数です。
[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、イタリック体のスクリーン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

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、[シスコサービス](#)にアクセスしてください。
- サービス リクエストを送信するには、[シスコ サポート](#)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ク、トレーニング、認定関連の出版物を入手するには、[Cisco Press](#) にアクセスしてください。
- 特定の製品または製品ファミリの保証情報を探すには、[Cisco Warranty Finder](#) にアクセスしてください。

Cisco バグ検索ツール

[Cisco バグ検索ツール](#) (BST) は、追跡システム不具合とシスコ製品とソフトウェアの脆弱性の包括的なリストを維持する Cisco bug をゲートウェイとして動作する web ベースのツールです。BST は、製品とソフトウェアに関する詳細な障害情報を提供します。



第 1 章

新機能および変更された機能に関する情報

この章では、『Cisco Nexus 9000 シリーズ NX-OS リリース 9.x ソフトウェア アップグレード/ダウングレードガイド』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。

- [新機能および変更された機能に関する情報 \(1 ページ\)](#)

新機能および変更された機能に関する情報

表 1: NX-OS リリース 9.2(x) の新機能および変更された機能

機能	説明	変更されたリリース	参照先
NX-OS のオプション性	この機能が導入されました。	9.2(1)	Cisco NX-OS ソフトウェアのオプション (23 ページ)
アップグレードプロセス	Cisco NX-OS ソフトウェアをアップグレードする手順を修正しました。	9.2(1)	Cisco NX-OS ソフトウェアのアップグレード (17 ページ)
vPC トポロジ	vPC トポロジ内のアップグレードおよびダウングレード手順を追加しました。	9.2(1)	vPC のアップグレードとダウングレードの手順 (77 ページ)



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 \(4 ページ\)](#)
- [Cisco NX-OS ソフトウェア アップグレードの前提条件 \(8 ページ\)](#)
- [Cisco NX-OS ソフトウェア ダウングレードの前提条件 \(9 ページ\)](#)
- [Cisco NX-OS ソフトウェア アップグレードの注意事項 \(9 ページ\)](#)
- [Cisco NX-OS ソフトウェア ダウングレードの注意事項 \(15 ページ\)](#)
- [拡張 ISSU の設定 \(16 ページ\)](#)
- [Cisco NX-OS ソフトウェアのアップグレード \(17 ページ\)](#)
- [以前のソフトウェア リリースへのダウングレード \(20 ページ\)](#)

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。イメージのファイル名は、「nxos」から始まります (例: nxos.9.2.1.bin)。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 Nexus 9000 シリーズ スイッチと、Cisco Nexus 3132C-Z、3132Q-V、3164Q、3232C、3264C-E、3264Q、31108PC-V、31108TC-V、31128PQ、34180YC スイッチは、中断を伴うソフトウェア アップグレードおよび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-dk9.2.1.CSCab00001.gbin など)。SMU の詳細については、『[Cisco Nexus 9000 Series NX-OS System Management Configuration Guide](#)』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 Series FPGA/EPLD Upgrade Release Notes](#)』を参照してください。

ISSU について

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、In-Service Software Upgrade (ISSU、中断のないアップグレード) を実行できます。(サポート対象プラットフォームの完全なリストについては、『[Cisco NX-OS ソフトウェアアップグレードの注意事項](#)』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。

一部の Cisco Nexus 9000 シリーズスイッチと Cisco Nexus 3164Q および 31128PQ スイッチでは、拡張 ISSU がサポートされます。

Cisco Nexus 9000 シリーズスイッチでは、次の ISSU シナリオがサポートされます。

- シングル スーパーバイザ構成のトップオブラック (ToR) スイッチでの標準 ISSU の実行
- デュアル スーパーバイザ構成のエンドブロー (EoR) 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。

デュアルスーパーバイザ構成のエンドオブロー (EoR) スイッチでの標準 ISSU の実行

Cisco Nexus 9500 シリーズ スイッチは、ISSU に 2 つのスーパーバイザを必要とするモジュール方式の EoR スイッチです。必要な最低限の構成は、2 つのシステムコントローラと 2 つのファブリック モジュールです。

Cisco Nexus 9500 シリーズ スイッチでは、デフォルトの方式として並行アップグレードがサポートされます。並行方式では、モジュールを 1 つずつ順にアップグレードするのではなく、ひとまとまりの複数のモジュールをアップグレードします (次の図で概要を説明)。

図 1: Cisco Nexus 9500 シリーズスイッチの並行アップグレードプロセス



Cisco Nexus 9500 シリーズスイッチの並行アップグレードプロセスの各ステップは次のとおりです。

- 初めに、スーパーバイザがアップグレードされます（この手順にはスイッチオーバーを必要とします）。次に、ラインカード、ファブリックモジュール、システムコントローラ、および FEX がアップグレードされます。
- 並行アップグレードでスイッチオーバーが実行された後は、セカンダリスーパーバイザが動作を引き継ぎます。インストーラは、現在のラインカードとファブリックモジュールを特定します。
- 次に、インストーラはコンポーネントをバケットに分割します。半数のラインカードが最初のバケットに、半数のファブリックモジュールが2番目のバケットに、残りの半数のラインカードが3番目のバケットに、残りの半数のファブリックモジュールが4番目のバケットに、半数のシステムコントローラが5番目のバケットに、残りの半数のシステムコントローラが6番目のバケットに配置されます。
- 各バケットでアップグレードが正常に完了したら、次のバケットのアップグレードプロセスが開始されます。

- コンソールには、モジュールのバケット割り当てとアップグレードステータスが表示されます。

CLI を使用して順次アップグレードを行うことも可能です。

モジュラ スイッチに対して標準 ISSU を実行している間、データ プレーン トラフィックは中断されません。コントロール プレーンのダウンタイムは、約 6 秒未満です。



- (注) モジュール型 Cisco Nexus 9000 シリーズ スイッチ で ISSU を実行するための最小要件は、2 つのスーパーバイザ、2 つのシステム コントローラおよび 2 つのファブリック モジュールです。Cisco Nexus 9400 ラインカードには、ファブリック モジュールがフルに接続されている必要はありません。ファブリック モジュールが部分的に接続されていて、2 つのファブリック モジュールのみが Cisco Nexus 9400 ラインカードに使用されている場合、これらのファブリック モジュールはスロット 21 および 25 には接続しないでください。スロット 22、23、24、または 26 に接続してください。このようにすることで、システムは ISSU の実行中に高可用性を維持できます。

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



- (注) Cisco NX-OS リリース 9.2(1) または 9.2(2) 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修正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。

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。



- (注) 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ン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ンタイムは 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 Series NX-OS High Availability and Redundancy Guide](#)』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。

- デバイスまたはネットワークにアクセス可能などのユーザも、スケジュールされた時間にはデバイスまたはネットワークを設定しないでください。アップグレード中はデバイスを設定できません。show configuration session summary コマンドを使用して、アクティブなコンフィギュレーションセッションがないことを確認してください。
- デバイスで Cisco NX-OS ソフトウェア イメージをアップグレードまたはダウングレードする前に、アクティブなすべてのコンフィギュレーションセッションを保存、送信、または破棄します。デュアルスーパーバイザを搭載したデバイスでは、アクティブなコンフィギュレーションセッションがある場合、Cisco NX-OS ソフトウェアのアップグレード中にアクティブスーパーバイザモジュールをスタンバイスーパーバイザモジュールに切り替えられません。

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

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

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

コンフィギュレーションセッションの詳細については、『[Cisco Nexus 9000 Series NX-OS System Management Configuration Guide](#)』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



- (注) 『[Cisco Nexus 9000 Series NX-OS Release Notes](#)』には、各リリース固有のアップグレードの注意事項が含まれています。アップグレードを開始する前に、アップグレード先となるリリースのリリース ノートを参照してください。

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

- アップグレードは、ネットワークが安定しているときにスケジュールします。
- ソフトウェアイメージの破損につながるため、インストール手順の実行中の電源中断は回避してください。
- デュアル スーパーバイザ モジュールのあるデバイスでは、ソフトウェアのアップグレード中に切り替えが発生した場合に接続を維持するため、両方のスーパーバイザモジュール

がコンソールポートで接続されている必要があります。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 プラットフォームでは、Cisco NX-OS リリース 7.0(3)F3(4) より前のリリースからアップグレードするには **write erase** を実行してデバイスをリロードする必要があります。Cisco NX-OS リリース 7.0(3)F3(4) または 9.2(x) から以降の任意のリリース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が推奨されます。

- 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 Series NX-OS Security Configuration Guide, Release 9.x](#)』の「Configuring Control Plane Policing」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 Series Fundamentals Configuration Guide](#)』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 Series NX-OS System Management Configuration Guide, Release 9.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 3164Q、31128PQ、または 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。
- 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) 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
  LIISP 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-aggr1 %$ VDC-1 %$ Jun  5 19:11:20 %KERN-2-SYSTEM_MSG:
 [12633.219113] App vsh.bin on slot 1 vdc 1 SUP
sap 64098(cli_api queue) did not drop MTS_OPC_CLISH with msg_id 0x675ecf from sender
sap 64132(NULL) in 180 sec,
contact app owner - kernel

```

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。これらのログは 9.2(1) より前のリリースで無効化されました。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。
- Cisco NX-OS リリース 9.2(1) 以降では、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 9.2(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、これは無視できます。次の例を参照してください。

```

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

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

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

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

```

```

Preparing "bios" version info using image bootflash:/nxos.9.2.1.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)              9.2(1)I9(1)
yes
1       bios         v05.31(05/17/2018):v05.26(11/06/2017)  v05.31(05/17/2018)
no

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

```

- 次のプラットフォームで ISSU がサポートされています。



(注) 無停止の標準 ISSU は、Cisco NX-OS リリース 7.0(3)I7(4)、7.0(3)I7(5)、または 9.2(x) から Cisco NX-OS 9.2(x) リリースへのアップグレードでサポートされます。Cisco NX-OS 9.2(x) リリースへの拡張 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修正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。詳細については、[ISSU サポートマトリックス](#)を参照してください。

シリーズ	サポートされるプラットフォーム	ISSU をサポートしている初回リリース	ISSU でサポートされていない機能 ¹ 。
Cisco Nexus 9200 シリーズ	標準および拡張 ISSU : 9236C、9272Q、92160YC X、92300YC、および 92304QC	標準 ISSU : 7.0(3)I6(1) 拡張 ISSU : 7.0(3)I7(3)	ストレート型およびデュアル ホーム接続の FEX、セグメントルーティング、Tetration

シリーズ	サポートされるプラットフォーム	ISSU をサポートしている初回リリース	ISSU でサポートされていない機能 ¹ 。
Cisco Nexus 9300 シリーズ	<p>標準および拡張 ISSU : 9332PQ、9372PX、9372PX-E、9372TX、9372TX-E、9396PX、9396TX、93120TX、93128TX</p> <p>(注) これらの Cisco Nexus 9300 シリーズスイッチのうちの 1 つがスパンニングツリーのルートである場合、ISSU がサポートされます。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、show spanning-tree issu-impact コマンドを使用できます。</p>	<p>標準 ISSU : 7.0(3)I3(1)</p> <p>拡張ISSU : 7.0(3)I5(1)</p>	<p>デュアル ホーム接続 FEX、セグメントルーティング、VXLAN</p> <p>(注) ストレート型 FEX は、Cisco NX-OS リリース 7.0(3)I4(1) 以降でサポートされます。</p>
Cisco Nexus 9300-EX シリーズ	標準および拡張 ISSU : 93108TC-EX、93180LC-EX、および 93180YC-EX	<p>93108TC-EX および 93180YC-EX の標準 ISSU : 7.0(3)I6(1)</p> <p>93180LC-EX の標準 ISSU : 7.0(3)I7(1)</p> <p>拡張 ISSU : 7.0(3)I7(3)</p>	ストレート型およびデュアル ホーム接続の FEX、セグメントルーティング、Tetration
Cisco Nexus 9300-FX シリーズ	<p>標準 ISSU : なし</p> <p>拡張 ISSU : なし</p>		

シリーズ	サポートされるプラットフォーム	ISSU をサポートしている初回リリース	ISSU でサポートされていない機能 ¹ 。
Cisco Nexus 9500 シリーズ	標準 ISSU : 9504、9508、9516 (X9432PQ、X9464PX、X9464TX、X9536PQ、X9564PX、X9564TX、または X9636PQ ラインカード、デュアルスーパーバイザモジュール、最少 2 台のシステムコントローラ、2 台のファブリックモジュール搭載) (注) -R、-EX、および -FX ラインカード搭載の Cisco Nexus 9500 スイッチは ISSU をサポートしていません。 拡張 ISSU : なし	標準 ISSU : 7.0(3)I3(1)	デュアルホーム接続 FEX、セグメントルーティング、VXLAN (注) ストレート型 FEX は、Cisco NX-OS リリース 7.0(3)I4(1)以降でサポートされます。
Cisco Nexus 9000 シリーズ NX-OS ソフトウェアを実行する Cisco Nexus 3000 シリーズ	標準 ISSU : 3164Q、31128PQ、3232C、および 3264Q 拡張 ISSU : 3164Q および 31128PQ	3164Q および 31128PQ の標準 ISSU : 7.0(3)I3(1) 3232C および 3264Q の標準 ISSU : 7.0(3)I6(1) 3164Q および 31128PQ の拡張 ISSU : 7.0(3)I5(1)	FEX、セグメントルーティング、および 3164Q と 31128PQ の VXLAN 3232C と 3264Q のセグメントルーティング

¹ これらの機能に対して ISSU は中断を伴います。

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。

- 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。
- 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『[Hardware Installation Guide](#)』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 Series NX-OS System Management Configuration Guide](#)』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。

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 9.2(x) リリースへの拡張 ISSU はサポートされていません。これは、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になるカーネルの修正が存在するためです。

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

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： <pre>switch# configure terminal switch(config#)</pre>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	[no] boot mode lxc 例： <pre>switch(config)# boot mode lxc Using LXC boot mode</pre> 例：	拡張 (LXC) ISSU を有効または無効にします。

	コマンドまたはアクション	目的
	switch(config)# no boot mode lxc Using normal native boot mode	
ステップ 3	(任意) show boot mode 例 : switch(config)# show boot mode LXC boot mode is enabled 例 : switch(config)# show boot mode LXC boot mode is disabled	拡張 (LXC) ISSU の有効/無効を示します。
ステップ 4	copy running-config startup-config 例 : switch(config)# copy running-config startup-config	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	reload 例 : switch(config)# reload This command will reboot the system. (y/n)? [n] Y loader>	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、 Y 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 9.2(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) -R ライン カードを装備する Cisco Nexus 9500 プラットフォームの場合、以下の手順を使用して Cisco NX-OS 9.2(x) リリースからそれ以降の 9.2(x) リリースに、または Cisco NX-OS リリース 7.0(3)F3(4) から Cisco NX-OS 9.2(x) リリースにアップグレードしてください。



- (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 Series NX-OS Troubleshooting Guide](#)』を参照してください。

手順

ステップ 1 このアップグレード手順の例外については、ソフトウェア イメージ ファイルに関するリリース ノートを参照してください。『[Cisco Nexus 9000 Series NX-OS Release Notes](#)』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:
49152   Dec 10 14:43:39 2015 lost+found/
80850712 Dec 10 15:57:44 2015 n9000-dk9.9.2.1.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:n9000-dk9.9.2.1.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/
49152   Dec 10 14:43:39 2015 lost+found/
80850712 Dec 10 15:57:44 2015 n9000-dk9.9.2.1.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/n9000-dk9.9.2.1.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/nxos.9.2.2.bin
bootflash:nxos.9.2.2.bin
```

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

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

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

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

互換性チェック中、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
Incompatible image for ISSU	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。
Default upgrade is not hitless	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、 non-disruptive オプションを設定する必要があります。

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

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

- ステップ 12** `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:nxos.9.2.2.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。
- **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せん)。
- **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 NXOS ソフトウェア イメージがインストールされ、必要に応じて、実行中のイメージのさまざまなモジュールの BIOS がアップグレードされます。

ステップ 13 (任意) アップグレード プロセス全体を表示します。

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

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

```
switch# show version
```

ステップ 15 (任意) 必要に応じて、任意のライセンスをインストールし、必要な機能がデバイスで使用できるようにします。『[Cisco NX-OS Licensing Guide](#)』を参照してください。

以前のソフトウェア リリースへのダウングレード



(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 Series NX-OS Troubleshooting Guide](#)』を参照してください。

手順

ステップ 1 このダウングレード手順の例外については、ソフトウェア イメージ ファイルに関するリリース ノートを参照してください。『[Cisco Nexus 9000 Series NX-OS Release Notes](#)』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 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# copy scp://user@scpserver.cisco.com//download/n9000-dk9.9.2.1.bin  
bootflash:n9000-dk9.9.2.1.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:n9000-dk9.9.2.1.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:n9000-dk9.9.2.1.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) 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 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
```



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 \(23 ページ\)](#)
- [モジュラ パッケージの使用 \(25 ページ\)](#)
- [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 \(26 ページ\)](#)
- [RPM に関する情報 \(26 ページ\)](#)
- [YUM コマンドに関する情報 \(39 ページ\)](#)
- [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 \(58 ページ\)](#)
- [インストール操作ユーザ ロールの作成 \(61 ページ\)](#)

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。

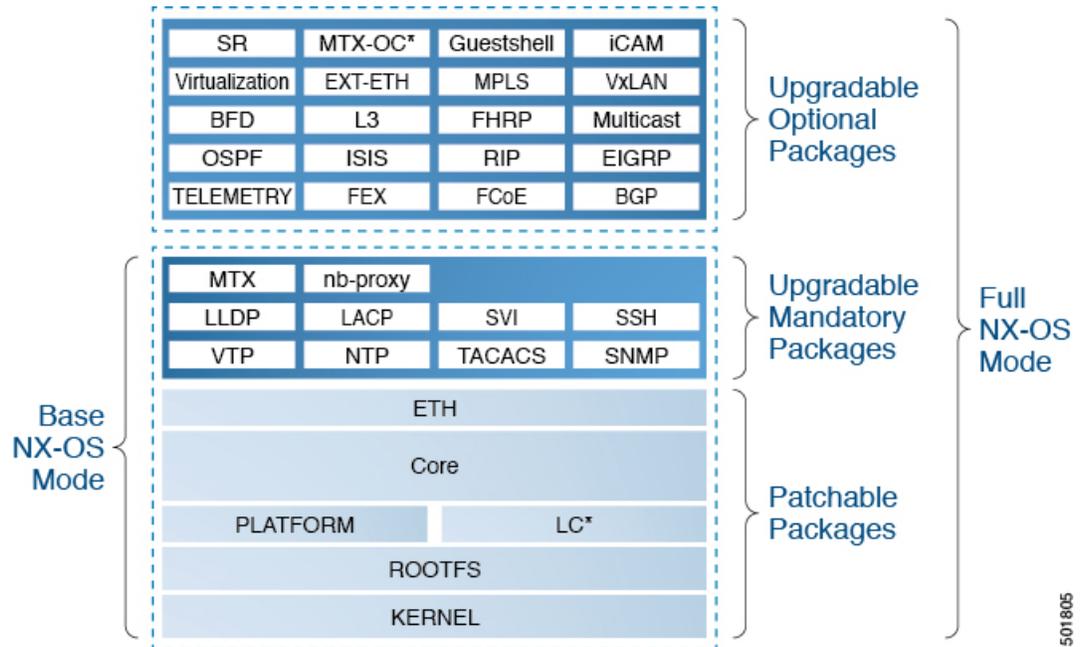
モジュール式の Cisco NX-OS ソフトウェアを使用する利点は次のとおりです。

- 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 2: 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にのみ使用する必要があります。

RPM と YUM

RPM (RedHat 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"> • feature openflow • feature evb • feature imp • feature netflow • feature sla_sender • feature sla_responder • feature sla twamp-server • feature sflow
FCoE	feature-set fcoe-npv
FEX	feature-set fex
FHRP	<ul style="list-style-type: none"> • feature hsrp • feature vrrpv3
iCAM	feature icam
ISIS	feature isis
MPLS	<ul style="list-style-type: none"> • feature mpls segment-routing • feature mpls evpn
マルチキャスト	<ul style="list-style-type: none"> • feature pim • feature pim6 • feature msdp • feature ngmvpn
OSPF	<ul style="list-style-type: none"> • feature ospf • feature ospfv3

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"> • feature nv overlay • feature fabric forwarding

NX-OS 機能 RPM インストールに関するガイドライン

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



- (注) RPM をシステムのリポジトリに手動でコピーをすることは避けてください。代わりに、install または 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 install add コマンドの一部として追加すると、RPM がこの場所にコピーされ、リロード時に常駐します。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、install add 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、install remove 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に選択する必要があります。
- ベース RPM は、**localdb** リポジトリに追加できません。

機能 RPM の操作に install CLI を使用する

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

表 5: 機能 RPM 操作の install CLI リファレンス

CLI	説明
install reset	<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>install reset コマンドも write erase 操作を実行します。次のメッセージがプロンプトに表示されます。</p> <pre>switch(config)# install reset</pre> <hr/> <p>WARNING!!This operation will remove all patches, upgraded packages, persisted etc configs, third party packages installed, startup configuration(write erase) and reload the switch with default packages.</p> <hr/> <p>Do you want to proceed with reset operation? (y/n)? [n]</p>
install reset nxos base	<p>この操作は、すべてのパッチ、アップグレードされたパッケージ、保持された etc コンフィグレーション、サードパーティのインストール済みパッケージ、スタートアップ設定（write erase）を削除して NX-OS をベースモードでインストールして、デフォルトのパッケージを使用してスイッチをリロードします。</p>
install reset nxos full	<p>この操作は、すべてのパッチ、アップグレードされたパッケージ、保持された etc コンフィグレーション、サードパーティのインストール済みパッケージ、スタートアップ設定（write erase）を削除して NXOS をフルモードでインストールして、デフォルトのパッケージ（必須およびオプションの RPM による）を使用してスイッチをリロードします。</p>

CLI	説明
install add <◇>	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 (patch/feature/third-party) を更新します。
install activate <rpm name>	リポジトリに存在する RPM をインストールします。
install commit <rpm name>	パッチ RPM に使用します。リロード時にパッチを保持します。
install deactivate <rpm name>	RPM をアンインストールします。
install remove <rpm name>	リポジトリから RPM ファイルを削除してリポジトリを更新します。
sh install active	ベース roots 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)。
sh install inactive	リポジトリに保持されていてインストールされていない RPM のリストを表示します。
sh install packages	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。

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

	コマンドまたはアクション	目的
ステップ 1	<pre>switch#install add bootflash:<keyfile> 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>	<p>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、/etc/pki/rpm-gpg/arm-Nexus9k-rel.gpg 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。</p>
ステップ 2	<pre>switch#install verify package <package-name></pre>	<p>パッケージを検証します。</p>

	コマンドまたはアクション	目的
ステップ 3	<p>または switch#install verify bootflash:<RPM file></p> <p>例 :</p> <pre>switch# install verify bootflash:vxlan-2.0.0.0-9.2.1.lib32_n9000.rpm</pre> <p>RSA signed switch#</p>	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

	コマンドまたはアクション	目的
ステップ 1	<p>show install packages</p> <p>例 :</p> <pre>switch# show install packages</pre> <p>Boot Image: NXOS Image: bootflash:/nxos.9.2.1.bin</p> <hr/> <pre>Installed Packages attr.x86_64 2.4.47-r0.0 installed Unsigned aufs-util.x86_64 3.14+git0+b59a2167a1-r0.0 installed Unsigned base-files.n9000 3.0.14-r89.0 installed Unsigned base-passwd.lib32_x86 3.5.29-r0.1.0 installed Unsigned bash.lib32_x86 4.3.30-r0.0 installed Unsigned bfd.lib32_n9000 2.0.0.0-9.2.1 installed Signed bgp.lib32_n9000 2.0.0.0-9.2.1 installed Signed binutils.x86_64 2.25.1-r0.0 installed Unsigned bridge-utils.x86_64 1.5-r0.0 installed Unsigned busybox.x86_64 1.23.2-r0.0 installed Unsigned busybox-udhcpc.x86_64 1.23.2-r0.0 installed Unsigned bzip2.x86_64 1.0.6-r5.0 installed</pre>	インストール済みのすべての RPM をクエリします。

	コマンドまたはアクション	目的
	<pre> 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+gitaa8187dd3b7ad67d8e5e3a15115d3eeef43a7ed1-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>	

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	<pre> install add <rpm>activate 例 : switch# install add bootflash:chef.rpm activate Adding the patch (/chef.rpm) [#####] 100% Install operation 868 completed successfully at Tue May 8 11:20:10 </pre>	RPM をインストールしてアクティブ化します。

	コマンドまたはアクション	目的
	<pre>2018 Activating the patch (/chef.rpm) [#####] 100% Install operation 869 completed successfully at Tue May 8 11:20:20 2018</pre>	

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

	コマンドまたはアクション	目的
ステップ 1	<pre>install add <rpm> 例 : switch# install add bootflash:vlan-2.0.1.0-9.2.1.lib32_n9000.rpm</pre>	RPM をインストールします。

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

例

```
switch#install activate vxlan

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

switch# show install active | grep vxlan

vxlan-2.0.0.0-9.2.1.lib32_n9000

switch# sh install inactive | grep vxlan

switch#
```

1 ステップの RPM のアップグレード

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

手順

	コマンドまたはアクション	目的
ステップ 1	install add <rpm>activate upgrade 例 : <pre>switch(config)# install add bootflash:bgp-2.0.2.0-9.2.1.lib32_n9000.rpm activate upgrade Adding the patch (/bgp-2.0.2.0-9.2.1.lib32_n9000.rpm)</pre>	RPM をインストールします。

	コマンドまたはアクション	目的
	<pre>[#####] 100% Install operation 870 completed successfully at Tue May 8 11:22:30 2018 Activating the patch (/bgp-2.0.2.0-9.2.1.lib32_n9000.rpm) [#####] 100% Install operation 871 completed successfully at Tue May 8 11:22:40 2018</pre>	

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.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

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 <rpm>activate downgrade</pre> <p>例 :</p>	RPM をダウングレードします。

	コマンドまたはアクション	目的
	<pre>switch(config)# install add bootflash:bgp-2.0.1.0-9.2.1.lib32_n9000.rpm activate downgrade Adding the patch (/bgp-2.0.1.0-9.2.1.lib32_n9000.rpm) [#####] 100% Install operation 872 completed successfully at Tue May 8 11:24:43 2018 Activating the patch (/bgp-2.0.1.0-9.2.1.lib32_n9000.rpm) [#####] 100% Install operation 873 completed successfully at Tue May 8 11:24:52 2018</pre>	

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

	コマンドまたはアクション	目的
ステップ 1	install remove <rpm> 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。

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。

```
#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
```

インストール済み RPM のリストをチェックする

```

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

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

RPM のインストール

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

```
bash-4.3# 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 | 951 B 00:00 ...
localdb/primary | 886 B 00:00 ...
localdb | 1/1
patching | 951 B 00:00 ...
thirdparty | 951 B 00:00 ...
thirdparty | 951 B 00:00 ...
```

```
Setting up Install Process
```

```
vxlan-2.0.1.0-9.2.1.lib32_n9000.rpm | 1.6 MB 00:00
```

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

Dependencies Resolved

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

Install 1 Package

Total size: 6.4 M

Installed size: 6.4 M

Is this ok [y/N]: y

Downloading Packages:

Running Transaction Check

Running Transaction Test

Transaction Test Succeeded

Running Transaction

Installing : vxlan-2.0.1.0-9.2.1.lib32_n9000

1/1

starting pre-install package version mgmt for vxlan

pre-install for vxlan complete

starting post-install package version mgmt for vxlan

post-install for vxlan complete

Installed:

vxlan.lib32_n9000 0:2.0.1.0-9.2.1

Complete!

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

```
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	Repository	Arch	Version	Size
Updating:				
vxlan		lib32_n9000	2.0.1.0-9.2.1	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 ...
patching 2/2
thirdparty | 951 B 00:00 ...
thirdparty | 951 B 00:00 ...
```

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

```
Dependencies Resolved
```

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

```

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 ...

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

groupinstall コマンドを使用する

```

thirdparty
| 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 ...

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	groups-repo	2.0.0.0-9.2.1	1.2 M
rip	lib32_n9000	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。

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

groupupdate コマンドを使用する

```
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	localdb	2.4 M	2.0.1.0-9.2.1
eigrp	lib32_n9000	locald	428 k	2.0.1.0-9.2.1
isis	lib32_n9000	local	1.2 M	2.0.1.0-9.2.1
ospf	lib32_n9000	localdb	2.8 M	2.0.1.0-9.2.1
rip	lib32_n9000	localdb	214 k	2.0.1.0-9.2.1
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

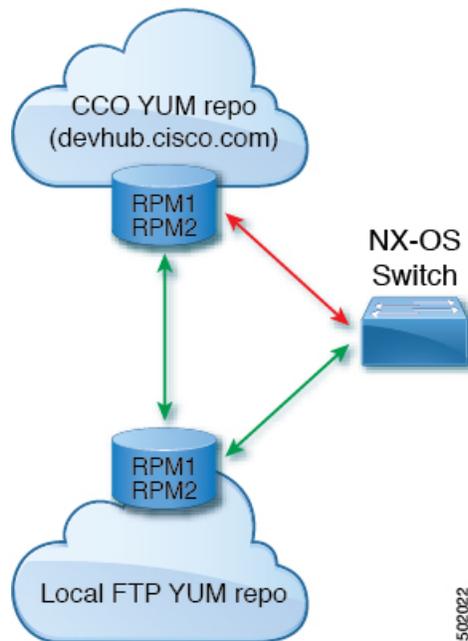
NX-OS CLI コマンド	YUM コマンド
show install inactive	yum list --patch-only available
show install active	yum list --patch-only installed
show install committed	yum list --patch-only committed
show install packages	yum list --patch-only
show install pkg-info	yum info --patch-only

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

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 3: 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 `open-nxos` リポジトリには、<https://devhub.cisco.com/artifactory/open-nxos/9.2.1/> からアクセスできます。

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

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

手順

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

ローカル FTP YUM リポジトリを作成する

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

手順

	コマンドまたはアクション	目的
ステップ 1	cat /etc/yum.repos.d/local.repo 例 : bash-4.3# cat /etc/yum.repos.d/local.repo [localrepo] name=localrepo baseurl= https://cshb.cisco.com/artifactory/qa-nxos/7.0-3-I2-1/x86_64/ enabled=1 gpgcheck=0 sslverify=0	/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、 local.repo リポジトリを作成してベース URL を追加します。

FTP サーバに到達するようにスイッチを設定する

	コマンドまたはアクション	目的
ステップ 2	bash 4.3 #yum repolist 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	nohup reposync -r <repo-name mentioned in the local.repo> -p <directory path to sync>& 例 : <pre>nohup reposync -r localrepo -p /var/ftp/ &</pre> <p>このコマンドは、/var/ftp/ 内部に名前 local.repo でディレクトリを作成し、すべてのパッケージを devhub.cisco.com 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	tail -f nouhup.out	同期のステータスを確認します。

FTP サーバに到達するようにスイッチを設定する

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

手順

	コマンドまたはアクション	目的
ステップ 1	run bash sudo su	Sudo ユーザとしてログインします。
ステップ 2	ip netns exec management ping <ip_address>	スイッチから ping 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。

	コマンドまたはアクション	目的
ステップ 3	cat /etc/yum/repos.d/ftp.repo 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 ssilverify=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	ip netns exec management bash	Bash シェルプロンプトを使用します。
ステップ 5	yum repolist 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	yum list available	新しいリポジトリで利用可能なパッケージをリストします。

インストール操作ユーザ ロールの作成

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。



第 4 章

高速リロードを使用した Cisco NX-OS ソフトウェアのアップグレード

この章では、スイッチの Cisco NX-OS ソフトウェアを高速リロードでアップグレード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。

- [高速リロードについて \(63 ページ\)](#)
- [イベントの高速リロードシーケンス \(64 ページ\)](#)
- [高速リロードの前提条件 \(64 ページ\)](#)
- [高速リロードに関する注意事項と制約事項 \(64 ページ\)](#)
- [高速リロードの実行と Cisco NX-OS ソフトウェアのアップグレード \(66 ページ\)](#)
- [高速リロードによる設定の保存 \(67 ページ\)](#)
- [その他の参考資料 \(68 ページ\)](#)

高速リロードについて

高速リロード機能を使用すると、`reload`を実行する前に、ユーザ名がフィギュレーションファイルに指定されていることを確認してください。高速リロードを使用して、スイッチ上のソフトウェアをアップグレードすることもできます。

高速リロード中、CPU 上で実行されている NXOS ソフトウェア イメージは新しいイメージをリロードし、CPU やファームウェアをリセットせずにこれを実行します。高速リロード中にトラフィックの短い中断がありますが、スイッチでこの機能を使用することで、コールドリブート時よりも高速にリロードできます。

高速リロードは、非中断モードで使用でき、その場合、プロンプトなしでインストールプロセスが実行されます。また、BGP 互換ピアの場合は、BGP グレースフルリスタートでインストールプロセスが実行されます。

イベントの高速リロードシーケンス

`fast-reload` コマンドを使用して高速リロードを実行すると、以下のイベントのシーケンスが発生します。

1. スイッチは、NXOS ソフトウェア イメージをロードし、カーネルをアップグレードします。すべてのアプリケーションはステートレス コールドリブートされ、スタートアップ コンフィギュレーションを介して再起動します。
2. コントロールプレーンが中断されます。この中断時に、すべての制御プロトコル通信が停止します。コントロールプレーンの中断は 90 秒未満です。
3. コントロールプレーンの中断後にすべてのコントロールプレーンのアプリケーションはステートレス コールドリブートされ、状態は保持されません。新しい設定は、スイッチのリロード時に適用されます。
4. データプレーンが中断されます。データプレーンの中断は 30 秒未満です。
5. フォワーディングプレーンでは、すべてのリンクが使用できなくなり、データプレーンの状態はリロード後は保持されません。トラフィックの転送が 30 秒以内に再開します。

高速リロードの前提条件

高速リロードには、次の前提条件があります。

- ブートフラッシュに十分な領域を使用できること。
- 高速リロードを許可するには、link aggregation control protocol (LACP) 高速タイマーが設定されていないこと。

高速リロードに関する注意事項と制約事項

高速リロードには、次の注意事項と制限事項があります。

- 高速リロードがサポートされるのは、Cisco Nexus 3164Q、3264C-E、および 92304QC スイッチのみです。
- 高速リロードは、Cisco NX-OS リリース 7.0(3)I7(4)、7.0(3)I7(5)、または 9.2(x) から Cisco NX-OS 9.2(x) リリースへのアップグレードでサポートされます。
- 高速リロードを使用した Cisco NX-OS ソフトウェアのダウングレードはサポートされていません。ソフトウェアをダウンロードするには、`install all` コマンドを使用します。
- スイッチまたはネットワークにアクセスできるすべての人がその時間にスイッチまたはネットワークを設定しないようにします。高速リロードの実行中にスイッチを設定するこ

とはできません。show configuration session summary コマンドを使用して、アクティブなコンフィギュレーションセッションがないことを確認してください。

- 高速リロードを実行する前に、アクティブなコンフィギュレーションセッションを保存、確定、または削除してください。アクティブなコンフィギュレーションセッションは警告なしで削除されます。
- 高速リロードを実行する前に、トポロジの変更（スパニング ツリー プロトコルの変更など）を実施してください。ただし、レイヤ2およびルーティングトポロジの変更はしないでください。
- 高速リロードの実行中にファンや電源の挿入や取り外しは行わないでください。
- 高速リロードは、ネットワークが安定しているときにスケジュールしてください。
- 高速リロードによる BIOS アップグレードはサポートされていません。
- コントロールプレーンの中断とデータプレーンの中断の間は、CPUの応答が停止します。
- **copy configuration-file startup-config** コマンドは、一部のコンフィギュレーションセットのみ高速リロードでサポートされます。
- **copy configuration-file startup-config** を実行し、続いて **fast-reload** コマンドまたは **reload** を実行する前に、ユーザ名がフィギュレーションファイルに指定されていることを確認してください。指定されていない場合は、スイッチにアクセスできなくなり、パスワード回復手順を実行してシステムをオンラインに戻すことが必要になります。パスワード回復手順については、『Cisco Nexus 9000 Series NX-OS Troubleshooting Guide』の「Power Cycling the Device to Recover the Administrator Password」の章を参照してください。
- 高速リロードは現在、次の2つの設定プロファイルをサポートしています。

高速リロードプロファイル 1

- 48 レイヤ 2 リンク
- 1 VLAN および SVI
- 16 レイヤ 3 ECMP リンク
- 6000 IPv4 LPM ルート、3000 IPv6 LPM ルート、200 IPv4 VIP、200 IPv6 VIP
- 2000 IPv4 ARP および 2000 Ipv6 ネイバー探索 (ND)

高速リロードプロファイル 2

- 24 レイヤ 2 ポートチャネル（それぞれに 2 メンバ）
- 24 VLAN および SVI
- 8 レイヤ 3 ポートチャネル ECMP（それぞれに 2 メンバ）
- 6000 IPv4 LPM ルート、3000 IPv6 LPM ルート、50 IPv4 VIP、50 IPv6 VIP
- 2000 IPv4 ARP および 2000 Ipv6 ネイバー探索 (ND)

高速リロードの実行と Cisco NX-OS ソフトウェアのアップグレード

この手順を使用することで、コールドリブート時よりも高速にデバイスをリブートできます。ソフトウェアイメージを指定すると、スイッチ上のソフトウェアがアップグレードされます。

始める前に

実行ソフトウェアイメージが存在することを確認し、高速リロード操作の影響を分析してください。

手順

ステップ 1 スイッチにログインします。

ステップ 2 `fast-reload [save-config] [trigger-gr] [nxos bootflash:nxos-image-name] [non-interruptive]` コマンドを使用して、高速リロードを実行します。

例：

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

次のオプションを使用できます。

- **save-config**— 後続の高速リロード操作で新しい NXOS ソフトウェア イメージがブート変数として使用されていることを確認します。 **save-config** オプションを使用しない場合、このコマンドは、ブート変数を保存しないため、後続の高速リロード操作で以前のソフトウェア イメージがブート変数として使用されます。
- **trigger-gr**— デフォルトでは、高速リロード機能には Border Gateway Protocol (BGP) のピアがグレースフル リスタートをできることが必要です。 **trigger-gr** オプションにより、アグレッシブ タイマーによる再起動のサポートが追加されます。
- **nxos bootflash:nxos-image-name**— NXOS ソフトウェア イメージの名前を指定します。高速リロード機能をサポートするソフトウェア バージョンを指定してください。
- **non-interruptive**— プロンプトなしで高速リロードを実行します。このオプションでは、すべてのエラーチェックと健全性チェックがスキップされたため、このオプションを選択する前に、高速リロードがシステムで動作可能であることを確認してください。

例

この例では、高速リロードを使用してスイッチの Cisco NX-OS ソフトウェアをアップグレードする方法を示しています。

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

高速リロードによる設定の保存

この表は、**fast-reload** コマンドのさまざまなバリエーションで設定を保存する場合に予想される動作を示しています。

コマンド	予想される動作
fast-reload	設定の変更がある場合、画面にプロンプトが表示され、その応答に基づいて copy running-config startup-config が実行されます。
fast-reload non-interruptive	プロンプトは表示されず、設定は保存されていません。 save-config オプションまたは copy running-config startup-config コマンドを使用して設定を保存する必要があります。
fast-reload nxos bootflash:nxos-image-name [non-interruptive trigger-gr]	イメージが同じイメージである場合でも、暗黙的に copy running-config startup-config が実行されます。
copy configuration-file startup-config fast-reload	起動後、暗黙的に copy configuration-file startup-config が実行され、起動されたイメージにブート変数が設定されます。
copy configuration-file startup-config fast-reload nxos bootflash:nxos-image-name	起動後、指定されたイメージにブート変数が暗黙的に設定され、 copy configuration-file startup-config が実行されます。



- (注) ユーザ名がコンフィグレーションファイルで指定されていることを確認してから、**copy configuration-file startup-config** を実行し、続いて **fast-reload** または **reload** コマンドを実行してください。指定されていない場合は、スイッチにアクセスできなくなり、パスワード回復手順を実行してシステムをオンラインに戻すことが必要になります。パスワード回復手順については、『[Cisco Nexus 9000 Series NX-OS Troubleshooting Guide](#)』の「Power Cycling the Device to Recover the Administrator Password」の章を参照してください。

その他の参考資料

関連資料

関連項目	マニュアル タイトル
reload コマンド	『Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide』



第 5 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 \(69 ページ\)](#)
- [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 \(72 ページ\)](#)
- [Cisco NX-OS への再変換 \(73 ページ\)](#)

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) Cisco Nexus 3164Q または 31128PQ スイッチを ACI ブートモードに変換することはできません。



(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

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。

- 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。
- スイッチで SPC サービスを有効化します。

例：

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

- 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に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 /secredata/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。

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。

手順

ステップ 1 スイッチをリロードします。

例：

```
admin@apic1:aci> 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

ステップ 1 スイッチをリロードします。

例：

```
admin@apic1:aci> 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 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-image-name
```

(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、**cmdline init_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 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
```

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：

```
switch# copy source filesystem 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 スーパーバイザで手順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。



第 6 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 \(77 ページ\)](#)
- [vPC のアップグレードとダウングレードの手順 \(77 ページ\)](#)

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](#)』の「vPC Forklift Upgrade Scenario」を参照してください。

vPC のアップグレードとダウングレードの手順

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。

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 rs** 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

```

4. 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。

