Cisco IOS XE Bengaluru 17.6.x プログラマビリティ コンフィギュレーション ガイド

初版：2021 年 7 月 31 日

最終更新：2022 年 12 月 26 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/

はじめに

ここでは、このマニュアルの表記法、および他資料の入手方法について説明します。また、シスコ製品のマニュアルの最新情報についても説明します。

（注） この製品のマニュアルセットは、偏向のない言語を使用するように配慮されています。このドキュメントセットでの偏向のない言語とは、年齢、障害、性別、人種的アイデンティティ、民族的アイデンティティ、性的指向、社会経済的地位、およびインターセクショナリティに基づく差別を意味しない言語として定義されています。製品ソフトウェアのユーザインターフェイスにハードコードされている言語、基準ドキュメントに基づいて使用されている言語、または参照されているサードパーティ製品で使用されている言語によりドキュメントに例外が存在する場合があります。

表記法

このマニュアルでは、以下の表記法を使用しています。

表記法	説明
^ または Ctrl	^ 記号と Ctrl は両方ともキーボードの Control（Ctrl）キーを表します。たとえば、^D または Ctrl+D というキーの組み合わせは、Ctrl キーを押しながら D キーを押すことを意味します（ここではキーを大文字で表記していますが、小文字で入力してもかまいません）。
太字	コマンド、キーワード、およびユーザーが入力するテキストは**太字**で記載されます。

表記法	説明
italic フォント	文書のタイトル、新規用語、強調する用語、およびユーザが値を指定する引数は、イタリック体で示しています。
`Courier` フォント	システムが表示する端末セッションおよび情報は、`courier` フォントで示しています。
太字の `courier` フォント	**太字の `Courier` フォント**は、ユーザが入力しなければならないテキストを示します。
[x]	角カッコの中の要素は、省略可能です。
...	構文要素の後の省略記号（3 つの連続する太字ではないピリオドでスペースを含まない）は、その要素を繰り返すことができることを示します。
|	パイプと呼ばれる縦棒は、一連のキーワードまたは引数の選択肢であることを示します。
[x | y]	どれか 1 つを選択できる省略可能なキーワードは、角カッコで囲み、縦棒で区切って示しています。
{x | y}	どれか 1 つを選択しなければならない必須キーワードは、波カッコで囲み、縦棒で区切って示しています。
[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。
string	引用符を付けない一組の文字。string の前後には引用符を使用しません。引用符を使用すると、その引用符も含めて string とみなされます。
< >	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。

読者への警告の表記法

このマニュアルでは、読者への警告に次の表記法を使用しています。

（注）	「注釈」です。役立つ情報やこのマニュアルに記載されていない参照資料を紹介しています。
ヒント	「問題解決に役立つ情報」です。
注意	「要注意」の意味です。機器の損傷またはデータ損失を予防するための注意事項が記述されています。
ワンポイント アドバイス	時間を節約する方法です。ここに紹介している方法で作業を行うと、時間を短縮できます。
警告	安全上の重要な注意事項 This warning symbol means danger. 人身事故を予防するための注意事項が記述されています。機器の取り扱い作業を行うときは、電気回路の危険性に注意し、一般的な事故防止対策に留意してください。各警告の最後に記載されているステートメント番号を基に、装置に付属の安全についての警告を参照してください。ステートメント 1071 SAVE THESE INSTRUCTIONS

関連資料

マニュアルの入手方法およびテクニカル サポート

マニュアルの入手方法、テクニカルサポート、その他の有用な情報について、毎月更新される『更新情報』を参照してください。シスコの新規および改訂版の技術マニュアルの一覧も示されています。

http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html

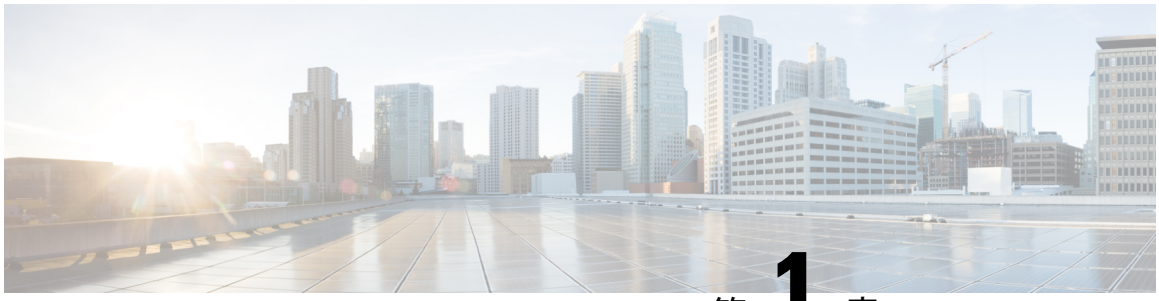
『更新情報』は RSS フィードとして購読できます。また、リーダー アプリケーションを使用してコンテンツがデスクトップに直接配信されるように設定することもできます。RSS フィードは無料のサービスです。シスコは現在、RSS バージョン 2.0 をサポートしています。

偏向のないドキュメントに関する免責事項

ガイダンス：それぞれのドキュメントで次の注意事項を再利用してください。

（注） この製品のマニュアルセットは、偏向のない言語を使用するように配慮されています。このドキュメントセットでの偏向のない言語とは、年齢、障害、性別、人種的アイデンティティ、民族的アイデンティティ、性的指向、社会経済的地位、およびインターセクショナリティに基づく差別を意味しない言語として定義されています。製品ソフトウェアのユーザインターフェイスにハードコードされている言語、基準ドキュメントに基づいて使用されている言語、または参照されているサードパーティ製品で使用されている言語によりドキュメントに例外が存在する場合があります。

目 次



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、すべての機能についてリリース固有の情報を記載しています。

新機能および変更された機能に関する情報

次の表は、新機能および変更機能、サポート対象のプラットフォーム、および機能へのリンクをまとめたものです。

表 *1*：新機能および変更機能に関する情報

機能	リリースとプラットフォーム
プロビジョニング	

新機能および変更された機能に関する情報

機能	リリースとプラットフォーム
ゼロ タッチ プロビジョニング	

新機能および変更された機能に関する情報

機能	リリースとプラットフォーム
	Cisco IOS XE Everest 16.5.1a • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.5.1b • Cisco 4000 シリーズ サービス統合型ルータ Cisco IOS XE Everest 16.6.2 • Cisco Catalyst 9400 シリーズ スイッチ Cisco IOS XE Fuji 16.7.1 • Cisco ASR 1000 アグリゲーション サービス ルータ（ASR1001-X、ASR1001-HX、ASR1002-X、ASR1002-HX） Cisco IOS XE Fuji 16.8.1a • Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ Cisco IOS XE Fuji 16.8.2 • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ（ASR1004、ASR1006、ASR1006-X、ASR1009-X、ASR1013） Cisco IOS XE Gibraltar 16.12.1 • Cisco Catalyst 9200 シリーズ スイッチ （注） この機能は C9200L SKU ではサポートされていません。 • Cisco Catalyst 9300L SKU • Cisco Catalyst 9600 シリーズ スイッチ • Cisco Catalyst 9800-40 ワイヤレスコントローラ • Cisco Catalyst 9800-80 ワイヤレスコントローラ Cisco IOS XE Amsterdam 17.2.1 • Cisco Cloud Services Router 1000V シリーズ

新機能および変更された機能に関する情報

機能	リリースとプラットフォーム
	• Cisco C1100 ターミナル サービス ゲートウェイ（C1100TGX-1N24P32A でのみサポート） Cisco IOS XE Amsterdam 17.3.1 • Cisco Catalyst 8200 シリーズ エッジ プラットフォーム • Cisco Catalyst 8300 シリーズ エッジ プラットフォーム Cisco IOS XE Bengaluru 17.4.1 • Cisco Catalyst 8000V Edge ソフトウェア
ゼロタッチプロビジョニング：HTTP ダウンロード	Cisco IOS XE Fuji 16.8.1 • Cisco 4000 シリーズ サービス統合型ルータ • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Fuji 16.8.1a • Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ
ゼロタッチプロビジョニングのための DHCPv6 のサポート	Cisco IOS XE Fuji 16.9.1 • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Amsterdam 17.3.2a • Cisco Catalyst 9800-40 シリーズ ワイヤレス コントローラ • Cisco Catalyst 9800-80 シリーズ ワイヤレス コントローラ

機能	リリースとプラットフォーム
iPXE	Cisco IOS XE Denali 16.3.2 および Cisco IOS XE Everest 16.5.1a • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3650 シリーズ スイッチ Cisco IOS XE Everest 16.6.1 • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.6.2 • Cisco Catalyst 9400 シリーズ スイッチ Cisco IOS XE Fuji 16.8.1a • Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ Cisco IOS XE Fuji 16.9.2 • Cisco Catalyst 9200 シリーズ スイッチ Cisco IOS XE Gibraltar 16.11.1 • Cisco Catalyst 9600 シリーズ スイッチ
シェルとスクリプト化	

新機能および変更された機能に関する情報

機能	リリースとプラットフォーム
ゲスト シェル	

新機能および変更された機能に関する情報

機能	リリースとプラットフォーム
	Cisco IOS XE Everest 16.5.1a • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.5.1b • Cisco 4000 シリーズ サービス統合型ルータ Cisco IOS XE Everest 16.6.2 • Cisco Catalyst 9400 シリーズ スイッチ Cisco IOS XE Fuji 16.7.1 • Cisco ASR 1000 アグリゲーション サービス ルータ（ASR1001-X、ASR1001-HX、ASR1002-X、ASR1002-HX） • Cisco Cloud Services Router 1000V シリーズ Cisco IOS XE Fuji 16.8.1 • Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ Cisco IOS XE Fuji 16.9.1 • Cisco 1000 シリーズ サービス統合型ルータ Cisco IOS XE Gibraltar 16.11.1b • Cisco Catalyst 9800-40 ワイヤレスコントローラ • Cisco Catalyst 9800-80 ワイヤレスコントローラ Cisco IOS XE Gibraltar 16.12.1 • Cisco Catalyst 9200 シリーズ スイッチ （注） この機能は C9200L SKU ではサポートされていません。 • Cisco Catalyst 9300L SKU • Cisco Catalyst 9600 シリーズ スイッチ Cisco IOS XE Amsterdam 17.3.1

新機能および変更された機能に関する情報

機能	リリースとプラットフォーム
	• Cisco Catalyst 8200 シリーズ エッジ プラットフォーム • Cisco Catalyst 8300 シリーズ エッジ プラットフォーム
ゲストシェルでの Python 3 のサポート	Cisco IOS XE Amsterdam 17.1.1 • Cisco 1000 シリーズ サービス統合型ルータ • Cisco ASR 1000 アグリゲーション サービス ルータ（ASR1000-RP1、ASR1000-RP2、ASR1000-RP3、ASR1001-X、ASR1001-HX、ASR1002-X、ASR1002-HX） • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ • Cisco ISR 4000 シリーズ サービス統合型ルータ
ゲストシェルからの NETCONF アクセス	Cisco IOS XE Bengaluru 17.6.1 • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 および 9300L シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 および 9500 ハイ パフォーマンス シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ

機能	リリースとプラットフォーム
Python API	Cisco IOS XE Everest 16.5.1a • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.5.1b • Cisco 4000 シリーズ サービス統合型ルータ Cisco IOS XE Everest 16.6.2 • Cisco Catalyst 9400 シリーズ スイッチ Cisco IOS XE Fuji 16.7.1 • Cisco ASR 1000 アグリゲーション サービス ルータ（ASR1001-X、ASR1001-HX、ASR1002-X、ASR1002-HX） • Cisco Cloud Services Router 1000V シリーズ Cisco IOS XE Fuji 16.8.1a • Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ

新機能および変更された機能に関する情報

機能	リリースとプラットフォーム
Python CLI モジュール	Cisco IOS XE Everest 16.5.1a • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.5.1b • Cisco 4000 シリーズ サービス統合型ルータ Cisco IOS XE Everest 16.6.2 • Cisco Catalyst 9400 シリーズ スイッチ Cisco IOS XE Fuji 16.7.1 • Cisco ASR 1000 アグリゲーション サービス ルータ（ASR1001-X、ASR1001-HX、ASR1002-X、ASR1002-HX） • Cisco Cloud Services Router 1000V シリーズ Cisco IOS XE Fuji 16.8.1 • 最低 4 GB の RAM を搭載した Cisco 4000 シリーズ サービス統合型ルータ モデル • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ（ASR1004、ASR1006、ASR1006-X、ASR1009-X、ASR1013） Cisco IOS XE Fuji 16.8.1a • Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ

機能	リリースとプラットフォーム
EEM Python モジュール	Cisco IOS XE Everest 16.5.1a • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.5.1b • Cisco 4000 シリーズ サービス統合型ルータ Cisco IOS XE Everest 16.6.2 • Cisco Catalyst 9400 シリーズ スイッチ Cisco IOS XE Fuji 16.7.1 • Cisco ASR 1000 アグリゲーション サービス ルータ（ASR1001-X、ASR1001-HX、ASR1002-X、ASR1002-HX） • Cisco Cloud Services Router 1000V シリーズ Cisco IOS XE Fuji 16.8.1a • Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ
モデル駆動型プログラマビリティ	

新機能および変更された機能に関する情報

機能	リリースとプラットフォーム
NETCONF プロトコル	Cisco IOS XE Denali 16.3.1 • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco 4000 シリーズ サービス統合型ルータ Cisco IOS XE Everest 16.5.1a • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.6.2 • Cisco Catalyst 9400 シリーズ スイッチ Cisco IOS XE Fuji 16.7.1 • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco Network Convergence System 4200 シリーズ Cisco IOS XE Fuji 16.9.2 • Cisco Catalyst 9200 シリーズ スイッチ Cisco IOS XE ジブラルタル 16.10.1 • Cisco Catalyst 9800 シリーズ ワイヤレス コントローラ • Cisco Network Convergence System 520 シリーズ • Cisco IR1101 耐環境性能 サービス統合型ルータ Cisco IOS XE Gibraltar 16.11.1 • Cisco Catalyst IE 3200、3300、3400 高耐久性シリーズ • Cisco エンベデッド サービス 3300 シリーズ スイッチ

機能	リリースとプラットフォーム
NETCONF および RESTCONF IPv6 のサポート	Cisco IOS XE Fuji 16.8.1a • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco cBR-8 コンバージド ブロードバンド ルータ • Cisco Cloud Services Router 1000V シリーズ Cisco IOS XE Gibraltar 16.11.1 • Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ
NETCONF グローバル ロックおよびセッションの kill	Cisco IOS XE Fuji 16.8.1a • Cisco 1100 シリーズ サービス統合型ルータ • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco cBR-8 コンバージド ブロードバンド ルータ • Cisco Cloud Services Router 1000V シリーズ

新機能および変更された機能に関する情報

機能	リリースとプラットフォーム
NETCONF-YANG SSH サーバのサポート	Cisco IOS XE Gibraltar 16.12.1 • Cisco 1000 シリーズ サービス統合型ルータ • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco ASR 1000 アグリゲーション サービス ルータ • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ • Cisco Catalyst 9800 シリーズ ワイヤレス コントローラ • Cisco cBR-8 コンバージド ブロードバンド ルータ • Cisco Cloud Services Router 1000V シリーズ • Cisco Network Convergence System 520 シリーズ • Cisco Network Convergence System 4200 シリーズ
コンフィギュレーション データベースの副次的同期	Cisco IOS XE Bengaluru 17.4.1 • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ

機能	リリースとプラットフォーム
RESTCONF プロトコル	

新機能および変更された機能に関する情報

機能	リリースとプラットフォーム
	Cisco IOS XE Everest 16.6.1 • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 アグリゲーション サービス ルータ • Cisco Cloud Services Router 1000V シリーズ Cisco IOS XE Fuji 16.8.1a • Cisco 1000 シリーズ サービス統合型ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 および 9500 ハイ パフォーマンス シリーズ スイッチ • Cisco cBR-8 コンバージド ブロードバンド ルータ • Cisco Network Convergence System 4200 シリーズ Cisco IOS XE Fuji 16.9.2 • Cisco Catalyst 9200 シリーズ スイッチ Cisco IOS XE Gibraltar 16.11.1 • Cisco Catalyst 9600 シリーズ スイッチ • Cisco Catalyst 9800-CL ワイヤレスコントローラ • Cisco Catalyst 9800-40 ワイヤレスコントローラ • Cisco Catalyst 9800-80 ワイヤレスコントローラ • Cisco Network Convergence System 520 シリーズ Cisco IOS XE Gibraltar 16.12.1 • Cisco Catalyst 9800-L ワイヤレスコントローラ Cisco IOS XE Amsterdam 17.3.1

機能	リリースとプラットフォーム
	• Cisco Catalyst 8200 シリーズ エッジ プラットフォーム • Cisco Catalyst 8300 シリーズ エッジ プラットフォーム Cisco IOS XE Bengaluru 17.4.1 • Cisco Catalyst 8000V Edge ソフトウェア
RESTCONF YANG パッチのサポート	Cisco IOS XE Amsterdam 17.1.1 • Cisco 1000 シリーズ サービス統合型ルータ • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 1000 アグリゲーション サービス ルータ（ASR1000-RP2、ASR1000-RP3、ASR1001-HX、ASR1001-X、ASR1002-HX、ASR1002-X） • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco cBR-8 コンバージド ブロードバンド ルータ • Cisco Cloud Services Router 1000V シリーズ • Cisco Catalyst IE3200 高耐久性シリーズ • Cisco Catalyst IE3300 高耐久性シリーズ • Cisco Catalyst IE3400 高耐久性シリーズ • Cisco IR1101 耐環境性能 サービス統合型ルータ • Cisco Network Convergence System 520 シリーズ • Cisco Network Convergence System 4200 シリーズ

新機能および変更された機能に関する情報

機能	リリースとプラットフォーム
NETCONF および RESTCONF のサービスレベル ACL	Cisco IOS XE Gibraltar 16.11.1 • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco Catalyst IE3200 高耐久性シリーズ • Cisco Catalyst IE3300 高耐久性シリーズ • Cisco Catalyst IE3400 高耐久性シリーズ • Cisco エンベデッド サービス 3300 シリーズ スイッチ • Cisco IR1101 耐環境性能 サービス統合型ルータ • Cisco Network Convergence System 4200 シリーズ • Cisco Network Convergence System 520 シリーズ

機能	リリースとプラットフォーム
gNMI プロトコル	Cisco IOS XE Fuji 16.8.1a • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Gibraltar 16.10.1 • Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ Cisco IOS XE Gibraltar 16.11.1 • Cisco Catalyst 9600 シリーズ スイッチ Cisco IOS XE Gibraltar 16.12.1 • Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300L SKU • Cisco cBR-8 コンバージド ブロードバンド ルータ Cisco IOS XE Amsterdam 17.1.1 • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco Network Convergence System 520 シリーズ • Cisco Network Convergence System 4200 シリーズ Cisco IOS XE Amsterdam 17.2.1r • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ

新機能および変更された機能に関する情報

機能	リリースとプラットフォーム
gNMI ユーザ名とパスワードによる認証	Cisco IOS XE Gibraltar 16.12.1 • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ • Cisco Catalyst IE3200 高耐久性シリーズ • Cisco Catalyst IE3200 高耐久性シリーズ • Cisco Catalyst IE3300 高耐久性シリーズ • Cisco Catalyst IE3400 高耐久性シリーズ • Cisco エンベデッド サービス 3300 シリーズ スイッチ
gRPC ネットワーク操作インターフェイス：gNOI 証明書の管理および証明書サービスによる gNOI ブートストラップ	Cisco IOS XE Amsterdam 17.3.1 • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ
gNOI OS インストールサービス	Cisco IOS XE Bengaluru 17.5.1 • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ

機能	リリースとプラットフォーム
モデルベースの AAA	Cisco IOS XE Fuji 16.8.1 • Cisco 1000 シリーズ サービス統合型ルータ • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco Cloud Services Router 1000V シリーズ • Cisco Network Convergence System 4200 Cisco IOS XE Fuji 16.8.1a • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ

新機能および変更された機能に関する情報

機能	リリースとプラットフォーム
モデル駆動型テレメトリ NETCONF ダイヤルイン	

新機能および変更された機能に関する情報

機能	リリースとプラットフォーム
	Cisco IOS XE Everest 16.6.1 • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.6.2 • Cisco Catalyst 9400 シリーズ スイッチ Cisco IOS XE Fuji 16.7.1 • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 アグリゲーション サービス ルータ（ASR1001-HX、ASR1001-X、ASR1002-HX、ASR1002-X） Cisco IOS XE Fuji 16.8.1 • Cisco 1000 シリーズ サービス統合型ルータ • Cisco ASR 1000 RP2 および RP3 シリーズ アグリゲーション サービス ルータ Cisco IOS XE Fuji 16.8.1a • Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ Cisco IOS XE Fuji 16.9.1 • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco cBR-8 コンバージド ブロードバンド ルータ • Cisco Network Convergence System 4200 シリーズ Cisco IOS XE Fuji 16.9.2 • Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300L SKU Cisco IOS XE Gibraltar 16.10.1

新機能および変更された機能に関する情報

機能	リリースとプラットフォーム
	• Cisco クラウド サービス ルータ 1000v • Cisco Network Convergence System 520 シリーズ Cisco IOS XE Gibraltar 16.11.1 • Cisco Catalyst 9600 シリーズ スイッチ Cisco IOS XE Amsterdam 17.3.1 • Cisco Catalyst 8200 シリーズ エッジ プラットフォーム • Cisco Catalyst 8300 シリーズ エッジ プラットフォーム Cisco IOS XE Bengaluru 17.4.1 • Cisco Catalyst 8000V Edge ソフトウェア

新機能および変更された機能に関する情報

機能	リリースとプラットフォーム
モデル駆動型テレメトリ gRPC ダイヤルアウト	Cisco IOS XE Gibraltar 16.10.1 • Cisco 1000 シリーズ サービス統合型ルータ • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300 および 9300L シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 および 9500 ハイ パフォーマンス シリーズ スイッチ • Cisco cBR-8 コンバージド ブロードバンド ルータ • Cisco Cloud Services Router 1000V シリーズ • Cisco Network Convergence System 520 シリーズ • Cisco Network Convergence System 4200 シリーズ Cisco IOS XE Gibraltar 16.11.1 • Cisco Catalyst 9600 シリーズ スイッチ Cisco IOS XE Amsterdam 17.3.1 • Cisco Catalyst 8200 シリーズ エッジ プラットフォーム • Cisco Catalyst 8300 シリーズ エッジ プラットフォーム Cisco IOS XE Bengaluru 17.4.1 • Cisco Catalyst 8000V Edge ソフトウェア

新機能および変更された機能に関する情報

機能	リリースとプラットフォーム
モデル駆動型テレメトリ gNMI ダイヤルイン	Cisco IOS XE Gibraltar 16.12.1 • Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300 および 9300L シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 および 9500 ハイ パフォーマンス シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ • Cisco cBR-8 コンバージド ブロードバンド ルータ Cisco IOS XE Amsterdam 17.1.1 • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco Network Convergence System 520 シリーズ • Cisco Network Convergence System 4200 シリーズ Cisco IOS XE Amsterdam 17.2.1 • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ Cisco IOS XE Amsterdam 17.3.1 • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ

機能	リリースとプラットフォーム
GRPC ダイヤルアウト用の TLS	Cisco IOS XE Amsterdam 17.1.1 • Cisco 1000 シリーズ サービス統合型ルータ • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ ASR1000-RP1、ASR1000-RP2、ASR1000-RP3、ASR1001-HX、ASR1001-X、ASR1002-HX、ASR1002-X） • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 および 9500 ハイ パフォーマンス シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ • Cisco cBR-8 コンバージド ブロードバンド ルータ • Cisco Cloud Services Router 1000V シリーズ • Cisco Catalyst IE3200 高耐久性シリーズ • Cisco Catalyst IE3300 高耐久性シリーズ • Cisco Catalyst IE3400 高耐久性シリーズ • Cisco IR1101 耐環境性能 サービス統合型ルータ • Cisco Network Convergence System 520 シリーズ • Cisco Network Convergence System 4200 シリーズ
TLDP 変更時の通知	Cisco IOS XE Amsterdam 17.2.1 • Cisco 4000 シリーズ サービス統合型ルータ • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ

新機能および変更された機能に関する情報

機能	リリースとプラットフォーム
テレメトリのサブスクリプションの kill	Cisco IOS XE Gibraltar 16.11.1 • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco IR1101 耐環境性能 サービス統合型ルータ • Cisco Network Convergence System 4200 シリーズ • Cisco Network Convergence System 520 シリーズ Cisco IOS XE Gibraltar 16.11.1a • Cisco 1000 シリーズ サービス統合型ルータ • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco Cloud Services Router 1000V シリーズ
gRPC サブスクリプションの FQDN サポート	Cisco IOS XE Bengaluru 17.6.1 • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ（RSP2） • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9800-40 ワイヤレスコントローラ • Cisco Catalyst 9800-80 ワイヤレスコントローラ

新機能および変更された機能に関する情報

機能	リリースとプラットフォーム
サービス中モデル更新プログラム	Cisco IOS XE Everest 16.5.1a • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.5.1b • Cisco 4000 シリーズ サービス統合型ルータ Cisco IOS XE Everest 16.6.1 • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ Cisco IOS XE Everest 16.6.2 • Cisco Catalyst 9400 シリーズ スイッチ Cisco IOS XE Fuji 16.7.1 • Cisco ASR 1000 アグリゲーション サービス ルータ Cisco IOS XE Fuji 16.8.1a • Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ
アプリケーション ホスティング	
アプリケーション ホスティング	Cisco IOS XE Gibraltar 16.12.1 • Cisco Catalyst 9300 シリーズ スイッチ Cisco IOS XE Amsterdam 17.1.1 • Cisco Catalyst 9400 シリーズ スイッチ Cisco IOS XE Amsterdam 17.2.1 • Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ Cisco IOS XE Bengaluru 17.5.1 • Cisco Catalyst 9410 シリーズ スイッチ

新機能および変更された機能に関する情報

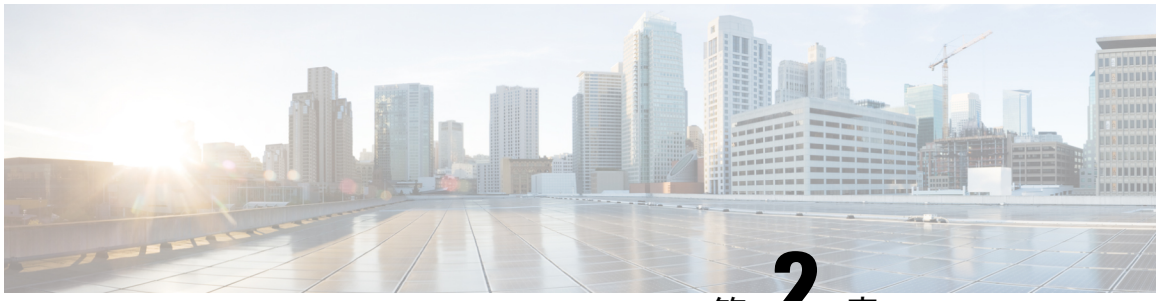
機能	リリースとプラットフォーム
アプリケーションホスティング：前面パネルのネットワークポートアクセス	Cisco IOS XE Gibraltar 16.12.1 • Cisco Catalyst 9300 シリーズ スイッチ Cisco IOS XE Amsterdam 17.1.1 • Cisco Catalyst 9400 シリーズ スイッチ （注） Cisco Catalyst 9410R スイッチは、前面パネルのアプリケーションホスティングをサポートしていません。
アプリケーションホスティング：前面パネルのUSBポートアクセス	Cisco IOS XE Gibraltar 16.12.1 • Cisco Catalyst 9300 シリーズ スイッチ Cisco IOS XE Amsterdam 17.1.1 • Cisco Catalyst 9400 シリーズ スイッチ （注） Cisco Catalyst 9410R スイッチは、前面パネルのアプリケーションホスティングをサポートしていません。
ネイティブDockerコンテナ：アプリケーションの自動再起動	Cisco IOS XE Amsterdam 17.2.1 • Cisco Catalyst 9300 シリーズ スイッチ Cisco IOS XE Bengaluru 17.5.1 • Cisco Catalyst 9410 シリーズ スイッチ
アプリケーションホスティング：ThousandEyes の統合	Cisco IOS XE Amsterdam 17.3.3 • Cisco Catalyst 9300 および 9300L シリーズ スイッチ Cisco IOS XE Bengaluru 17.5.1 • Cisco Catalyst 9400 シリーズ スイッチ Cisco IOS XE Bengaluru 17.6.1 • Cisco Catalyst 9300X シリーズ スイッチ
ThousandEyes BrowserBot	Cisco IOS XE Bengaluru 17.6.1 • Cisco Catalyst 9300、9300L、および 9300X シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ

機能	リリースとプラットフォーム
OpenFlow	
OpenFlow	Cisco IOS XE Fuji 16.9.1 • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチおよび Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ
OpenFlow Power over Ethernet	Cisco IOS XE Gibraltar 16.12.1 • Catalyst 9300 シリーズ スイッチ • Catalyst 9400 シリーズ スイッチ
OpenFlow フィールド書き換え	Cisco IOS XE Bengaluru 17.4.1 • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチおよび Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ
オープンフローモードでのハイアベイラビリティ	Cisco IOS XE Bengaluru 17.5.1 • Cisco Catalyst 9400 シリーズ スイッチ

新機能および変更された機能に関する情報

第 I 部

プロビジョニング



第 2 章

ゼロ タッチ プロビジョニング

ネットワーク プロビジョニングの課題に対応するため、シスコは、ゼロ タッチ プロビジョニング モデルを導入しました。このモジュールでは、ゼロ タッチ プロビジョニング機能について説明します。

（注） ゼロ タッチ プロビジョニング機能は自動的に有効になり、設定は不要です。

ゼロタッチプロビジョニングの制約事項

ゼロタッチプロビジョニングは、Cisco Catalyst 9200L SKU ではサポートされていません。

ゼロ タッチ プロビジョニングについて

ゼロ タッチ プロビジョニングの概要

ゼロ タッチ プロビジョニングは、異機種混在ネットワーク環境でのネットワーク デバイス プロビジョニングを自動化する、オープン ブートストラップ インターフェイスを提供します。

ゼロタッチプロビジョニングをサポートするデバイスが起動し、スタートアップ コンフィギュレーションが見つからない場合（初期インストール時）、デバイスはゼロ タッチ プロビジョニング モードに入ります。デバイスは、Dynamic Host Control Protocol（DHCP）サーバを検索し、インターフェイスの IP アドレス、ゲートウェイ、ドメイン ネーム システム（DNS）サーバの IP アドレスをブートストラップして、ゲスト シェルを有効にします。次にデバイスは

HTTP/TFTP サーバの IP アドレスまたは URL を取得し、HTTP/TFTP サーバからデバイスを構成する Python スクリプトをダウンロードします。

ゲスト シェルは、Python スクリプトを実行するための環境を提供します。ゲスト シェルは、ダウンロードした Python スクリプトを実行して、初期構成をデバイスに適用します。

初期プロビジョニングが完了したら、ゲスト シェルは有効化されたままになります。詳細については、「ゲスト シェル」の章を参照してください。

（注） ゼロ タッチ プロビジョニングが失敗した場合、デバイスは自動インストールにフォールバックして、コンフィギュレーション ファイルをロードします。詳細については、「Using AutoInstall and Setup」を参照してください。

ゼロ タッチ プロビジョニングのための DHCP サーバの設定

ゼロ タッチプロビジョニングでは、プロビジョニングされる新しいデバイスと同じネットワークで DHCP サーバを実行する必要があります。ゼロ タッチ プロビジョニングは、管理用ポートとインバンド ポートの両方でサポートされます。

新しいデバイスをオンにすると、そのデバイスは、Python スクリプトが存在する HTTP/TFTP サーバの IP アドレス情報と Python スクリプトのフォルダ パスを DHCP サーバから取得します。Python スクリプトの詳細については、「Python API」および「Python CLI モジュール」の各章を参照してください。

DHCP サーバは、次のオプションで DHCP 検出イベントに応答します。

- オプション 150：（任意）管理ネットワーク上の、実行される Python スクリプトをホストしている HTTP/TFTP サーバを指す IP アドレスの一覧が含まれます。
- オプション 67：HTTP/TFTP サーバ上の Python スクリプトのファイル パスが含まれます。

これらの DHCP オプションを受信すると、デバイスは、HTTP/TFTP サーバに接続して Python スクリプトをダウンロードします。この時点で、デバイスはHTTP/TFTP サーバに到達するルートを持たないため、DHCP サーバによって提供されるデフォルトのルートを使用します。

DHCPv6 のサポート

Cisco IOS XE Fuji 16.9.1 では、Dynamic Host Control Protocol バージョン 6（DHCPv6）のサポートがゼロタッチプロビジョニング機能に追加されました。DHCPv6 はデフォルトで有効になっており、スタートアップ コンフィギュレーションなしでブートするすべてのデバイスで機能します。

（注） DHCPv6 は Catalyst 9300 および 9500 シリーズ スイッチでのみサポートされます。

DHCPv6 は、Python スクリプトの TFTP と HTTP の両方のダウンロードによってサポートされています。Python スクリプトの HTTP または TFTP のダウンロードが失敗した場合、デバイスは開始時点（設定なしの状態）に戻ります。DHCPv4 と DHCPv6 の両方が機能するためには、正しい HTTP ファイル パスが DHCP 設定で使用できる必要があります。

同じインターフェイスに IPv4 と IPv6 の両方のアドレスがあるか、またはネットワーク内に 2 つの異なるインターフェイスがあることが考えられます。つまり、一方は IPv4 トラフィックを受信でき、他方は IPv6 トラフィックを受信できます。導入環境では DHCPv4 または DHCPv6 オプションのいずれかを使用することをお勧めします。

次に、DHCPv4: /etc/dhcp/dhcpd.conf の例を示します。

```
host <hostname> {
  hardware ethernet xx:xx:xx:xx:xx:xx;
  option dhcp-client-identifier "xxxxxxxxxxxxx";
  option host-name "<hostname>".
  option log-servers x.x.x.x;
  fixed-address x.x.x.x;
  if option vendor-class-identifier = "..." {
    option vendor-class-identifier "...";
    if exists user-class and option user-class = "iPXE" {
      filename "http://x.x.x.x/…/<image>";
    } else {
      filename "http://x.x.x.x/…/<script-name>";
    }
  }
}
```

次に、ISC DHCPv6 サーバの設定例を示します。

```
 option dhcp6.bootfile-url  "http://[2001:DB8::21]/sample_day0_script.py";
```

ゼロ タッチ プロビジョニングの構成例

TFTP コピーを使用しての管理ポートにおける DHCP サーバ設定の例

次に、デバイスの管理ポート経由で接続されている場合に TFTP コピーを使用して行う DHCP サーバ設定の例を示します。

```
Device> enable
Device# configure terminal
Device(config)# ip dhcp excluded-address 10.1.1.1
Device(config)# ip dhcp excluded-address vrf Mgmt-vrf 10.1.1.1 10.1.1.10
Device(config)# ip dhcp pool pnp_device_pool
Device(config-dhcp)# vrf Mgmt-vrf
Device(config-dhcp)# network 10.1.1.0 255.255.255.0
Device(config-dhcp)# default-router 10.1.1.1
Device(config-dhcp)# option 150 ip 203.0.113.254
Device(config-dhcp)# option 67 ascii /sample_python_dir/python_script.py
Device(config-dhcp)# exit
Device(config)# interface gigabitethernet 1/0/2
Device(config-if)# no ip dhcp client request tftp-server-address
```

```
Device(config-if)# end
```

HTTP コピーを使用しての管理ポートにおける DHCP サーバ設定の例

次に、デバイスの管理ポート経由で接続されている場合に HTTP コピーを使用して行う DHCP サーバ設定の例を示します。

```
Device> enable
Device# configure terminal
Device(config)# ip dhcp pool pnp_device_pool
Device(config-dhcp)# vrf Mgmt-vrf
Device(config-dhcp)# network 10.1.1.0 255.255.255.0
Device(config-dhcp)# default-router 10.1.1.1
Device(config-dhcp)# option 67 ascii http://198.51.100.1:8000/sample_python_2.py
Device(config-dhcp)# end
```

TFTP コピーを使用したインバンド ポートでのサンプル DHCP サーバ構成

次に示すのは、デバイスのインバンド ポート経由で接続されている場合の、TFTP コピーを使用したサンプル DHCP サーバ構成です。

```
Device> enable
Device# configure terminal
Device(config)# ip dhcp excluded-address 10.1.1.1
Device(config)# ip dhcp pool pnp_device_pool
Device(config-dhcp)# network 10.1.1.0 255.255.255.0
Device(config-dhcp)# default-router 10.1.1.1
Device(config-dhcp)# option 150 ip 203.0.113.254
Device(config-dhcp)# option 67 ascii /sample_python_dir/python_script.py
Device(config-dhcp)# exit
Device(config)# interface gigabitethernet 1/0/2
Device(config-if)# no ip dhcp client request tftp-server-address
Device(config-if)# end
```

HTTP コピーを使用したインバンド ポートでのサンプル DHCP サーバ構成

次に示すのは、デバイスのインバンド ポート経由で接続されている場合の、HTTP コピーを使用したサンプル DHCP サーバ構成です。

```
Device> enable
Device# configure terminal
Device(config)# ip dhcp excluded-address 10.1.1.1
Device(config)# ip dhcp pool pnp_device_pool
Device(config-dhcp)# network 10.1.1.0 255.255.255.0
```

```
Device(config-dhcp)# default-router 10.1.1.1
Device(config-dhcp)# option 67 ascii http://192.0.2.1:8000/sample_python_2.py
Device(config-dhcp)# end
```

Linux Ubuntu デバイス上でのサンプル DHCP サーバの構成

次の DHCP サーバ構成例は、サーバがデバイスの管理ポートまたはインバンド ポートのどちらかに接続されていることと、Python スクリプトが TFTP サーバからコピーされることを示しています。

```
root@ubuntu-server:/etc/dhcp# more dhcpd.conf
subnet 10.1.1.0 netmask 255.255.255.0 {
range 10.1.1.2  10.1.1.255;
      host 3850 {
          fixed-address                      10.1.1.246 ;
          hardware ethernet                  CC:D8:C1:85:6F:00;
          option bootfile-name !<opt 67>     " /python_dir/python_script.py";
         option tftp-server-name !<opt 150>  "203.0.113.254";
      }
  }
```

次のサンプル DHCP 構成は、Python スクリプトが HTTP サーバからデバイスにコピーされることを示しています。

```
Day0_with_mgmt_port_http
------------------------
subnet 192.168.1.0 netmask 255.255.255.0 {
 range 192.168.1.2 192.168.1.255;
      host C2-3850 {
          fixed-address                   192.168.1.246 ;
          hardware ethernet               CC:D8:C1:85:6F:00;
          option bootfile-name            "http://192.168.1.46/sample_python_2.py";
      }
  }
```

DHCP サーバが実行状態になったら、管理ネットワーク接続デバイスを起動します。これにより構成の残りの部分は自動的に実行されます。

TFTP コピーを使用する管理ポートでの DHCPv6 サーバ設定の例

次に、デバイスの管理ポート経由で接続されている場合に TFTP コピーを使用して行う DHCPv6 サーバ設定の例を示します。

```
Device> enable
Device# configure terminal
Device(config)# ipv6 dhcp pool ztp
Device(config-dhcpv6)# address prefix 2001:DB8::1/64
Device(config-dhcpv6)# domain-name cisco.com
Device(config-dhcpv6)# bootfile-url tftp://[2001:db8::46]/sample_day0_script.py
```

```
Device(config-dhcpv6)# exit
Device(config)# interface vlan 20
Device(config-if)# ipv6 dhcp server ztp
Device(config-if)# end
```

サンプルの Python プロビジョニング スクリプト

次に示すのは、HTTP サーバまたは TFTP サーバのいずれかから使用できるサンプル Python スクリプトです。

```
print "\n\n *** Sample ZTP Day0 Python Script *** \n\n"

# Importing cli module
import cli


print "\n\n *** Executing show platform  *** \n\n"
cli_command = "show platform"
cli.executep(cli_command)

print "\n\n *** Executing show version *** \n\n"
cli_command = "show version"
cli.executep(cli_command)

print "\n\n *** Configuring a Loopback Interface *** \n\n"
cli.configurep(["interface loop 100", "ip address 10.10.10.10 255.255.255.255", "end"])


print "\n\n *** Executing show ip interface brief  *** \n\n"
cli_command = "sh ip int brief"
cli.executep(cli_command)

print "\n\n *** ZTP Day0 Python Script Execution Complete *** \n\n"
```

Cisco 4000 シリーズ サービス統合型ルータの起動ログ

次のゼロ タッチ プロビジョニングのブート ログでは、ゲスト シェルが正常に有効にされ、Python スクリプトがゲスト シェルにダウンロードされ、ゲスト シェルがダウンロードした Python スクリプトを実行してデバイスをデイ ゼロに設定していることが示されています。

```
% failed to initialize nvram
! <This message indicates that the startup configuration
is absent on the device. This is the first indication that the Day Zero work flow is
going to start.>

This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
```

```
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

cisco ISR4451-X/K9 (2RU) processor with 7941237K/6147K bytes of memory.
Processor board ID FJC1950D091
4 Gigabit Ethernet interfaces
32768K bytes of non-volatile configuration memory.
16777216K bytes of physical memory.
7341807K bytes of flash memory at bootflash:.
0K bytes of WebUI ODM Files at webui:.

%INIT: waited 0 seconds for NVRAM to be available



         --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: %
!!<DO NOT TOUCH. This is Zero-Touch Provisioning>>
Generating 2048 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 1 seconds)
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
Guestshell enabled successfully


 *** Sample ZTP Day0 Python Script ***


 *** Configuring a Loopback Interface ***


Line 1 SUCCESS: interface loop 100
Line 2 SUCCESS: ip address 10.10.10.10 255.255.255.255
Line 3 SUCCESS: end


 *** Executing show ip interface brief  ***


Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0/0   unassigned      YES unset  down                  down
GigabitEthernet0/0/1   unassigned      YES unset  down                  down
GigabitEthernet0/0/2   unassigned      YES unset  down                  down
GigabitEthernet0/0/3   192.168.1.246   YES DHCP   up                    up
GigabitEthernet0       192.168.1.246   YES DHCP   up                    up
Loopback100            10.10.10.10     YES TFTP   up                    up


 *** ZTP Day0 Python Script Execution Complete ***
```

```
Press RETURN to get started!
```

デイ ゼロ プロビジョニングが完了すると、IOS プロンプトがアクセス可能になります。

Cisco Catalyst 9000 シリーズ スイッチの起動ログ

次のセクションでは、ゼロタッチプロビジョニングの起動ログのサンプルを表示します。このようなログでは、ゲストシェルが正常に有効にされ、Python スクリプトがゲストシェルにダウンロードされ、ゲストシェルがダウンロードした Python スクリプトを実行してデバイスをデイゼロに設定していることが示されています。

```
% Checking backup nvram
% No config present. Using default config

FIPS: Flash Key Check : Begin
FIPS: Flash Key Check : End, Not Found, FIPS Mode Not Enabled

! <This message indicates that the startup configuration
is absent on the device. This is the first indication that the Day Zero
 work flow is
going to start.>
```

Cisco IOS XE Everest 16.6.x から Cisco IOS XE Fuji 16.8.x へ

このセクションでは、.py スクリプトを実行する前の起動ログのサンプルを表示します。

```
Press RETURN to get started!

The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable


 *** Sample ZTP Day0 Python Script ***

...

 *** ZTP Day0 Python Script Execution Complete ***
```

このセクションでは、デイゼロプロビジョニング用にデバイスを設定する方法を示します。

```
Initializing Hardware...

System Bootstrap, Version 17.2.1r[FC1], RELEASE SOFTWARE (P)
Compiled Thu 02/20/2020 23:47:51.50 by rel
```

```
Current ROMMON image : Primary
Last reset cause     : SoftwareReload
C9300-48UXM platform with 8388608 Kbytes of main memory

Preparing to autoboot. [Press Ctrl-C to interrupt]  0
boot: attempting to boot from [flash:cat9k_iosxe.16.06.05.SPA.bin]
boot: reading file cat9k_iosxe.16.06.05.SPA.bin
###############################################################################################


Both links down, not waiting for other switches
Switch number is 1

              Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

           cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706



Cisco IOS Software [Everest], Catalyst L3 Switch Software (CAT9K_IOSXE),
Version 16.6.5, RELEASE SOFTWARE (fc3)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2018 by Cisco Systems, Inc.
Compiled Mon 10-Dec-18 12:52 by mcpre



Cisco IOS-XE software, Copyright (c) 2005-2018 by cisco Systems, Inc.
All rights reserved.  Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0.  The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY.  You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0.  For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.


% Checking backup nvram
% No config present. Using default config


FIPS: Flash Key Check : Begin
FIPS: Flash Key Check : End, Not Found, FIPS Mode Not Enabled

This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
```

```
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

cisco C9300-48UXM (X86) processor with 1392780K/6147K bytes of memory.
Processor board ID FCW2144L045
2048K bytes of non-volatile configuration memory.
8388608K bytes of physical memory.
1638400K bytes of Crash Files at crashinfo:.
11264000K bytes of Flash at flash:.
0K bytes of WebUI ODM Files at webui:.

Base Ethernet MAC Address          : ec:1d:8b:0a:68:00
Motherboard Assembly Number        : 73-17959-06
Motherboard Serial Number          : FOC21418FPQ
Model Revision Number              : B0
Motherboard Revision Number        : A0
Model Number                       : C9300-48UXM
System Serial Number               : FCW2144L045


%INIT: waited 0 seconds for NVRAM to be available

SETUP: new interface Vlan1 placed in "shutdown" state


Press RETURN to get started!


*Sep  4 20:35:07.330: %SMART_LIC-6-AGENT_READY: Smart Agent for Licensing is initialized
*Sep  4 20:35:07.493: %IOSXE_RP_NV-3-NV_ACCESS_FAIL: Initial read of NVRAM contents
failed
*Sep  4 20:35:07.551: %IOSXE_RP_NV-3-BACKUP_NV_ACCESS_FAIL: Initial read of backup NVRAM
 contents failed
*Sep  4 20:35:10.932: dev_pluggable_optics_selftest attribute table internally inconsistent
 @ 0x1D4

*Sep  4 20:35:13.406: %CRYPTO-4-AUDITWARN: Encryption audit check could not be performed
*Sep  4 20:35:13.480: %SPANTREE-5-EXTENDED_SYSID: Extended SysId enabled for type vlan
*Sep  4 20:35:13.715: %LINK-3-UPDOWN: Interface Lsmpi18/3, changed state to up
*Sep  4 20:35:13.724: %LINK-3-UPDOWN: Interface EOBC18/1, changed state to up
*Sep  4 20:35:13.724: %LINEPROTO-5-UPDOWN: Line protocol on Interface LI-Null0, changed
 state to up
*Sep  4 20:35:13.724: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to
down
*Sep  4 20:35:13.725: %LINK-3-UPDOWN: Interface LIIN18/2, changed state to up
*Sep  4 20:35:13.749: WCM-PKI-SHIM: buffer allocation failed for SUDI support check
*Sep  4 20:35:13.749: PKI/SSL unable to send Sudi support to WCM
*Sep  4 20:35:14.622: %IOSXE_MGMTVRF-6-CREATE_SUCCESS_INFO: Management vrf Mgmt-vrf
created with ID 1,
    ipv4 table-id 0x1, ipv6 table-id 0x1E000001
*Sep  4 20:34:42.022: %STACKMGR-6-STACK_LINK_CHANGE: Switch 1 R0/0: stack_mgr:  Stack
port 1 on Switch 1 is nocable
*Sep  4 20:34:42.022: %STACKMGR-6-STACK_LINK_CHANGE: Switch 1 R0/0: stack_mgr:  Stack
port 2 on Switch 1 is down
*Sep  4 20:34:42.022: %STACKMGR-6-STACK_LINK_CHANGE: Switch 1 R0/0: stack_mgr:  Stack
port 2 on Switch 1 is nocable
*Sep  4 20:34:42.022: %STACKMGR-6-SWITCH_ADDED: Switch 1 R0/0: stack_mgr:  Switch 1 has
 been added to the stack.
*Sep  4 20:34:42.022: %STACKMGR-6-SWITCH_ADDED: Switch 1 R0/0: stack_mgr:  Switch 1 has
 been added to the stack.
*Sep  4 20:34:42.022: %STACKMGR-6-SWITCH_ADDED: Switch 1 R0/0: stack_mgr:  Switch 1 has
 been added to the stack.
```

```
*Sep  4 20:34:42.022: %STACKMGR-6-ACTIVE_ELECTED: Switch 1 R0/0: stack_mgr:  Switch 1
has been elected ACTIVE.
*Sep  4 20:35:14.728: %LINEPROTO-5-UPDOWN: Line protocol on Interface Lsmpi18/3, changed
 state to up
*Sep  4 20:35:14.728: %LINEPROTO-5-UPDOWN: Line protocol on Interface EOBC18/1, changed
 state to up
*Sep  4 20:35:15.506: %HMANRP-6-HMAN_IOS_CHANNEL_INFO: HMAN-IOS channel event for switch
 1: EMP_RELAY: Channel UP!
*Sep  4 20:35:15.510: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed
state to down
*Sep  4 20:35:34.501: %LINK-5-CHANGED: Interface Vlan1, changed state to administratively
 down
*Sep  4 20:35:34.717: %SYS-5-RESTART: System restarted --
Cisco IOS Software [Everest], Catalyst L3 Switch Software (CAT9K_IOSXE), Version 16.6.5,
 RELEASE SOFTWARE (fc3)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2018 by Cisco Systems, Inc.
Compiled Mon 10-Dec-18 12:52 by mcpre
*Sep  4 20:35:34.796: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
*Sep  4 20:35:35.266: %SYS-6-BOOTTIME: Time taken to reboot after reload =  283 seconds
*Sep  4 20:35:35.796: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0,
 changed state to up
*Sep  4 20:35:36.607: %LINK-3-UPDOWN: Interface GigabitEthernet1/1/1, changed state to
down
*Sep  4 20:35:36.607: %LINK-3-UPDOWN: Interface GigabitEthernet1/1/2, changed state to
down
*Sep  4 20:35:36.607: %LINK-3-UPDOWN: Interface GigabitEthernet1/1/3, changed state to
down
*Sep  4 20:35:36.608: %LINK-3-UPDOWN: Interface GigabitEthernet1/1/4, changed state to
down
*Sep  4 20:35:36.608: %LINK-3-UPDOWN: Interface TenGigabitEthernet1/1/1, changed state
to down
*Sep  4 20:35:36.608: %LINK-3-UPDOWN: Interface TenGigabitEthernet1/1/2, changed state
to down
*Sep  4 20:35:36.608: %LINK-3-UPDOWN: Interface TenGigabitEthernet1/1/3, changed state
to down
*Sep  4 20:35:36.608: %LINK-3-UPDOWN: Interface TenGigabitEthernet1/1/4, changed state
to down
*Sep  4 20:35:36.608: %LINK-3-UPDOWN: Interface TenGigabitEthernet1/1/5, changed state
to down
*Sep  4 20:35:36.609: %LINK-3-UPDOWN: Interface TenGigabitEthernet1/1/6, changed state
to down
*Sep  4 20:35:36.609: %LINK-3-UPDOWN: Interface TenGigabitEthernet1/1/7, changed state
to down
*Sep  4 20:35:36.609: %LINK-3-UPDOWN: Interface TenGigabitEthernet1/1/8, changed state
to down
*Sep  4 20:35:36.609: %LINK-3-UPDOWN: Interface FortyGigabitEthernet1/1/1, changed state
 to down
*Sep  4 20:35:36.609: %LINK-3-UPDOWN: Interface FortyGigabitEthernet1/1/2, changed state
 to down
*Sep  4 20:35:37.607: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/1/1,
 changed state to down
*Sep  4 20:35:37.608: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/1/2,
 changed state to down
*Sep  4 20:35:37.608: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/1/3,
 changed state to down
*Sep  4 20:35:37.609: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/1/4,
 changed state to down
*Sep  4 20:35:37.609: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet1/1/1, changed state to down
*Sep  4 20:35:37.609: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet1/1/2, changed state to down
*Sep  4 20:35:37.609: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet1/1/3, changed state to down
```

```
*Sep  4 20:35:37.609: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet1/1/4, changed state to down
*Sep  4 20:35:37.609: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet1/1/5, changed state to down
*Sep  4 20:35:37.609: %LINEPROTO-5-UPDOWN: Line protocol on Interface
TenGigabitEthernet1/1/6, changed state to down
*Sep  4 20:35:43.511: AUTOINSTALL: Obtain tftp server address (opt 150) 159.14.27.2
*Sep  4 20:35:43.511: PNPA: Setting autoinstall complete to true for 159.14.27.2
*Sep  4 20:35:57.673: %PLATFORM_PM-6-FRULINK_INSERTED: 8x10G uplink module inserted in
the switch 1 slot 1
*Sep  4 20:36:19.562: [IOX DEBUG] Guestshell start API is being invoked

*Sep  4 20:36:19.562: [IOX DEBUG] provided idb is mgmt interface

*Sep  4 20:36:19.562: [IOX DEBUG] Setting up guestshell to use mgmt-intf

*Sep  4 20:36:19.562: [IOX DEBUG] Setting up chasfs for iox related activity

*Sep  4 20:36:19.562: [IOX DEBUG] Setting up for iox pre-clean activity if needed

*Sep  4 20:36:19.562: [IOX DEBUG] Waiting for iox pre-clean setup to take affect

*Sep  4 20:36:19.562: [IOX DEBUG] Waited for 1 sec(s) for iox pre-clean setup to take
affect

*Sep  4 20:36:19.562: [IOX DEBUG] Auto-configuring iox feature

*Sep  4 20:36:19.563: [IOX DEBUG] Waiting for CAF and ioxman to be up, in that order

*Sep  4 20:36:20.076: %UICFGEXP-6-SERVER_NOTIFIED_START: Switch 1 R0/0: psd:  Server iox
 has been notified to start
*Sep  4 20:36:23.564: [IOX DEBUG] Waiting for another 5 secs

*Sep  4 20:36:28.564: [IOX DEBUG] Waiting for another 5 secs
The process for the command is not responding or is otherwise unavailable

*Sep  4 20:36:33.564: [IOX DEBUG] Waiting for another 5 secs
The process for the command is not responding or is otherwise unavailable

*Sep  4 20:36:34.564: [IOX DEBUG] Waited for 16 sec(s) for CAF and ioxman to come up

*Sep  4 20:36:34.564: [IOX DEBUG] Validating if CAF and ioxman are running

*Sep  4 20:36:34.564: [IOX DEBUG] CAF and ioxman are up and running

*Sep  4 20:36:34.564: [IOX DEBUG] Building the simple mgmt-intf enable command string

*Sep  4 20:36:34.564: [IOX DEBUG] Enable command is: request platform software iox-manager

    app-hosting guestshell enable

*Sep  4 20:36:34.564: [IOX DEBUG] Issuing guestshell enable command and waiting for it
to be up
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable

*Sep  4 20:36:38.578: [IOX DEBUG] Waiting for another 5 secs
The process for the command is not responding or is otherwise unavailable

*Sep  4 20:36:39.416: %LINK-3-UPDOWN: Interface TenGigabitEthernet1/0/48, changed state
 to up
*Sep  4 20:36:40.416: %LINEPROTO-5-UPDOWN: Line protocol on Interface
```

```
TenGigabitEthernet1/0/48,
    changed state to upThe process for the command is not responding or is otherwise
unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable

*Sep  4 20:36:43.586: [IOX DEBUG] Waiting for another 5 secs
Guestshell enabled successfully

*Sep  4 20:37:45.321: [IOX DEBUG] Checking for guestshell mount path

*Sep  4 20:37:45.321: [IOX DEBUG] Validating if guestshell is ready for use

*Sep  4 20:37:45.321: [IOX DEBUG] Guestshell enabled successfully


 *** Sample ZTP Day0 Python Script ***



 *** Executing show platform  ***


Switch  Ports    Model                Serial No.   MAC address     Hw Ver.       Sw Ver.

------  -----   ---------            -----------  --------------  -------       --------

 1       62     C9300-48UXM           FCW2144L045  ec1d.8b0a.6800  V01           16.6.5

Switch/Stack Mac Address : ec1d.8b0a.6800 - Local Mac Address
Mac persistency wait time: Indefinite
                                     Current
Switch#   Role        Priority      State
-------------------------------------------
*1       Active          1          Ready


 *** Executing show version ***


Cisco IOS XE Software, Version 16.06.05
Cisco IOS Software [Everest], Catalyst L3 Switch Software (CAT9K_IOSXE), Version 16.6.5,
 RELEASE SOFTWARE (fc3)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2018 by Cisco Systems, Inc.
Compiled Mon 10-Dec-18 12:52 by mcpre
Cisco IOS-XE software, Copyright (c) 2005-2018 by cisco Systems, Inc.
All rights reserved.  Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0.  The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY.  You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0.  For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.
ROM: IOS-XE ROMMON
BOOTLDR: System Bootstrap, Version 17.2.1r[FC1], RELEASE SOFTWARE (P)
Switch uptime is 2 minutes
Uptime for this control processor is 4 minutes
System returned to ROM by Reload Command
System image file is "flash:cat9k_iosxe.16.06.05.SPA.bin"
Last reload reason: Reload Command
This product contains cryptographic features and is subject to United
```

```
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.
A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html
If you require further assistance please contact us by sending email to
export@cisco.com.
Technology Package License Information:
------------------------------------------------------------------
Technology-package                  Technology-package
Current             Type            Next reboot
------------------------------------------------------------------
network-advantage   Permanent       network-advantage
cisco C9300-48UXM (X86) processor with 1392780K/6147K bytes of memory.
Processor board ID FCW2144L045
36 Ethernet interfaces
1 Virtual Ethernet interface
4 Gigabit Ethernet interfaces
20 Ten Gigabit Ethernet interfaces
2 Forty Gigabit Ethernet interfaces
2048K bytes of non-volatile configuration memory.
8388608K bytes of physical memory.
1638400K bytes of Crash Files at crashinfo:.
11264000K bytes of Flash at flash:.
0K bytes of WebUI ODM Files at webui:.
Base Ethernet MAC Address          : ec:1d:8b:0a:68:00
Motherboard Assembly Number        : 73-17959-06
Motherboard Serial Number          : FOC21418FPQ
Model Revision Number              : B0
Motherboard Revision Number        : A0
Model Number                       : C9300-48UXM
System Serial Number               : FCW2144L045
Switch Ports Model              SW Version        SW Image              Mode
------ ----- -----              ----------        ----------            ----
*    1 62    C9300-48UXM        16.6.5            CAT9K_IOSXE           BUNDLE
Configuration register is 0x102


 *** Configuring a Loopback Interface ***


Line 1 SUCCESS: interface loop 100
Line 2 SUCCESS: ip address 10.10.10.10 255.255.255.255
Line 3 SUCCESS: end


 *** Executing show ip interface brief  ***


Interface              IP-Address      OK? Method Status                Protocol
Vlan1                  unassigned      YES unset  administratively down down
GigabitEthernet0/0     10.127.128.3    YES DHCP   up                    up
Tw1/0/1                unassigned      YES unset  down                  down
Tw1/0/2                unassigned      YES unset  down                  down
Tw1/0/3                unassigned      YES unset  down                  down
Tw1/0/4                unassigned      YES unset  down                  down
Tw1/0/5                unassigned      YES unset  down                  down
Tw1/0/6                unassigned      YES unset  down                  down
Tw1/0/7                unassigned      YES unset  down                  down
Tw1/0/8                unassigned      YES unset  down                  down
```

```
Tw1/0/9                unassigned      YES unset  down                  down
Tw1/0/10               unassigned      YES unset  down                  down
Tw1/0/11               unassigned      YES unset  down                  down
Tw1/0/12               unassigned      YES unset  down                  down
Tw1/0/13               unassigned      YES unset  down                  down
Tw1/0/14               unassigned      YES unset  down                  down
Tw1/0/15               unassigned      YES unset  down                  down
Tw1/0/16               unassigned      YES unset  down                  down
Tw1/0/17               unassigned      YES unset  down                  down
Tw1/0/18               unassigned      YES unset  down                  down
Tw1/0/19               unassigned      YES unset  down                  down
Tw1/0/20               unassigned      YES unset  down                  down
Tw1/0/21               unassigned      YES unset  down                  down
Tw1/0/22               unassigned      YES unset  down                  down
Tw1/0/23               unassigned      YES unset  down                  down
Tw1/0/24               unassigned      YES unset  down                  down
Tw1/0/25               unassigned      YES unset  down                  down
Tw1/0/26               unassigned      YES unset  down                  down
Tw1/0/27               unassigned      YES unset  down                  down
Tw1/0/28               unassigned      YES unset  down                  down
Tw1/0/29               unassigned      YES unset  down                  down
Tw1/0/30               unassigned      YES unset  down                  down
Tw1/0/31               unassigned      YES unset  down                  down
Tw1/0/32               unassigned      YES unset  down                  down
Tw1/0/33               unassigned      YES unset  down                  down
Tw1/0/34               unassigned      YES unset  down                  down
Tw1/0/35               unassigned      YES unset  down                  down
Tw1/0/36               unassigned      YES unset  down                  down
Te1/0/37               unassigned      YES unset  down                  down
Te1/0/38               unassigned      YES unset  down                  down
Te1/0/39               unassigned      YES unset  down                  down
Te1/0/40               unassigned      YES unset  down                  down
Te1/0/41               unassigned      YES unset  down                  down
Te1/0/42               unassigned      YES unset  down                  down
Te1/0/43               unassigned      YES unset  down                  down
Te1/0/44               unassigned      YES unset  down                  down
Te1/0/45               unassigned      YES unset  down                  down
Te1/0/46               unassigned      YES unset  down                  down
Te1/0/47               unassigned      YES unset  down                  down
Te1/0/48               unassigned      YES unset  up                    up
GigabitEthernet1/1/1   unassigned      YES unset  down                  down
GigabitEthernet1/1/2   unassigned      YES unset  down                  down
GigabitEthernet1/1/3   unassigned      YES unset  down                  down
GigabitEthernet1/1/4   unassigned      YES unset  down                  down
Te1/1/1                unassigned      YES unset  down                  down
Te1/1/2                unassigned      YES unset  down                  down
Te1/1/3                unassigned      YES unset  down                  down
Te1/1/4                unassigned      YES unset  down                  down
Te1/1/5                unassigned      YES unset  down                  down
Te1/1/6                unassigned      YES unset  down                  down
Te1/1/7                unassigned      YES unset  down                  down
Te1/1/8                unassigned      YES unset  down                  down
Fo1/1/1                unassigned      YES unset  down                  down
Fo1/1/2                unassigned      YES unset  down                  down
Loopback100            10.10.10.10     YES TFTP   up                    up


 *** Configuring username, password, SSH  ***


Line 1 SUCCESS: username cisco privilege 15 password cisco
Line 2 SUCCESS: ip domain name domain
Line 3 SUCCESS: line vty 0 15
Line 4 SUCCESS: login local
```

```
Line 5 SUCCESS: transport input all
Line 6 SUCCESS: end


 *** ZTP Day0 Python Script Execution Complete ***
```

Cisco IOS XE Fuji 16.9.x から Cisco IOS XE Gibraltar 16.11.x へ

このセクションでは、.py スクリプトを実行する前の起動ログのサンプルを表示します。

```
--- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: The process for the
 command is not
responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
guestshell installed successfully
Current state is: DEPLOYED
guestshell activated successfully
Current state is: ACTIVATED
guestshell started successfully
Current state is: RUNNING
Guestshell enabled successfully
```

このセクションでは、デイゼロプロビジョニング用にデバイスを設定する方法を示します。

```
Both links down, not waiting for other switches
Switch number is 1

              Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

           Cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706



Cisco IOS Software [Fuji], Catalyst L3 Switch Software (CAT9K_IOSXE), Version 16.9.4,
RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2019 by Cisco Systems, Inc.
Compiled Thu 22-Aug-19 18:14 by mcpre



PLEASE READ THE FOLLOWING TERMS CAREFULLY. INSTALLING THE LICENSE OR
LICENSE KEY PROVIDED FOR ANY CISCO SOFTWARE PRODUCT, PRODUCT FEATURE,
AND/OR SUBSEQUENTLY PROVIDED SOFTWARE FEATURES (COLLECTIVELY, THE
```

```
"SOFTWARE"), AND/OR USING SUCH SOFTWARE CONSTITUTES YOUR FULL
ACCEPTANCE OF THE FOLLOWING TERMS. YOU MUST NOT PROCEED FURTHER IF YOU
ARE NOT WILLING TO BE BOUND BY ALL THE TERMS SET FORTH HEREIN.

Your use of the Software is subject to the Cisco End User License Agreement
(EULA) and any relevant supplemental terms (SEULA) found at
http://www.cisco.com/c/en/us/about/legal/cloud-and-software/software-terms.html.

You hereby acknowledge and agree that certain Software and/or features are
licensed for a particular term, that the license to such Software and/or
features is valid only for the applicable term and that such Software and/or
features may be shut down or otherwise terminated by Cisco after expiration
of the applicable license term (e.g., 90-day trial period). Cisco reserves
the right to terminate any such Software feature electronically or by any
other means available. While Cisco may provide alerts, it is your sole
responsibility to monitor your usage of any such term Software feature to
ensure that your systems and networks are prepared for a shutdown of the
Software feature.


% Checking backup nvram
% No config present. Using default config

FIPS: Flash Key Check : Key Not Found, FIPS Mode Not Enabled
cisco C9300-48UXM (X86) processor with 1419044K/6147K bytes of memory.
Processor board ID FCW2144L045
2048K bytes of non-volatile configuration memory.
8388608K bytes of physical memory.
1638400K bytes of Crash Files at crashinfo:.
11264000K bytes of Flash at flash:.
0K bytes of WebUI ODM Files at webui:.

Base Ethernet MAC Address          : ec:1d:8b:0a:68:00
Motherboard Assembly Number        : 73-17959-06
Motherboard Serial Number          : FOC21418FPQ
Model Revision Number              : B0
Motherboard Revision Number        : A0
Model Number                       : C9300-48UXM
System Serial Number               : FCW2144L045


%INIT: waited 0 seconds for NVRAM to be available


         --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: The process for the
 command is not
    responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
```

```
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
guestshell installed successfully
Current state is: DEPLOYED
guestshell activated successfully
Current state is: ACTIVATED
guestshell started successfully
Current state is: RUNNING
Guestshell enabled successfully


HTTP server statistics:
Accepted connections total: 0

 *** Sample ZTP Day0 Python Script ***




 *** Executing show platform  ***


Switch  Ports    Model                Serial No.  MAC address     Hw Ver.       Sw Ver.

------  -----   ---------            -----------  --------------  -------      --------
```

```
 1       64     C9300-48UXM           FCW2144L045  ec1d.8b0a.6800  V01           16.9.4

Switch/Stack Mac Address : ec1d.8b0a.6800 - Local Mac Address
Mac persistency wait time: Indefinite
                                     Current
Switch#   Role        Priority      State
-----------------------------------------
*1       Active          1          Ready


 *** Executing show version ***


Cisco IOS XE Software, Version 16.09.04
Cisco IOS Software [Fuji], Catalyst L3 Switch Software (CAT9K_IOSXE), Version 16.9.4,
RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2019 by Cisco Systems, Inc.
Compiled Thu 22-Aug-19 18:14 by mcpre
Cisco IOS-XE software, Copyright (c) 2005-2019 by cisco Systems, Inc.
All rights reserved.  Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0.  The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY.  You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0.  For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.
ROM: IOS-XE ROMMON
BOOTLDR: System Bootstrap, Version 17.2.1r[FC1], RELEASE SOFTWARE (P)
Switch uptime is 4 minutes
Uptime for this control processor is 5 minutes
System returned to ROM by Reload Command
System image file is "flash:cat9k_iosxe.16.09.04.SPA.bin"
Last reload reason: Reload Command
This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.
A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html
If you require further assistance please contact us by sending email to
export@cisco.com.
Technology Package License Information:
------------------------------------------------------------------------------
Technology-package                                   Technology-package
Current                       Type                      Next reboot
------------------------------------------------------------------------------
network-advantage       Smart License                  network-advantage
None                    Subscription Smart License     None
Smart Licensing Status: UNREGISTERED/EVAL EXPIRED
cisco C9300-48UXM (X86) processor with 1419044K/6147K bytes of memory.
Processor board ID FCW2144L045
36 Ethernet interfaces
1 Virtual Ethernet interface
4 Gigabit Ethernet interfaces
20 Ten Gigabit Ethernet interfaces
2 TwentyFive Gigabit Ethernet interfaces
2 Forty Gigabit Ethernet interfaces
2048K bytes of non-volatile configuration memory.
```

```
8388608K bytes of physical memory.
1638400K bytes of Crash Files at crashinfo:.
11264000K bytes of Flash at flash:.
0K bytes of WebUI ODM Files at webui:.
Base Ethernet MAC Address          : ec:1d:8b:0a:68:00
Motherboard Assembly Number        : 73-17959-06
Motherboard Serial Number          : FOC21418FPQ
Model Revision Number              : B0
Motherboard Revision Number        : A0
Model Number                       : C9300-48UXM
System Serial Number               : FCW2144L045
Switch Ports Model              SW Version        SW Image              Mode
------ ----- -----              ----------        ----------            ----
*    1 64    C9300-48UXM        16.9.4            CAT9K_IOSXE           BUNDLE
Configuration register is 0x102


 *** Configuring a Loopback Interface ***


Line 1 SUCCESS: interface loop 100
Line 2 SUCCESS: ip address 10.10.10.10 255.255.255.255
Line 3 SUCCESS: end


 *** Executing show ip interface brief  ***


Any interface listed with OK? value "NO" does not have a valid configuration
Interface              IP-Address      OK? Method Status                Protocol
Vlan1                  unassigned      NO  unset  up                    up
GigabitEthernet0/0     10.127.128.5    YES DHCP   up                    up
Tw1/0/1                unassigned      YES unset  down                  down
Tw1/0/2                unassigned      YES unset  down                  down
Tw1/0/3                unassigned      YES unset  down                  down
Tw1/0/4                unassigned      YES unset  down                  down
Tw1/0/5                unassigned      YES unset  down                  down
Tw1/0/6                unassigned      YES unset  down                  down
Tw1/0/7                unassigned      YES unset  down                  down
Tw1/0/8                unassigned      YES unset  down                  down
Tw1/0/9                unassigned      YES unset  down                  down
Tw1/0/10               unassigned      YES unset  down                  down
Tw1/0/11               unassigned      YES unset  down                  down
Tw1/0/12               unassigned      YES unset  down                  down
Tw1/0/13               unassigned      YES unset  down                  down
Tw1/0/14               unassigned      YES unset  down                  down
Tw1/0/15               unassigned      YES unset  down                  down
Tw1/0/16               unassigned      YES unset  down                  down
Tw1/0/17               unassigned      YES unset  down                  down
Tw1/0/18               unassigned      YES unset  down                  down
Tw1/0/19               unassigned      YES unset  down                  down
Tw1/0/20               unassigned      YES unset  down                  down
Tw1/0/21               unassigned      YES unset  down                  down
Tw1/0/22               unassigned      YES unset  down                  down
Tw1/0/23               unassigned      YES unset  down                  down
Tw1/0/24               unassigned      YES unset  down                  down
Tw1/0/25               unassigned      YES unset  down                  down
Tw1/0/26               unassigned      YES unset  down                  down
Tw1/0/27               unassigned      YES unset  down                  down
Tw1/0/28               unassigned      YES unset  down                  down
Tw1/0/29               unassigned      YES unset  down                  down
Tw1/0/30               unassigned      YES unset  down                  down
Tw1/0/31               unassigned      YES unset  down                  down
Tw1/0/32               unassigned      YES unset  down                  down
```

```
Tw1/0/33               unassigned      YES unset  down                  down
Tw1/0/34               unassigned      YES unset  down                  down
Tw1/0/35               unassigned      YES unset  down                  down
Tw1/0/36               unassigned      YES unset  down                  down
Te1/0/37               unassigned      YES unset  down                  down
Te1/0/38               unassigned      YES unset  down                  down
Te1/0/39               unassigned      YES unset  down                  down
Te1/0/40               unassigned      YES unset  down                  down
Te1/0/41               unassigned      YES unset  down                  down
Te1/0/42               unassigned      YES unset  down                  down
Te1/0/43               unassigned      YES unset  down                  down
Te1/0/44               unassigned      YES unset  down                  down
Te1/0/45               unassigned      YES unset  down                  down
Te1/0/46               unassigned      YES unset  down                  down
Te1/0/47               unassigned      YES unset  down                  down
Te1/0/48               unassigned      YES unset  up                    up
GigabitEthernet1/1/1   unassigned      YES unset  down                  down
GigabitEthernet1/1/2   unassigned      YES unset  down                  down
GigabitEthernet1/1/3   unassigned      YES unset  down                  down
GigabitEthernet1/1/4   unassigned      YES unset  down                  down
Te1/1/1                unassigned      YES unset  down                  down
Te1/1/2                unassigned      YES unset  down                  down
Te1/1/3                unassigned      YES unset  down                  down
Te1/1/4                unassigned      YES unset  down                  down
Te1/1/5                unassigned      YES unset  down                  down
Te1/1/6                unassigned      YES unset  down                  down
Te1/1/7                unassigned      YES unset  down                  down
Te1/1/8                unassigned      YES unset  down                  down
Fo1/1/1                unassigned      YES unset  down                  down
Fo1/1/2                unassigned      YES unset  down                  down
TwentyFiveGigE1/1/1    unassigned      YES unset  down                  down
TwentyFiveGigE1/1/2    unassigned      YES unset  down                  down
Loopback100            10.10.10.10     YES TFTP   up                    up


 *** Configuring username, password, SSH  ***


Line 1 SUCCESS: username cisco privilege 15 password cisco
**CLI Line # 1:  WARNING: Command has been added to the configuration using a type 0
password.
    However, type 0 passwords will soon be deprecated. Migrate to a supported password
type
Line 2 SUCCESS: ip domain name domain
Line 3 SUCCESS: line vty 0 15
Line 4 SUCCESS: login local
Line 5 SUCCESS: transport input all
Line 6 SUCCESS: end


 *** ZTP Day0 Python Script Execution Complete ***




Press RETURN to get started!
```

Cisco IOS XE Gibraltar 16.12.x から Cisco IOS XE Amsterdam 17.1.x へ

このセクションでは、.py スクリプトを実行する前の起動ログのサンプルを表示します。

```
--- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: day0guestshell
installed successfully
Current state is: DEPLOYED
day0guestshell activated successfully
Current state is: ACTIVATED
day0guestshell started successfully
Current state is: RUNNING
Guestshell enabled successfully


 *** Sample ZTP Day0 Python Script ***

...

 *** ZTP Day0 Python Script Execution Complete ***


Guestshell destroyed successfully
```

このセクションでは、デイゼロプロビジョニング用にデバイスを設定する方法を示します。

```
Both links down, not waiting for other switches
Switch number is 1

               Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

           Cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706



Cisco IOS Software [Gibraltar], Catalyst L3 Switch Software (CAT9K_IOSXE), Version
16.12.3a,
RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2020 by Cisco Systems, Inc.
Compiled Tue 28-Apr-20 09:37 by mcpre


This software version supports only Smart Licensing as the software licensing mechanism.


PLEASE READ THE FOLLOWING TERMS CAREFULLY. INSTALLING THE LICENSE OR
LICENSE KEY PROVIDED FOR ANY CISCO SOFTWARE PRODUCT, PRODUCT FEATURE,
AND/OR SUBSEQUENTLY PROVIDED SOFTWARE FEATURES (COLLECTIVELY, THE
"SOFTWARE"), AND/OR USING SUCH SOFTWARE CONSTITUTES YOUR FULL
ACCEPTANCE OF THE FOLLOWING TERMS. YOU MUST NOT PROCEED FURTHER IF YOU
ARE NOT WILLING TO BE BOUND BY ALL THE TERMS SET FORTH HEREIN.
```

```
Your use of the Software is subject to the Cisco End User License Agreement
(EULA) and any relevant supplemental terms (SEULA) found at
http://www.cisco.com/c/en/us/about/legal/cloud-and-software/software-terms.html.

You hereby acknowledge and agree that certain Software and/or features are
licensed for a particular term, that the license to such Software and/or
features is valid only for the applicable term and that such Software and/or
features may be shut down or otherwise terminated by Cisco after expiration
of the applicable license term (e.g., 90-day trial period). Cisco reserves
the right to terminate any such Software feature electronically or by any
other means available. While Cisco may provide alerts, it is your sole
responsibility to monitor your usage of any such term Software feature to
ensure that your systems and networks are prepared for a shutdown of the
Software feature.


% Checking backup nvram
% No config present. Using default config

FIPS: Flash Key Check : Key Not Found, FIPS Mode Not Enabled

All TCP AO KDF Tests Pass
cisco C9300-48UXM (X86) processor with 1343703K/6147K bytes of memory.
Processor board ID FCW2144L045
2048K bytes of non-volatile configuration memory.
8388608K bytes of physical memory.
1638400K bytes of Crash Files at crashinfo:.
11264000K bytes of Flash at flash:.
0K bytes of WebUI ODM Files at webui:.

Base Ethernet MAC Address          : ec:1d:8b:0a:68:00
Motherboard Assembly Number        : 73-17959-06
Motherboard Serial Number          : FOC21418FPQ
Model Revision Number              : B0
Motherboard Revision Number        : A0
Model Number                       : C9300-48UXM
System Serial Number               : FCW2144L045


         --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: day0guestshell
installed successfully
Current state is: DEPLOYED
day0guestshell activated successfully
Current state is: ACTIVATED
day0guestshell started successfully
Current state is: RUNNING
Guestshell enabled successfully


HTTP server statistics:
Accepted connections total: 0

 *** Sample ZTP Day0 Python Script ***



 *** Executing show platform  ***
```

```
Switch  Ports    Model               Serial No.   MAC address     Hw Ver.       Sw Ver.
------  -----   ---------           -----------  --------------  -------       --------
 1       65     C9300-48UXM          FCW2144L045  ec1d.8b0a.6800  V01           16.12.3a

Switch/Stack Mac Address : ec1d.8b0a.6800 - Local Mac Address
Mac persistency wait time: Indefinite
                                     Current
Switch#   Role        Priority      State
-----------------------------------------
*1       Active          1          Ready


 *** Executing show version ***


Cisco IOS XE Software, Version 16.12.03a
Cisco IOS Software [Gibraltar], Catalyst L3 Switch Software (CAT9K_IOSXE), Version
16.12.3a,
RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2020 by Cisco Systems, Inc.
Compiled Tue 28-Apr-20 09:37 by mcpre
Cisco IOS-XE software, Copyright (c) 2005-2020 by cisco Systems, Inc.
All rights reserved.  Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0.  The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY.  You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0.  For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.
ROM: IOS-XE ROMMON
BOOTLDR: System Bootstrap, Version 17.2.1r[FC1], RELEASE SOFTWARE (P)
Switch uptime is 4 minutes
Uptime for this control processor is 9 minutes
System returned to ROM by Reload Command
System image file is "flash:cat9k_iosxe.16.12.03a.SPA.bin"
Last reload reason: Reload Command
This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.
A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html
If you require further assistance please contact us by sending email to
export@cisco.com.
Technology Package License Information:
------------------------------------------------------------------------------
Technology-package                                  Technology-package
Current                       Type                     Next reboot
------------------------------------------------------------------------------
network-advantage       Smart License                 network-advantage
None                    Subscription Smart License    None
AIR License Level: AIR DNA Advantage
Next reload AIR license Level: AIR DNA Advantage
Smart Licensing Status: UNREGISTERED/EVAL EXPIRED
cisco C9300-48UXM (X86) processor with 1343703K/6147K bytes of memory.
Processor board ID FCW2144L045
```

```
1 Virtual Ethernet interface
4 Gigabit Ethernet interfaces
36 2.5 Gigabit Ethernet interfaces
20 Ten Gigabit Ethernet interfaces
2 TwentyFive Gigabit Ethernet interfaces
2 Forty Gigabit Ethernet interfaces
2048K bytes of non-volatile configuration memory.
8388608K bytes of physical memory.
1638400K bytes of Crash Files at crashinfo:.
11264000K bytes of Flash at flash:.
0K bytes of WebUI ODM Files at webui:.
Base Ethernet MAC Address          : ec:1d:8b:0a:68:00
Motherboard Assembly Number        : 73-17959-06
Motherboard Serial Number          : FOC21418FPQ
Model Revision Number              : B0
Motherboard Revision Number        : A0
Model Number                       : C9300-48UXM
System Serial Number               : FCW2144L045
Switch Ports Model              SW Version        SW Image              Mode
------ ----- -----              ----------        ----------            ----
*    1 65    C9300-48UXM        16.12.3a          CAT9K_IOSXE           BUNDLE
Configuration register is 0x102


 *** Configuring a Loopback Interface ***


Line 1 SUCCESS: interface loop 100
Line 2 SUCCESS: ip address 10.10.10.10 255.255.255.255
Line 3 SUCCESS: end


 *** Executing show ip interface brief  ***


Interface              IP-Address      OK? Method Status                Protocol
Vlan1                  unassigned      YES unset  up                    up
GigabitEthernet0/0     10.127.128.10   YES DHCP   up                    up
Tw1/0/1                unassigned      YES unset  down                  down
Tw1/0/2                unassigned      YES unset  down                  down
Tw1/0/3                unassigned      YES unset  down                  down
Tw1/0/4                unassigned      YES unset  down                  down
Tw1/0/5                unassigned      YES unset  down                  down
Tw1/0/6                unassigned      YES unset  down                  down
Tw1/0/7                unassigned      YES unset  down                  down
Tw1/0/8                unassigned      YES unset  down                  down
Tw1/0/9                unassigned      YES unset  down                  down
Tw1/0/10               unassigned      YES unset  down                  down
Tw1/0/11               unassigned      YES unset  down                  down
Tw1/0/12               unassigned      YES unset  down                  down
Tw1/0/13               unassigned      YES unset  down                  down
Tw1/0/14               unassigned      YES unset  down                  down
Tw1/0/15               unassigned      YES unset  down                  down
Tw1/0/16               unassigned      YES unset  down                  down
Tw1/0/17               unassigned      YES unset  down                  down
Tw1/0/18               unassigned      YES unset  down                  down
Tw1/0/19               unassigned      YES unset  down                  down
Tw1/0/20               unassigned      YES unset  down                  down
Tw1/0/21               unassigned      YES unset  down                  down
Tw1/0/22               unassigned      YES unset  down                  down
Tw1/0/23               unassigned      YES unset  down                  down
Tw1/0/24               unassigned      YES unset  down                  down
Tw1/0/25               unassigned      YES unset  down                  down
Tw1/0/26               unassigned      YES unset  down                  down
```

```
Tw1/0/27               unassigned      YES unset  down                  down
Tw1/0/28               unassigned      YES unset  down                  down
Tw1/0/29               unassigned      YES unset  down                  down
Tw1/0/30               unassigned      YES unset  down                  down
Tw1/0/31               unassigned      YES unset  down                  down
Tw1/0/32               unassigned      YES unset  down                  down
Tw1/0/33               unassigned      YES unset  down                  down
Tw1/0/34               unassigned      YES unset  down                  down
Tw1/0/35               unassigned      YES unset  down                  down
Tw1/0/36               unassigned      YES unset  down                  down
Te1/0/37               unassigned      YES unset  down                  down
Te1/0/38               unassigned      YES unset  down                  down
Te1/0/39               unassigned      YES unset  down                  down
Te1/0/40               unassigned      YES unset  down                  down
Te1/0/41               unassigned      YES unset  down                  down
Te1/0/42               unassigned      YES unset  down                  down
Te1/0/43               unassigned      YES unset  down                  down
Te1/0/44               unassigned      YES unset  down                  down
Te1/0/45               unassigned      YES unset  down                  down
Te1/0/46               unassigned      YES unset  down                  down
Te1/0/47               unassigned      YES unset  down                  down
Te1/0/48               unassigned      YES unset  up                    up
GigabitEthernet1/1/1   unassigned      YES unset  down                  down
GigabitEthernet1/1/2   unassigned      YES unset  down                  down
GigabitEthernet1/1/3   unassigned      YES unset  down                  down
GigabitEthernet1/1/4   unassigned      YES unset  down                  down
Te1/1/1                unassigned      YES unset  down                  down
Te1/1/2                unassigned      YES unset  down                  down
Te1/1/3                unassigned      YES unset  down                  down
Te1/1/4                unassigned      YES unset  down                  down
Te1/1/5                unassigned      YES unset  down                  down
Te1/1/6                unassigned      YES unset  down                  down
Te1/1/7                unassigned      YES unset  down                  down
Te1/1/8                unassigned      YES unset  down                  down
Fo1/1/1                unassigned      YES unset  down                  down
Fo1/1/2                unassigned      YES unset  down                  down
TwentyFiveGigE1/1/1    unassigned      YES unset  down                  down
TwentyFiveGigE1/1/2    unassigned      YES unset  down                  down
Ap1/0/1                unassigned      YES unset  up                    up
Loopback100            10.10.10.10     YES TFTP   up                    up


 *** Configuring username, password, SSH  ***


Line 1 SUCCESS: username cisco privilege 15 password cisco
**CLI Line # 1:  WARNING: Command has been added to the configuration using a type 0
password.
However, type 0 passwords will soon be deprecated. Migrate to a supported password type
Line 2 SUCCESS: ip domain name domain
Line 3 SUCCESS: line vty 0 15
Line 4 SUCCESS: login local
Line 5 SUCCESS: transport input all
Line 6 SUCCESS: end


 *** ZTP Day0 Python Script Execution Complete ***



Guestshell destroyed successfully
```

```
Press RETURN to get started!
```

Cisco IOS XE Amsterdam 17.2.x 以降のリリース

このセクションでは、.py スクリプトを実行する前の起動ログのサンプルを表示します。

```
--- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]:
Acquired IPv4 address 10.127.128.8 on Interface GigabitEthernet0/0
Received following DHCPv4 options:
        bootfile        : test.py
        tftp-server-ip  : 159.14.27.2

OK to enter CLI now...

pnp-discovery can be monitored without entering enable mode

Entering enable mode will stop pnp-discovery

Attempting bootfile tftp://159.14.27.2/test.py
day0guestshell activated successfully
Current state is: ACTIVATED
day0guestshell started successfully
Current state is: RUNNING
Guestshell enabled successfully


 *** Sample ZTP Day0 Python Script ***

...

 *** ZTP Day0 Python Script Execution Complete ***

Guestshell destroyed successfully
```

このセクションでは、デイゼロプロビジョニング用にデバイスを設定する方法を示します。

```
Both links down, not waiting for other switches
Switch number is 1

              Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

           Cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706



Cisco IOS Software [Amsterdam], Catalyst L3 Switch Software (CAT9K_IOSXE), Version 17.2.1,
 RELEASE SOFTWARE (fc4)
Technical Support: http://www.cisco.com/techsupport
```

```
Copyright (c) 1986-2020 by Cisco Systems, Inc.
Compiled Thu 26-Mar-20 03:29 by mcpre


This software version supports only Smart Licensing as the software licensing mechanism.


PLEASE READ THE FOLLOWING TERMS CAREFULLY. INSTALLING THE LICENSE OR
LICENSE KEY PROVIDED FOR ANY CISCO SOFTWARE PRODUCT, PRODUCT FEATURE,
AND/OR SUBSEQUENTLY PROVIDED SOFTWARE FEATURES (COLLECTIVELY, THE
"SOFTWARE"), AND/OR USING SUCH SOFTWARE CONSTITUTES YOUR FULL
ACCEPTANCE OF THE FOLLOWING TERMS. YOU MUST NOT PROCEED FURTHER IF YOU
ARE NOT WILLING TO BE BOUND BY ALL THE TERMS SET FORTH HEREIN.

Your use of the Software is subject to the Cisco End User License Agreement
(EULA) and any relevant supplemental terms (SEULA) found at
http://www.cisco.com/c/en/us/about/legal/cloud-and-software/software-terms.html.

You hereby acknowledge and agree that certain Software and/or features are
licensed for a particular term, that the license to such Software and/or
features is valid only for the applicable term and that such Software and/or
features may be shut down or otherwise terminated by Cisco after expiration
of the applicable license term (e.g., 90-day trial period). Cisco reserves
the right to terminate any such Software feature electronically or by any
other means available. While Cisco may provide alerts, it is your sole
responsibility to monitor your usage of any such term Software feature to
ensure that your systems and networks are prepared for a shutdown of the
Software feature.


% Checking backup nvram
% No config present. Using default config

FIPS: Flash Key Check : Key Not Found, FIPS Mode Not Enabled

All TCP AO KDF Tests Pass
cisco C9300-48UXM (X86) processor with 1338934K/6147K bytes of memory.
Processor board ID FCW2144L045
2048K bytes of non-volatile configuration memory.
8388608K bytes of physical memory.
1638400K bytes of Crash Files at crashinfo:.
11264000K bytes of Flash at flash:.

Base Ethernet MAC Address          : ec:1d:8b:0a:68:00
Motherboard Assembly Number        : 73-17959-06
Motherboard Serial Number          : FOC21418FPQ
Model Revision Number              : B0
Motherboard Revision Number        : A0
Model Number                       : C9300-48UXM
System Serial Number               : FCW2144L045
CLEI Code Number                   :


No startup-config, starting autoinstall/pnp/ztp...

Autoinstall will terminate if any input is detected on console

Autoinstall trying DHCPv4 on GigabitEthernet0/0

Autoinstall trying DHCPv6 on GigabitEthernet0/0


         --- System Configuration Dialog ---
```

```
Would you like to enter the initial configuration dialog? [yes/no]:
Acquired IPv4 address 10.127.128.8 on Interface GigabitEthernet0/0
Received following DHCPv4 options:
        bootfile        : test.py
        tftp-server-ip  : 159.14.27.2

OK to enter CLI now...

pnp-discovery can be monitored without entering enable mode

Entering enable mode will stop pnp-discovery

Attempting bootfile tftp://159.14.27.2/test.py
day0guestshell activated successfully
Current state is: ACTIVATED
day0guestshell started successfully
Current state is: RUNNING
Guestshell enabled successfully


 *** Sample ZTP Day0 Python Script ***



 *** Executing show platform  ***


Switch  Ports    Model               Serial No.   MAC address     Hw Ver.       Sw Ver.

------  -----   ---------           -----------  --------------  -------       --------

 1       65     C9300-48UXM          FCW2144L045  ec1d.8b0a.6800  V01           17.02.01

Switch/Stack Mac Address : ec1d.8b0a.6800 - Local Mac Address
Mac persistency wait time: Indefinite
                                    Current
Switch#   Role        Priority      State
-----------------------------------------
*1       Active          1          Ready


 *** Executing show version ***


Cisco IOS XE Software, Version 17.02.01
Cisco IOS Software [Amsterdam], Catalyst L3 Switch Software (CAT9K_IOSXE), Version 17.2.1,
 RELEASE SOFTWARE (fc4)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2020 by Cisco Systems, Inc.
Compiled Thu 26-Mar-20 03:29 by mcpre
Cisco IOS-XE software, Copyright (c) 2005-2020 by cisco Systems, Inc.
All rights reserved.  Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0.  The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY.  You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0.  For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.
ROM: IOS-XE ROMMON
BOOTLDR: System Bootstrap, Version 17.2.1r[FC1], RELEASE SOFTWARE (P)
Switch uptime is 2 minutes
Uptime for this control processor is 8 minutes
```

```
System returned to ROM by Reload Command
System image file is "flash:cat9k_iosxe.17.02.01.SPA.bin"
Last reload reason: Reload Command
This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.
A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html
If you require further assistance please contact us by sending email to
export@cisco.com.
Technology Package License Information:
------------------------------------------------------------------------------
Technology-package                                      Technology-package
Current                       Type                         Next reboot
------------------------------------------------------------------------------
network-advantage       Smart License                    network-advantage
None                    Subscription Smart License       None
AIR License Level: AIR DNA Advantage
Next reload AIR license Level: AIR DNA Advantage
Smart Licensing Status: UNREGISTERED/EVAL EXPIRED
cisco C9300-48UXM (X86) processor with 1338934K/6147K bytes of memory.
Processor board ID FCW2144L045
1 Virtual Ethernet interface
4 Gigabit Ethernet interfaces
36 2.5 Gigabit Ethernet interfaces
20 Ten Gigabit Ethernet interfaces
2 TwentyFive Gigabit Ethernet interfaces
2 Forty Gigabit Ethernet interfaces
2048K bytes of non-volatile configuration memory.
8388608K bytes of physical memory.
1638400K bytes of Crash Files at crashinfo:.
11264000K bytes of Flash at flash:.
Base Ethernet MAC Address          : ec:1d:8b:0a:68:00
Motherboard Assembly Number        : 73-17959-06
Motherboard Serial Number          : FOC21418FPQ
Model Revision Number              : B0
Motherboard Revision Number        : A0
Model Number                       : C9300-48UXM
System Serial Number               : FCW2144L045
CLEI Code Number                   :
Switch Ports Model              SW Version        SW Image              Mode
------ ----- -----              ----------        ----------            ----
*    1 65    C9300-48UXM        17.02.01          CAT9K_IOSXE           BUNDLE
Configuration register is 0x102


 *** Configuring a Loopback Interface ***


Line 1 SUCCESS: interface loop 100
Line 2 SUCCESS: ip address 10.10.10.10 255.255.255.255
Line 3 SUCCESS: end


 *** Executing show ip interface brief  ***


Interface              IP-Address      OK? Method Status                Protocol
Vlan1                  unassigned      YES unset  up                    up
```

```
GigabitEthernet0/0     10.127.128.8    YES DHCP   up                    up
Tw1/0/1                unassigned      YES unset  down                  down
Tw1/0/2                unassigned      YES unset  down                  down
Tw1/0/3                unassigned      YES unset  down                  down
Tw1/0/4                unassigned      YES unset  down                  down
Tw1/0/5                unassigned      YES unset  down                  down
Tw1/0/6                unassigned      YES unset  down                  down
Tw1/0/7                unassigned      YES unset  down                  down
Tw1/0/8                unassigned      YES unset  down                  down
Tw1/0/9                unassigned      YES unset  down                  down
Tw1/0/10               unassigned      YES unset  down                  down
Tw1/0/11               unassigned      YES unset  down                  down
Tw1/0/12               unassigned      YES unset  down                  down
Tw1/0/13               unassigned      YES unset  down                  down
Tw1/0/14               unassigned      YES unset  down                  down
Tw1/0/15               unassigned      YES unset  down                  down
Tw1/0/16               unassigned      YES unset  down                  down
Tw1/0/17               unassigned      YES unset  down                  down
Tw1/0/18               unassigned      YES unset  down                  down
Tw1/0/19               unassigned      YES unset  down                  down
Tw1/0/20               unassigned      YES unset  down                  down
Tw1/0/21               unassigned      YES unset  down                  down
Tw1/0/22               unassigned      YES unset  down                  down
Tw1/0/23               unassigned      YES unset  down                  down
Tw1/0/24               unassigned      YES unset  down                  down
Tw1/0/25               unassigned      YES unset  down                  down
Tw1/0/26               unassigned      YES unset  down                  down
Tw1/0/27               unassigned      YES unset  down                  down
Tw1/0/28               unassigned      YES unset  down                  down
Tw1/0/29               unassigned      YES unset  down                  down
Tw1/0/30               unassigned      YES unset  down                  down
Tw1/0/31               unassigned      YES unset  down                  down
Tw1/0/32               unassigned      YES unset  down                  down
Tw1/0/33               unassigned      YES unset  down                  down
Tw1/0/34               unassigned      YES unset  down                  down
Tw1/0/35               unassigned      YES unset  down                  down
Tw1/0/36               unassigned      YES unset  down                  down
Te1/0/37               unassigned      YES unset  down                  down
Te1/0/38               unassigned      YES unset  down                  down
Te1/0/39               unassigned      YES unset  down                  down
Te1/0/40               unassigned      YES unset  down                  down
Te1/0/41               unassigned      YES unset  down                  down
Te1/0/42               unassigned      YES unset  down                  down
Te1/0/43               unassigned      YES unset  down                  down
Te1/0/44               unassigned      YES unset  down                  down
Te1/0/45               unassigned      YES unset  down                  down
Te1/0/46               unassigned      YES unset  down                  down
Te1/0/47               unassigned      YES unset  down                  down
Te1/0/48               unassigned      YES unset  up                    up
GigabitEthernet1/1/1   unassigned      YES unset  down                  down
GigabitEthernet1/1/2   unassigned      YES unset  down                  down
GigabitEthernet1/1/3   unassigned      YES unset  down                  down
GigabitEthernet1/1/4   unassigned      YES unset  down                  down
Te1/1/1                unassigned      YES unset  down                  down
Te1/1/2                unassigned      YES unset  down                  down
Te1/1/3                unassigned      YES unset  down                  down
Te1/1/4                unassigned      YES unset  down                  down
Te1/1/5                unassigned      YES unset  down                  down
Te1/1/6                unassigned      YES unset  down                  down
Te1/1/7                unassigned      YES unset  down                  down
Te1/1/8                unassigned      YES unset  down                  down
Fo1/1/1                unassigned      YES unset  down                  down
Fo1/1/2                unassigned      YES unset  down                  down
TwentyFiveGigE1/1/1    unassigned      YES unset  down                  down
```

```
TwentyFiveGigE1/1/2    unassigned      YES unset  down                  down
Ap1/0/1                unassigned      YES unset  up                    up
Loopback100            10.10.10.10     YES TFTP   up                    up


 *** Configuring username, password, SSH  ***


Line 1 SUCCESS: username cisco privilege 15 password cisco
**CLI Line # 1:  WARNING: Command has been added to the configuration using a type 0
password.
However, type 0 passwords will soon be deprecated. Migrate to a supported password type
Line 2 SUCCESS: ip domain name domain
Line 3 SUCCESS: line vty 0 15
Line 4 SUCCESS: login local
Line 5 SUCCESS: transport input all
Line 6 SUCCESS: end


 *** ZTP Day0 Python Script Execution Complete ***



Guestshell destroyed successfully
Script execution success!



Press RETURN to get started!
```

ゼロ タッチ プロビジョニングの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *2:* ゼロ タッチ プロビジョニングの機能情報

機能名	リリース	機能情報
ゼロ タッチ プロビジョニング	Cisco IOS XE Everest 16.5.1a Cisco IOS XE Everest 16.5.1b Cisco IOS XE Fuji 16.7.1 Cisco IOS XE Fuji 16.8.2 Cisco IOS XE Gibraltar 16.12.1 Cisco IOS XE Amsterdam 17.2.1 Cisco IOS XE Amsterdam 17.3.1	

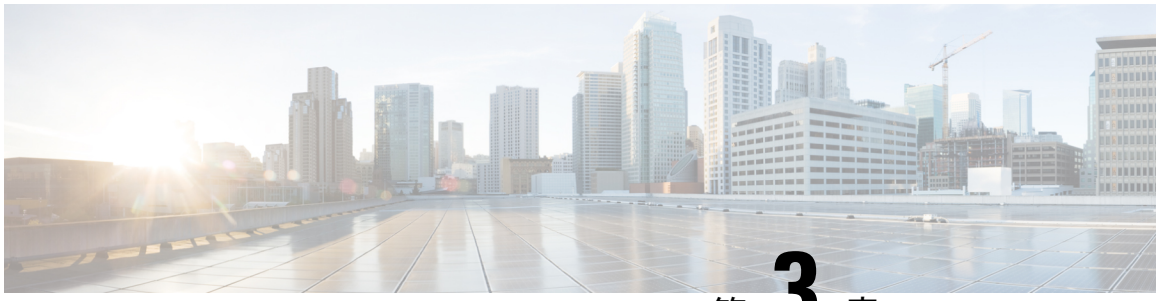
機能名	リリース	機能情報
		ネットワーク プロビジョニングの課題に対応するため、シスコは、ゼロ タッチ プロビジョニング モデルを導入しました。 Cisco IOS XE Everest 16.5.1a では、この機能は次のプラットフォームに実装されていました。 • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.5.1b では、この機能は次のプラットフォームに実装されていました。 • ゲストシェルをサポートするための、最低 8 GB の RAM を搭載した Cisco 4000 シリーズ サービス統合型ルータ モデル。 Cisco IOS XE Fuji 16.7.1 では、この機能は次のプラットフォームに実装されていました。 • Cisco ASR 1000 アグリゲーション サービス ルータ（ASR1001-X、ASR1001-HX、ASR1002-X、ASR1002-HX） Cisco IOS XE Fuji 16.8.2 では、この機能は次のプラットフォームに実装されていました。 • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ（ASR1004、ASR1006、ASR1006-X、ASR1009-X、ASR1013）

機能名	リリース	機能情報
		この機能は、Cisco IOS XE Gibraltar 16.12.1 で次のプラットフォームに実装されました。 • Cisco Catalyst 9200 シリーズ スイッチ （注） この機能は C9200L SKU ではサポートされていません。 • Cisco Catalyst 9300L SKU • Cisco Catalyst 9600 シリーズ スイッチ • Cisco Catalyst 9800-40 ワイヤレスコントローラ • Cisco Catalyst 9800-80 ワイヤレスコントローラ この機能は、Cisco IOS XE Amsterdam 17.2.1で次のプラットフォームに実装されました。 • Cisco Cloud Services Router 1000V シリーズ • Cisco C1100 ターミナル サービス ゲートウェイ（C1100TGX-1N24P32A でのみサポート） この機能は、Cisco IOS XE Amsterdam 17.3.1 で次のプラットフォームに実装されました。 • Cisco Catalyst 8200 シリーズ エッジ プラットフォーム • Cisco Catalyst 8300 シリーズ エッジ プラットフォーム • Cisco Catalyst 8500 および 8500L シリーズ エッジ プラットフォーム

機能名	リリース	機能情報
		この機能は、Cisco IOS XE Bengaluru 17.4.1で次のプラットフォームに実装されました。 • Cisco Catalyst 8000V Edge ソフトウェア
ゼロ タッチ プロビジョニング：HTTP ダウンロード	Cisco IOS XE Fuji 16.8.1 Cisco IOS XE Fuji 16.8.1a	ゼロ タッチプロビジョニングは、HTTP および TFTP のファイル ダウンロードをサポートします。 Cisco IOS XE Everest 16.8.1 では、この機能は次のプラットフォームに実装されていました。 • Cisco 4000 シリーズ サービス統合型ルータ • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Fuji 16.8.1a では、この機能は Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチに実装されていました。

機能名	リリース	機能情報
ゼロタッチ プロビジョニングのための DHCPv6 のサポート	Cisco IOS XE Fuji 16.9.1 Cisco IOS XE Amsterdam 17.3.2a	Cisco IOS XE Fuji 16.9.1 では、この機能は次のプラットフォームに実装されていました。 • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Amsterdam 17.3.2a では、この機能は次のプラットフォームに導入されました。 • Cisco Catalyst 9800-40 ワイヤレスコントローラ • Cisco Catalyst 9800-80 ワイヤレスコントローラ

機能名	リリース	機能情報
コンフィギュレーションデータベースの副次的同期	Cisco IOS XE Bengaluru 17.4.1	DMI の設定変更中に、コマンドまたは RPC の設定時にトリガーされる変更の部分的な同期が行われます。これは副次的同期と呼ばれ、同期時間と NETCONF のダウンタイムを短縮します。 この機能は、次のプラットフォームに実装されていました。 • Cisco ASR 1000 アグリゲーション サービス ルータ • Cisco Catalyst 8500 および 8500L シリーズ エッジ プラットフォーム • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ



第 3 章

iPXE

iPXE は、ネットワーク ブーティングのオープン スタンダードである Pre-boot eXecution Environment（PXE）の拡張版です。このモジュールでは、iPXE 機能および設定方法について説明します。

iPXE について

iPXE について

iPXE は、ネットワーク ブーティングのオープン スタンダードである Pre-boot eXecution Environment（PXE）の拡張版です。

iPXE ネットブートは、次を提供します。

- IPv4 および IPv6 プロトコル
- FTP/HTTP/TFTP ブート イメージのダウンロード
- イメージへの埋め込みスクリプト
- Dynamic Host Configuration Protocol バージョン 4（DHCPv4）や DHCPv6 を使用したステートレスおよびステートフル アドレス自動設定（SLAAC）、ブート URI、および IPv6 ルータ アドバタイズメントに応じた DHCPv6 オプションのパラメータ。

ネットブート要件

ネット ブーティングの主な要件は、次のとおりです。

- 適切に設定された DHCP サーバ。
- FTP/HTTP/TFTP サーバ上で使用可能なブート イメージ。
- ネットワーク ベースのソースから起動するように設定されたデバイス。

iPXE の概要

ネットワーク ブートローダは、ネットワーク ベースのソースからのブート処理をサポートします。ブートローダは、HTTP、FTP、またはTFTPサーバにあるイメージを起動します。ネットワーク ブート ソースは、iPXE のようなソリューションを使用して自動検出されます。

iPXE により、オフラインのデバイスのネットワーク ブートが可能になります。ブート モードには次の 3 種類があります。

- iPXE タイムアウト：iPXE ネットワーク ブートを介して起動します。IPXE_TIMEOUT ROMmon 変数を使用して、iPXE ネットワーク ブートのタイムアウトを秒単位で設定します。iPXE タイムアウトを設定するには **boot ipxe timeout** コマンドを使用します。タイムアウト時間を経過すると、デバイス ブートがアクティブになります。
- iPXE 期限なし：iPXE ネットワーク ブートを介して起動します。**boot ipxe forever** コマンドが設定されている場合、デバイスは DHCP 要求を期限なしで送信します。これは iPXE のみを使うブートです（つまり、ブートローダは、有効なDHCP応答を受け取るまでDHCP要求を期限なしで送信するため、デバイス ブートまたはコマンド プロンプトにフォールバックすることはありません）。
- デバイス：設定されているローカル デバイスの BOOT 行を使ってブートします。デバイス ブートが設定された場合、設定されているIPXE_TIMEOUT ROMmon変数は無視されます。次のように指定してデバイス ブートをアクティブ化できます。
  - BOOTMODE=ipxe-forever の場合は、ユーザの介入がなければデバイス ブートがアクティブになりません（ENABLE_BREAK=yes の場合にのみ可能）。
  - BOOTMODE=ipxe-timeout の場合は、IPXE_TIMEOUT 変数で指定した秒数が経過するとデバイス ブートがアクティブになります。
  - BOOTMODE=device の場合は、デバイス ブートがアクティブになります。これはデフォルトのアクティブ モードです。
  - デバイス ブートは CLI を使用してアクティブ化することもできます。

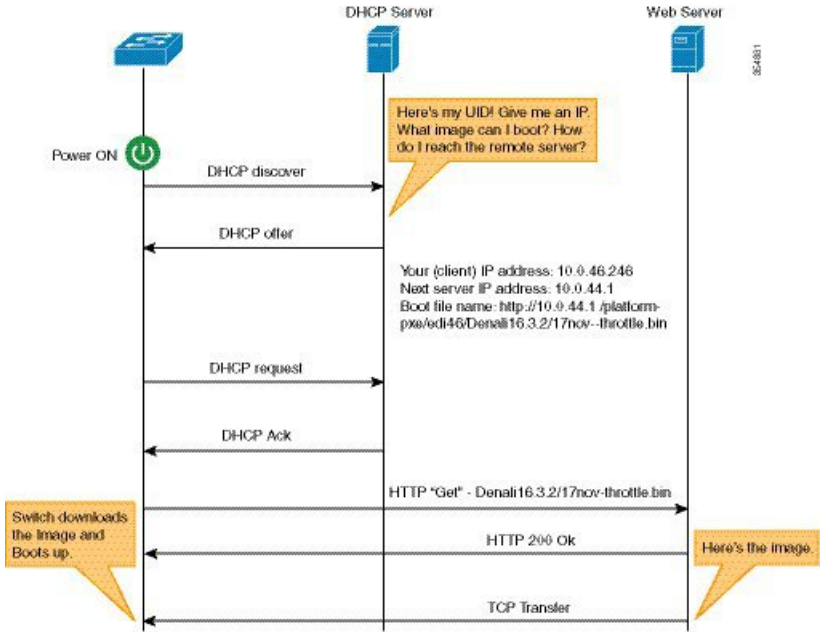
（注） デバイス ブートは、デフォルトのブート モードです。

（注） このマニュアルでは、手動ブートという用語も使われています。手動ブートは、ROMmon のリロードを行うかどうかを決定するフラグです。デバイスが ROMmon モードの場合は、手動で **boot** コマンドを実行する必要があります。

手動ブートを YES に設定した場合は、ROMmon またはデバイス プロンプトがアクティブになります。手動ブートを NO に設定した場合は、autoboot 変数が実行されます。つまり、BOOT 変数で設定された値に従います。

ここでは、iPXE ブートローダの動作について説明します。

図 *1 : iPXE* ブートローダのワークフロー



1. ブートローダは DHCP 検出メッセージを送信し、サーバが応答すると、ブートローダは DHCP 要求を送信します。
2. DHCP 応答には、IP アドレスとのブート ファイル名が含まれています。ブート ファイル名は、ブート イメージが TFTP サーバ（tftp://server/filename）、FTP サーバ（ftp://userid:password@server/filename）、または HTTP サーバ（http://server/filename）から取得されることを示しています。
3. ブートローダがネットワーク ソースからイメージをダウンロードして起動します。
4. DHCP 応答が受信されない場合、ブートローダはブート モードの設定に基づいて、DHCP 要求を期限なしで、または指定された期間の間送信し続けます。タイムアウトが発生すると、ブートローダはデバイスベースのブートに戻ります。設定されたブート モードが **ipxe-forever** の場合のみ、デバイスは DHCP 要求を期限なしで送信します。**ipxe-timeout** ブート モード コマンドが設定されている場合、DHCP 要求は指定された時間にわたって送信され、タイムアウトが経過すると、デバイス ブート モードがアクティブになります。

（注） 現在の iPXE 実装は管理ポート（GigabitEthernet0/0）のみを経由して動作するため、前面パネル ポートを介して送信される DHCP 要求はサポートされていません。

ネットワーク ブートに対して静的なネットワーク設定を使用する場合、ROMmon は次の環境変数を使用します（すべて必須です）。

- BOOT：セミコロン（;）で区切られた起動元の URL。
- IP_ADDRESS：静的に割り当てられたデバイスの IP アドレス。
- DEFAULT_GATEWAY：デバイスのデフォルト ゲートウェイ。
- IP_SUBNET_MASK：IPv4 または IPv6 プレフィックス情報。

  IPv4：WWW.XXX.YYY.ZZZ という形式のデバイスのサブネット マスク（255.255.255.0 など）。

  IPv6：NNN という形式のデバイスのサブネット プレフィックス長（64、112 など）。

手動ブートが無効になっている場合、ブートローダは、設定された ROMmon iPXE 変数の値に基づいて、デバイス ブートを実行するかネットワーク ブートを実行するかを決定します。手動ブートが有効か無効かにかかわらず、ブートローダは BOOTMODE 変数を使用して、デバイス ブートとネットワーク ブートのどちらを実行するかを決定します。手動ブートは、ユーザによって **boot manual switch** コマンドが設定済みであることを意味します。手動ブートが無効になっている場合にデバイスをリロードすると、起動プロセスが自動的に開始されます。

iPXE が無効になっている場合は、デバイスの起動方法の決定に、既存の BOOT 変数の内容が使用されます。BOOT 変数には、ネットワークベースの Uniform Resource Identifier（URI）（たとえば、http://、ftp://、tftp://）が含まれている場合があり、ネットワーク ブートが開始されます。しかし、ネットワーク イメージ パスの取得に DHCP は使用されません。静的なネットワーク設定は、IP_ADDRESS 変数、DEFAULT_GATEWAY 変数、および IP_SUBNET_MASK 変数から取得されます。BOOT 変数には、デバイスのファイル システム ベースのパスが含まれている場合もあり、この場合は、デバイスのファイル システム ベースのブートが開始されます。

起動に使用される DHCP サーバは、製品 ID（PID）（DHCP オプション 60 で判別可能）、シャーシのシリアル番号（DHCP オプション 61 で判別可能）、またはデバイスの MAC アドレスを使用して、デバイスを識別できます。**show inventory** および **show switch** コマンドでもデバイスでこれらの値を表示します。

次に、**show inventory** コマンドの出力例を示します。

```
Device# show inventory

NAME:"c38xx Stack", DESCR:"c38xx Stack"
PID:WS-3850-12X-48U-L, VID:V01 ,  SN: F0C1911V01A

NAME:"Switch 1", DESCR:"WS-C3850-12X48U-L"
PID:WS-C3850-12X48U-L, VID:V01 , SN:F0C1911V01A

NAME:"Switch1 -Power Supply B", DESCR:"Switch1 -Power Supply B"
PID:PWR-C1-1100WAC, VID:V01, SN:LIT1847146Q
```

次に、 **show switch** コマンドの出力例を示します。

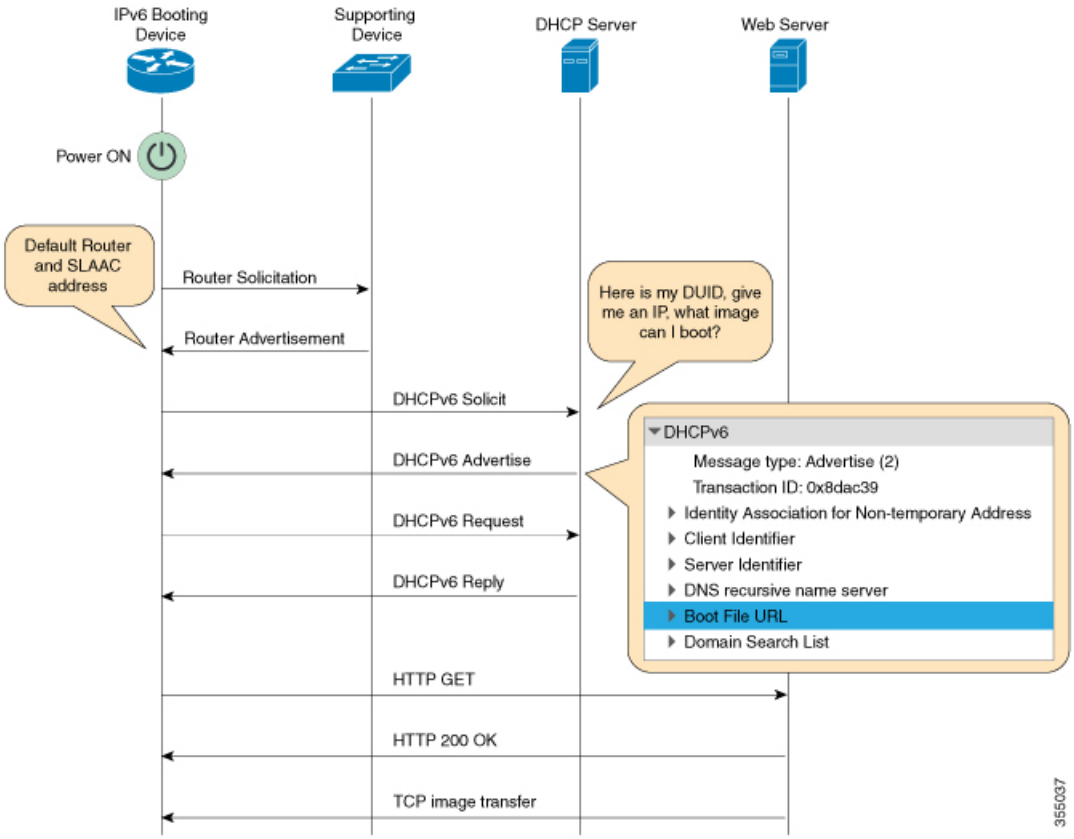
```
Device# show switch

Switch/Stack Mac Address : 046c.9d01.7d80 - Local Mac Address
Mac persistency wait time: Indefinite
                                             H/W   Current
Switch#   Role    Mac Address     Priority Version  State
-----------------------------------------------------------------------------------
 1       Member   046c.9d1e.1a00     1              Ready
 2       Standby  046c.9d01.7d80     1              Ready
*3       Active   f8b7.e24e.9a00     1      P2B     Ready
```

次の ROMmon 変数が iPXE に設定されている必要があります。

- BOOTMODE = ipxe-forever | ipxe-timeout | device
- IPXE_TIMEOUT = seconds

IPv6 iPXE ネットワーク ブート

次の図は、Cisco デバイス上の IPv6 iPXE ネットワーク ブートの動作を表します。



次に、上掲の図の 4 つの要素を説明します。

- IPv6 ブート デバイス：iPXE ブートによって起動するデバイス。

- サポート デバイス：IPv6アドレスで、ルータ アドバタイズメント（RA）メッセージを生成するように設定された Cisco デバイス。

（注） この図では、IPv6 ブート デバイス、サポート デバイス、および DHCP サーバは、同じサブネット上にあります。ただし、サポート デバイスと DHCP サーバが異なるサブネット上にある場合、ネットワーク内にリレー エージェントを設ける必要があります。

- DHCP サーバ：任意の DHCP サーバ。
- Web サーバ：任意の Web サーバ。

この項では、IPv6 iPXE ブート プロセスを説明します。

1. デバイスは、ルータ要請である Internet Control Message Protocol IPv6（ICMPv6）タイプ 133 パケットをローカル サブネット上の IPv6 デバイスに送信します。
2. ローカル サブネット上の IPv6 デバイスは、ルータ アドバタイズメント（RA）メッセージである ICMPv6 タイプ 134 パケットで応答します。ルータ要請メッセージを送信したデバイスは、ステートレス アドレス自動設定（SLAAC）アドレスを完成させるため、RA パケットからデフォルト ルータとプレフィックスの情報を取得します。
3. デバイスは、DHCPv6 要請メッセージを、すべての DHCP エージェントについて、マルチキャスト グループ アドレス ff02::1:2 に送信します。

   次に、iPXE ブートの際の DHCPv6 要請パケットのフィールドの例を示します。

```
DHCPv6
 Message type: Solicit (1)
 Transaction ID: 0x36f5f1
 Client Identifier
 Vendor Class
 Identity Association for Non-Temporary Address
 Option Request
 User Class
 Vendor-specific Information
```

   DHCPv6 要請メッセージには、次の情報が含まれています。

   - DHCP 固有識別子（DUID）：クライアントを識別します。iPXE では、DUID-EN をサポートしています。EN は、エンタープライズ番号（Enterprise Number）の略です。この DUID は、ベンダーに割り当てられた固有の識別子に基づいています。
   - DHCP および DHCPv6 のオプション

4. DHCPv6 サーバが設定されている場合、そのサーバは、128 ビット IPv6 アドレス、ブートファイルの Uniform Resource Identifier（URI）、ドメイン ネーム システム（DNS）サーバおよびドメイン検索リスト、ならびにクライアントとサーバの ID を含む DHCPv6 アドバタイズ パケットで応答します。クライアント ID にはクライアント（この図では IPv6 ブー

トデバイス）の DUID が、サーバ ID には DHCPv6 サーバの DUID が、それぞれ含まれています。

5. それを受け、クライアントは、マルチキャスト グループ アドレス ff02::1:2 に DHCPv6 要求パケットを送信し、アドバタイズされたパラメータを要求します。
6. サーバは、クライアントのリンク ローカル（FE80::）の IPv6 アドレスにユニキャスト DHCPv6 応答を返します。次に、DHCPv6 応答パケットのフィールドの例を示します。

```
DHCPv6
 Message type: Reply (7)
 Transaction ID: 0x790950
 Identity Association for Non-Temporary Address
 Client Identifier
 Server Identifier
 DNS recursive name server
 Boot File URL
 Domain Search List
```

7. 次に、デバイスは、Web サーバに HTTP GET 要求を送信します。
8. 要求されたイメージが指定されたパスで使用可能な場合、Web サーバは、HTTP GET 要求に OK を返します。
9. TCP イメージ転送によりイメージがコピーされ、デバイスが起動します。

ROMmon モードでの IPv6 アドレスの割り当て

DHCP クライアントは、次の優先順位を使用して、ROMmon モードで使用する IPv6 アドレスを決定します。

1. DHCP サーバによって割り当てられたアドレス
2. ステートレス アドレス自動設定（SLAAC）アドレス
3. リンクローカル アドレス
4. サイトローカル アドレス

デバイスは、イメージをブートするのにDHCPサーバによって割り当てられたアドレスを使用します。DHCPv6 サーバがアドレスの割り当てに失敗した場合、デバイスは、SLAAC アドレスの使用を試行します。DHCP サーバによって割り当てられたアドレスと SLAAC アドレスの両方が使用できない場合、デバイスは、リンクローカル アドレスを使用します。ただし、イメージのコピーを正常に行うには、リモート FTP/HTTP/TFTP サーバがデバイスと同じローカル サブネット上にある必要があります。

最初の3つのアドレスが使用できない場合、デバイスは、自動的に生成されるサイトローカル アドレスを使用します。

サポートされる ROMmon 変数

Cisco IOS XE Fuji 16.8.1 では、次の ROMmon 変数がサポートされています。

- BAUD：デバイスのコンソール ボー レートをシスコの標準ボー レート（1200、2400、4800、9600、19200、38400、57600、115200 など）のいずれかに変更します。無効な値はすべて拒否されます。BAUD 変数が設定されていない場合は、デフォルトで 9600 になります。対応する CLI コマンドは、
- ENABLE_BREAK：ROMmon のブレークを有効にします。デフォルト値は NO です。
- MANUAL_BOOT：手動ブートが 1 に設定されている場合、ROMmon またはデバイス プロンプトがアクティブになります。手動ブートが 0 に設定されている場合、デバイスはリロードされますが、ROMmon モードはアクティブになりません。
- SWITCH_IGNORE_STARTUP_CFG：値が 1 の場合は、デバイスでスタートアップ コンフィギュレーションが無視されます。値が設定されていない場合は、値がゼロとみなされます。これは読み取り専用変数であり、IOS のみが変更できます。

iPXE がサポートする DHCP オプション

iPXE ブートは、ROMmon モードで次の DHCPv4 および DHCPv6 オプションをサポートしています。

（注） Catalyst 9000 シリーズ スイッチは、DHCP オプション 60、オプション 77、DHCPv6 オプション 1、オプション 15、およびオプション 16 をサポートしています。DHCP オプション 61 は、Catalyst 9300 および 9500 シリーズ スイッチでのみサポートされています。

- DHCP オプション 60：ベンダー クラス識別子。このオプションには、ROMmon 環境変数 MODEL_NUM の値が設定されます。
- DHCP オプション 61：クライアント識別子。このオプションには、ROMmon 環境変数 SYSTEM_SERIAL_NUM の値が設定されます。

（注） このオプションは Catalyst 9400 シリーズ スイッチではサポートされていません。

- DHCP オプション 77：ユーザ クラス オプション。このオプションは、DHCP 検出パケットに追加されるもので、iPXE という文字列に等しい値を含んでいます。このオプションは、DHCP サーバからブートするためのイメージを探す iPXE DHCP クライアントを分離する際に使用されます。

次に、ISC DHCP サーバからの DHCPv4 設定で、オプション 77 の使用が示されている例を示します。この例における if 条件は、オプション 77 が存在しており、文字列 iPXE に等しい場合は、イメージのブート ファイルの URI がアドバタイズされることを示します。

```
host Switch2 {
     fixed-address 192.168.1.20 ;
     hardware ethernet CC:D8:C1:85:6F:11 ;
           #user-class = length of string + ASCII code for iPXE
     if exists user-class and option user-class = 04:68:50:58:45 {
         filename "http://192.168.1.146/test-image.bin"
     }
}
```

- DHCPv6 オプション 1：クライアント識別子オプション。このオプションには、RFC 3315 で規定されている ROMmon 環境変数 SYSTEM_SERIAL_NUM の値が設定されます。ROMmon 環境変数で推奨される形式は MAC_ADDR です。
- DHCPv6 オプション 15：ユーザ クラス オプション。このオプションは、DHCPv6 要請メッセージ内の IPv6 ユーザ クラス オプションであり、文字列 iPXE が設定されます。次に、ISC DHCP サーバで定義されているオプション 15 の例を示します。

```
option dhcp6.user-class code 15 = string ;
```

次に、DHCPv6 オプション 15 が使用されている DHCP サーバ設定の例を示します。

```
#Client-specific parameters
host switch1 {
     #assigning a fixed IPv6 address
     fixed-address6 2001:DB8::CAFE ;
     #Client DUID in hexadecimal format contains: DUID-type"2" + "EN=9" + "Chassis
serial number"
    host-identifier option dhcp6.client-id      00:02:00:00:00:09:46:4F:43:31:38:33:
31:58:31:41:53;
    #User class 00:04:69:50:58:45 is len 4 + "iPXE"
    if option dhcp6.user-class = 00:04:69:50:58:45 {
       option dhcp6.bootfile-url
"http://[2001:DB8::461/platform-pxe/edi46/test-image.bin";
    }
}
```

- DHCPv6 オプション 16：ベンダー クラス オプション。デバイスの製品 ID（PID）が含まれています。PID は、**show inventory** コマンドの出力または MODEL_NUM ROMmon 変数から特定できます。オプション 16 は ISC DHCP サーバのデフォルトのオプションではなく、次のように定義することができます。

```
option dhcp6.vendor-class-data code 16 = string;
```

次に、DHCPv6 オプション 16 が使用されている設定例を示します。

```
# Source: dhcpd6ConfigPD

host host1-ipxe6-auto-host1 {
     fixed-address6 2001:DB8::1234;
     host-identifier option dhcp6.client-id 00:02:00:00:00:09:46:4F:
```

```
     43:31:38:33:31:58:31:41:53;
     if option dhcp6.vendor-class-data = 00:00:00:09:00:0E:57:53:2D:
     43:33:38:35:30:2D:32:34:50:2D:4D {
     option dhcp6.bootfile-url
"http://[2001:DB8::46]/platform-pxe/host1/17jan-polaris.bin";
```

次の表で、この出力に表示される重要なフィールドを説明します。

表 *3:* サンプル出力フィールドの説明

フィールド	説明
dhcp6.client-id	クライアントを識別する DHCP 固有識別子（DUID）。
dhcp6.user-class	DHCPv6 オプション 15、ユーザ クラス オプション。
dhcp6.vendor-class-data	DHCPv6 オプション 16、スイッチの製品 ID（PID）を含むベンダー クラス オプション。
dhcp6.bootfile-url	ブート ファイル URI を要求する DHCPv6 オプション 6。

DHCPv6 固有識別子

RFC 3315 によって定義されている DHCPv6 識別子（DUID）には、次の 3 種類があります。

- DUID-LLT：DUID リンク層アドレスと時刻。DHCP デバイスに接続しているネットワーク インターフェイスのリンク層アドレスに、生成された時刻のタイムスタンプが追加されたものです。
- DUID-EN：EN は、エンタープライズ番号（Enterprise Number）の略です。この DUID は、ベンダーに割り当てられた固有の ID に基づいています。
- DUID-LL：DHCP（クライアント/サーバ）デバイスに永久的に接続されているネットワーク インターフェイスのリンク層アドレスを使用して形成される DUID です。

この機能をサポートしているシスコ デバイスは、DHCP クライアント（DHCPv6 要請パケット内のデバイス）を識別するのに DUID-EN（DUID タイプ 2）を使用します。Catalyst 9000 シリーズ スイッチは、DUID-EN だけでなく DUID-LL（DUID タイプ 3）もサポートしています。DUID-EN は優先される型です。ただし、スイッチがこの型を作成できない場合は、DUID-LL が作成されて使用されます。

iPXE の設定方法

iPXE の設定

手順の概要

1. **enable**
2. **configure terminal**
3. • **boot ipxe forever** [*switch number*]
   • **boot ipxe timeout** *seconds* [*switch number*]
4. **boot system** {**switch** *switch-number* | **all**} {**flash:** | **ftp:** | **http:** | **usbflash0** | **tftp:**}
5. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	• **boot ipxe forever** [*switch number*]
	• **boot ipxe timeout** *seconds* [*switch number*]
	例：
	`Device(config)# boot ipxe forever switch 2`
	例：
	`Device(config)# boot ipxe timeout 30 switch 2`
		BOOTMODE ROMmon 変数を設定します。
		• **forever** キーワードは、BOOTMODE ROMmon 変数を IPXE-FOREVER として設定します。
		• **timeout** キーワードは、BOOTMODE ROMmon 変数を IPXE-TIMEOUT として設定します。

ステップ 4	**boot system** {**switch** *switch-number*	**all**} {**flash:**	**ftp:**	**http:**	**usbflash0**	**tftp:**}
						例：
						`Device(config)# boot system switch 1 http://192.0.2.42/image-filename`
						または
						`Device(config)# boot system switch 1 http://[2001:db8::1]/image-filename`
							指定した場所からイメージを起動します。
							• リモートの FTP/HTTP/TFTP サーバには、IPv4 または IPv6 アドレスを使用できます。
							• 角かっこ内に IPv6 アドレスを入力する必要があります（RFC 2732 に従って）。そうしない場合、デバイスは起動しません。

	コマンドまたはアクション	目的
ステップ 5	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

デバイス ブートの設定

デバイス ブートは、**no boot ipxe** または **default boot ipxe** コマンドのいずれかを使用して設定できます。

手順の概要

1. **enable**
2. **configure terminal**
3. • **no boot ipxe**
   • **default boot ipxe**
4. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	• **no boot ipxe**
	• **default boot ipxe**
	例：
	`Device(config)# no boot ipxe`
	例：
	`Device(config)# default boot ipxe`
		デバイスブートを設定します。デフォルトのブートモードはデバイス ブートです。
		デバイスでデフォルト設定を有効にします。

ステップ 4	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

iPXE の設定例

例：iPXE 構成

以下は、デバイスがイメージで起動するまで、DHCP 要求を期限なしで送信するように iPXE を設定する例を示しています。

```
Device# configure terminal
Device(config)# boot ipxe forever switch 2
Device(config)# end
```

以下は、ブート モードを ipxe-timeout に設定する方法の例を示します。設定されているタイムアウト値は 200 秒です。設定されているタイムアウト経過後に iPXE ブート障害が発生する場合、設定されているデバイス ブートがアクティブになります。この例で、設定済みのデバイス ブートは http://[2001:db8::1]/image-filename です。

```
Device# configure terminal
Device(config)# boot ipxe timeout 200 switch 2
Device(config)# boot system http://[2001:db8::1]/image-filename
Device(config)# end
```

サンプルの iPXE ブート ログ

次に示すのは、ROMmon モードのデバイスからのサンプル ブート ログです。ここでは、**ipxe-timeout** コマンドを使用した手動ブートが設定されます。

```
switch: boot

pxemode:(ipxe-timeout) 60s timeout
00267.887 ipxe_get_booturl: Get URL from DHCP; timeout 60s
00267.953 ipxe_get_booturl: trying DHCPv6 (#1) for 10s
IPv4:
           ip addr 192.168.1.246
           netmask 255.255.255.0
           gateway 192.168.1.46
IPv6:
link-local addr fe80::ced8:c1ff:fe85:6f00
site-local addr fec0::ced8:c1ff:fe85:6f00
      DHCP addr 2001:db8::cafe
    router addr fe80::f29e:63ff:fe42:4756
     SLAAC addr 2001:db8::ced8:c1ff:fe85:6f00 /64
Common:
        macaddr cc:d8:c1:85:6f:00
            dns 2001:db8::46
       bootfile
http://[2001:DB8::461/platform-pxe/edi46/17jan-dev.bin--13103--2017-Feb28--13-54-50
         domain cisco.com
00269.321 ipxe_get_booturl: got URL
(http://[2001:DB8::461/platform-pxe/edi46/17jan-dev.bin--13103--2017-Feb-28--13-54-50)
```

```
Reading full image into memory ….…….…….…….…….…….…….…….…….…….…….…….…….…….…….…….…….…...
Bundle Image
-----------------------------------------------
Kernel Address    : 0x5377a7e4
Kernel Size       : 0x365e3c/3563068
Initramfs Address : 0x53ae0620
Initramfs Size    : 0x13a76f0/20608752
Compression Format: mzip
```

iPXE 用のサンプル DHCPv6 サーバ構成

次に示すのは、参考のために Internet Systems Consortium（ISC）の DHCP サーバから取得した DHCPv6 サーバ設定の例です。先頭に文字 # がある行は、続く構成を説明しているコメントです。

```
Default-least-time 600;
max-lease-time-7200;
log-facility local7;

#Global configuration
#domain search list
option dhcp6.domain-search "cisco.com" ;
#User-defined options:new-name code new-code = definition ;
option dhcp6.user-class code 15 = string ;
option dhcp6.vendor-class-data code 16 = string;

subnet6 2001:db8::/64 {
 #subnet range for clients requiring an address
 range6 2001:db8:0000:0000::/64;

#DNS server options
option dhcp6.name-servers 2001:db8::46;

}
#Client-specific parameters
host switch1 {
    #assigning a fixed IPv6 address
    fixed-address6 2001:DB8::CAFE ;
    #Client DUID in hexadecimal that contains: DUID-type "2" + "EN=9" + "Chassis serial
 number"
    host-identifier option dhcp6.client-id 00:02:00:00:00:09:46:4F:43:31:38:33:
31:58:31:41:53;
   option dhcp6.bootfile-url "http://[2001:DB8::461/platform-pxe/edi46/test-image.bin";
}
```

DHCP サーバ コマンドの詳細については、*ISC DHCP* サーバ の Web サイトを参照してください。

この設定例では、dhcp6.client-id オプションはスイッチを識別し、エンタープライズ クライアント DUID が続きます。クライアント DUID は、16 進形式の 00:02 + 00:00:00:09 + のシャーシ シリアル番号を理解するために分解できます。ここで 2 はエンタープライズ クライアント DUID タイプ、9 はシスコのエンタープライズ DUID の予約済みコードをそれぞれ参照し、16 進形式でのシャーシ シリアル番号の ASCII コードが続きます。このサンプルのスイッチのシャーシ シリアル番号は、FOC1831X1AS です。

ブート ファイル URI は、指定された DUID を使用してのみスイッチにアドバタイズされます。

DHCPv6 ベンダー クラス オプション 16 も、DHCP サーバ上のスイッチを識別するために使用できます。オプション 16 をユーザ定義オプションとして定義するには、次のように設定します。

```
option dhcp6.vendor-class-data code 16 = string;
```

次に示すのは、スイッチ製品 ID を使用して形成された DHCPv6 ベンダー クラス オプション 16 に基づいてスイッチを識別する、DHCP サーバの構成例です。

```
# Source: dhcp6ConfigPID

host edi-46-ipxe6-auto-edi46 {
    fixed-address6 2001:DB8::1234;
    host-identifier option dhcp6.client-id 00:02:00:00:00:09:
    46:4F:43:31:38:33:31:58:31:58:31:41:53;
    if option dhcp6.vendor-class-data = 00:00:00:09:00:0E:57:
    53:2D:43:33:38:35:30:2D:32:34:50:2D:4C {
      option dhcp6.bootfile-url "http://[2001:DB8::461/platform-pxe/edi46/17jan-dev.bin";

    }
}
```

この構成例では、dhcp6.vendor-class-data オプションは、DHCPv6 オプション 16 を参照します。dhcp6.vendor-class-data で、00:00:00:09 はシスコのエンタープライズ DUID、0E は PID の長さ、および残りは 16 進形式の PID です。PID は、**show inventory** コマンドまたは CFG_MODEL_NUM ROMmon 変数の出力から特定することもできます。このサンプル構成で使用される PID は、WS-C3850-24P-L です。

サーバ構成の DHCPv6 オプションおよび DUID は、ISC DHCP サーバのガイドラインに従って、16 進形式で指定する必要があります。

iPXE のトラブルシューティングのヒント

この項では、トラブルシューティングのヒントを説明します。

- 電源投入時に iPXE ブートが有効化されると、デバイスは、最初に DHCPv6 要請メッセージの送信を試行し、その後で、DHCPv4 検出メッセージの送信を試行します。ブート モードが **ipxe-forever** の場合、デバイスは、この 2 つを期限なしで反復し続けます。
- 起動モードが iPXE タイムアウトの場合、デバイスは、最初に DHCPv6 要請メッセージを、次に DHCPv4 検出メッセージを送信した後、タイムアウト時間が経過すると、デバイス ブートにフォールバックします。
- iPXE ブートを中断するには、コンソールにシリアル ブレイクを送信します。

  UNIX Telnet クライアントを使用している場合は、Ctrl キーを押した状態で] キーを押すと、ブレイクが送信されます。その他の Telnet クライアントを使用している場合、または

シリアルポートに直接接続している場合は、ブレイクの送信のトリガーは、別のキーストロークまたはコマンドの場合があります。

- DHCP サーバはイメージで応答するものの DNS サーバがホスト名を解決できない場合、DNS デバッグを有効にします。

（注） ISC の DHCP サーバの使用をお勧めします。IOS の DHCP ではこの機能はまだ検証されていません。

- HTTP サーバの接続をテストするには、HTTP コピーを使用して、HTTP サーバから少量のサンプル ファイルをデバイスにコピーします。たとえば ROMmon プロンプトで、**copy http://192.168.1.1/test null:**（フラッシュが通常はロックされており、テストに Null デバイスを使用する必要がある場合）または **http://[2001:db8::99]/test** と入力します。
- 手動ブートが有効化されており、ブート モードが iPXE タイムアウトである場合、デバイスが電源投入時に自動的に起動することはありません。ROMmon モードで **boot** コマンドを実行します。ブートプロセスが電源投入時に自動で発生するようにするには、手動ブートを無効にします。
- ROMmon モード の IPv6 アドレスやデフォルト ルータを含む現在の IPv6 パラメータを表示するには、**net6-show** コマンドを使用します。

（注） Catalyst 9000 シリーズ スイッチでは、 **net-show** show コマンドを使用します。

- 設定に基づいて、**net-dhcp** または **net6-dhcp** コマンドを使用します。**net-dhcp** コマンドは DHCPv4 用のテスト コマンド、**net6-dhcp** コマンドは DHCPv6 用のテスト コマンドです。

（注） Catalyst 9000 シリーズ スイッチでは、DHCPv6 に **net-dhcp -6** コマンドを使用します。

- 名前を解決するには、**dig** コマンドを使用します。

（注） Catalyst 9000 シリーズ スイッチでは、**dns-lookup** コマンドを使用して名前を解決します。

- Web サーバからの HTTP 応答コードを表示するには、HTTP デバッグ ログを有効にします。

- ステートレス アドレス自動設定（SLAAC）アドレスが生成されない場合、IPv6 RA メッセージを提供するルータがありません。この場合、IPv6 での iPXE ブートは、リンクローカルまたはサイトローカルのアドレスでのみ使用できます。

iPXE に関する追加情報

関連資料

関連項目	マニュアル タイトル
プログラマビリティ コマンド	『Programmability Command Reference, Cisco IOS XE Everest 16.6.1』

標準および RFC

標準/RFC	タイトル
RFC 3315	『*Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*』
RFC 3986	『*Uniform Resource Identifier (URI): Generic Syntax*』

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

iPXE の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

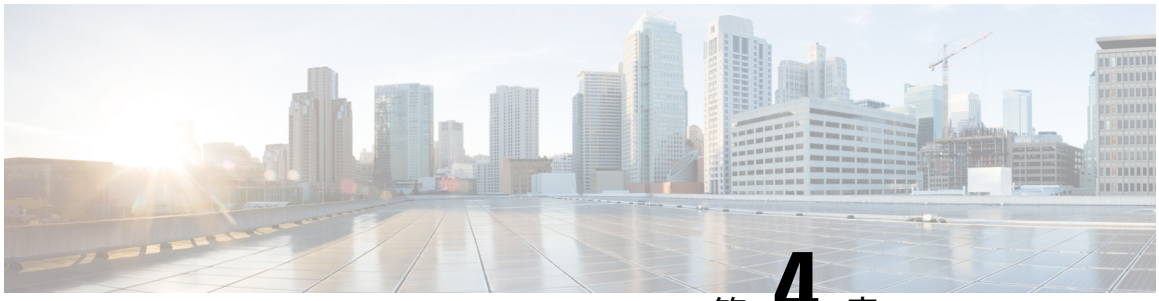
表 *4 : iPXE* の機能情報

機能名	リリース	機能情報
iPXE	Cisco IOS XE Denali 16.5.1a	ネットワーク ブートローダは、IPv4/IPv6 デバイス ベースまたはネットワーク ベースの送信元からのブート処理をサポートします。ネットワーク ブート ソースは、iPXE のようなソリューションを使用して自動的に検出される必要があります。 この機能は、次のプラットフォームに実装されていました。 • Catalyst 3650 シリーズ スイッチ • Catalyst 3850 シリーズ スイッチ
	Cisco IOS XE Denali 16.6.1	Cisco IOS XE Denali 16.6.1 では、この機能は次のプラットフォームに実装されていました。 • Catalyst 9300 シリーズ スイッチ • Catalyst 9500 シリーズ スイッチ
	Cisco IOS XE Everest 16.6.2	この機能は、Cisco IOS XE Everest 16.6.2 で、Cisco Catalyst 9400 シリーズ スイッチに実装されました。
	Cisco IOS XE Fuji 16.9.2	Cisco IOS XE Fuji 16.9.2 では、この機能は次のプラットフォームに実装されていました。 • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300L SKU
	Cisco IOS XE Gibraltar 16.11.1	Cisco IOS XE Gibraltar 16.11.1 では、この機能は Cisco Catalyst 9600 シリーズ スイッチに実装されていました。

機能名	リリース	機能情報
IPXE IPv6 のサポート	Cisco IOS XE 16.8.1a	IPXE は IPv6 プロトコルをサポートしています。 この機能は、次のプラットフォームに実装されていました。 • Catalyst 9300 シリーズ スイッチ • Catalyst 9400 シリーズ スイッチ • Catalyst 9500 シリーズ スイッチ

第 II 部

シェルとスクリプト化



第 4 章

ゲスト シェル

ゲスト シェルは仮想化された Linux ベースの環境で、Python などの カスタム Linux アプリケーションを実行して Cisco デバイスを自動で制御および管理するために設計されています。システムの自動プロビジョニング（デイ ゼロ）も含まれます。このコンテナ シェルは、ホスト デバイスから分離された安全な環境を提供します。ユーザはそこで、スクリプトまたはソフトウェア パッケージをインストールし、実行することができます。

このモジュールでは、ゲスト シェルとそれを有効にする方法について説明します。

ゲストシェルの制約事項

- ゲストシェルは、Cisco Catalyst 9200L SKU ではサポートされません。
- スタンバイルートプロセッサ（RP） では、NETCONF セッションを確立できません。

ゲスト シェルについて

ゲスト シェルの概要

ゲスト シェルは、仮想化された Linux ベースの環境であり、Cisco デバイスの自動制御と管理のための Python アプリケーションを含む、カスタム Linux アプリケーションを実行するように設計されています。ゲスト シェルを使用して、サードパーティ製 Linux アプリケーションをイ

ンストール、更新、および操作することもできます。ゲスト シェルはシステム イメージとともにバンドルされており、Cisco IOS コマンド **guestshell enable** を使用してインストールできます。

ゲスト シェル環境は、ネットワーキングではなく、ツール、Linux ユーティリティ、および管理性を意図したものです。

ゲスト シェルは、ホスト（Cisco スイッチおよびルータ）システムとカーネルを共有します。ユーザーは、ゲスト シェルの Linux シェルにアクセスし、コンテナの rootfs にあるスクリプトおよびソフトウェア パッケージを更新することができます。ただし、ゲスト シェル内のユーザーは、ホストのファイル システムおよびプロセスを変更することはできません。

ゲスト シェル コンテナは、IOx を使用して管理されます。IOx は、Cisco IOS XE デバイスのためのシスコのアプリケーション ホスティング インフラストラクチャです。IOx は、シスコ、パートナー、およびサードパーティの開発者によって開発されたアプリケーションおよびサービスをネットワーク エッジ デバイスでシームレスにホスティングすることを、各種の多様なハードウェア プラットフォームにおいて可能にします。

ゲストシェルのソフトウェア要件

ゲストシェル コンテナを使用すると、ユーザは、システム上で自分のスクリプトやアプリケーションを実行できるようになります。Intel x86 プラットフォーム上のゲストシェルコンテナは、CentOS 8.0 の最小限の rootfs を持つ Linux コンテナ（LXC）になります。ランタイム中に、CentOS 8.0 で Yum ユーティリティを使用して、Python バージョン 3.0 などの他の Python ライブラリをインストールすることができます。また、PIP を使用して Python パッケージをインストールまたは更新することもできます。

表 *5:* ゲストシェルのソフトウェア要件

	ゲスト シェル（LXC コンテナ）
オペレーティング システム	Cisco IOS XE
プラットフォーム	サポートされているすべての Cisco IOS XE プラットフォーム
ゲスト シェル環境	• CentOS 7 は Cisco IOS XE Amsterdam 17.2.1 以前のリリースでサポートされています。 • CentOS 8 は Cisco IOS XE Amsterdam 17.3.1 以降のリリースでサポートされています。 （注） CentOS は Python 3.6 のみをサポートします。
Python 2.7	Cisco IOS XE Amsterdam 17.3.1 までサポート

	ゲスト シェル（**LXC** コンテナ）
Python 3.6	Cisco IOS XE Amsterdam 17.1.1 以降のリリースでサポートされています。 Cisco IOS XE Amsterdam 17.1.1 および Cisco IOS XE Amsterdam 17.2.1 では、Python V2 がデフォルトです。ただし、Cisco IOS XE Amsterdam 17.3.1 以降のリリースでは、Python V3 がデフォルトです。 （注） Cisco Catalyst 9200 シリーズ スイッチは、Cisco IOS XE Amsterdam 17.3.1 以降のリリースで Python バージョン 3 をサポートしています。
事前にインストールされたカスタムの Python ライブラリ	• Cisco 組込イベント マネージャ • Cisco IOS XE CLI
サポートされる rootfs	SSH、Yum のインストール、および Python PIP のインストール
GNU C コンパイラ	サポート対象外
RPM のインストール	サポートあり
アーキテクチャ	x86 および ARM

ゲスト シェルのセキュリティ

シスコは、ゲスト シェル内のユーザまたはアプリケーションによってホスト システムが攻撃されることがないよう、セキュリティを提供しています。ゲスト シェルは、ホスト カーネルから分離され、非特権コンテナとして動作します。

ゲスト シェルのハードウェア要件

この項では、可変メモリ構成を持つ、サポート対象のプラットフォームにおけるハードウェア要件に関する情報を提供します。

表 *6 :* ゲストシェルのリソース要件

プラットフォーム	最小メモリ
Cisco 1000 シリーズ サービス統合型ルータ	4 GB
Cisco Cloud Services Router 1000V シリーズ	4 GB

プラットフォーム	最小メモリ
Cisco ISR 4000 シリーズ サービス統合型ルータ	8 GB DRAM（Cisco IOS XE Fuji 16.8.1 以前のリリース。） 4 GB DRAM（Cisco IOS XE Fuji 16.8.1 以降のリリース。）

他のすべてのプラットフォームは、ゲストシェルをサポートするのに十分なリソースを備えた状態で出荷されます。

（注） 仮想サービスがインストールされているアプリケーションとゲストシェルコンテナを同時に使用することはできません。

ゲスト シェルのストレージ要件

Cisco Catalyst 9300 シリーズ スイッチおよび Cisco Catalyst 9500 シリーズ スイッチでは、ゲストシェルを正常にインストールするには 1100 MB のハードディスク空き容量が必要です。

Cisco 4000 シリーズ サービス統合型ルータでは、ゲストシェルは、ネットワーク インターフェイス モジュール（NIM）の SSD（ハードディスク）がある場合、そこにインストールされます。ハードディスク ドライブが使用可能な場合、ゲスト シェルのインストールにブートフラッシュを選択することはできません。Cisco 4000 シリーズ サービス統合型ルータでは、ゲストシェルを正常にインストールするには 1100 MB のハードディスク（NIM-SSD）空き容量が必要です。

Cisco 4000 シリーズ サービス統合型ルータおよび Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ（オプションのハードディスクがそのルータに追加されている場合）では、ゲストシェルをハードディスクにインストールしており、そのハードディスクがルータに挿入されている場合にのみリソースのサイズ変更を実行できます。

（注） ブートフラッシュを介してインストールしたゲストシェルでは、アプリケーション ホスティング設定コマンドを使用したリソースのサイズ変更はできません。

ゲスト シェルのインストール中にハード ディスク容量が不足した場合、エラー メッセージが表示されます。

次に、Cisco ISR 4000 シリーズ サービス統合型ルータでのエラーメッセージの例を示します

```
% Error:guestshell_setup.sh returned error:255, message:
Not enough storage for installing guestshell. Need 1100 MB free space.
```

ブートフラッシュまたはハード ディスクの空き領域は、ゲスト シェルが追加データを格納するために使用されることがあります。Cisco 4000 シリーズ サービス統合型ルータでは、ゲスト

シェルに 800 MB のストレージ空き領域があります。ゲスト シェルはブートフラッシュにアクセスするため、その空き領域の全体を使用できます。

表 *7:* ゲスト シェルおよびゲスト シェル *Lite* が使用できるリソース

リソース	デフォルト	最小/最大
CPU	1 % （注） 1 % は非標準。800 CPU ユニット/システム CPU ユニットの全体	1/100 %
メモリ	256 MB 512 MB（Cisco Cloud Services Router 1000V シリーズ）	256/256 MB 512/512 MB（Cisco Cloud Services Router 1000V シリーズ）

ゲスト シェルの有効化と実行

guestshell enable コマンドは、ゲスト シェルをインストールします。このコマンドは、無効化されているゲスト シェルを再アクティブ化する際にも使用されます。

ゲスト シェルが有効化された状態でシステムをリロードすると、ゲスト シェルは有効化されたままになります。

（注） **guestshell enable** コマンドを使用する前に、IOx を設定しておく必要があります。

guestshell run bash コマンドは、ゲスト シェルの bash プロンプトを開きます。このコマンドを動作させるには、ゲスト シェルが事前に有効化されていることが必要です。

（注） 次のメッセージがコンソールに表示される場合、IOx が有効化されていません。**show iox-service** コマンドの出力をチェックして、IOx の状態を確認してください。

```
The process for the command is not responding or is otherwise unavailable
```

ゲストシェルを有効にする方法の詳細については、「Configuring the AppGigabitEthernet Interface for Guest Shell」および「Enabling Guest Shell on the Management Interface」のセクションを参照してください。

ゲスト シェルの無効化と破棄

guestshell disable コマンドを使用することで、ゲスト シェルを終了して無効化できます。ゲスト シェルが無効化された状態でシステムをリロードすると、ゲスト シェルは無効化されたままになります。

guestshell destroy コマンドは、フラッシュのファイル システムから rootfs を削除します。すべてのファイル、データ、インストールされている Linux アプリケーション、およびカスタムの Python ツールとユーティリティが削除され、回復できなくなります。

デバイスでのゲスト シェルへのアクセス

ネットワーク管理者は、Cisco IOS コマンドを使用して、ゲストシェル内のファイルおよびユーティリティを管理することができます。

ゲスト シェルのインストール中に、SSH アクセスがキー ベースの認証でセットアップされます。ゲストシェルへのアクセスは、Cisco IOS の最も高い特権（15）を持つユーザに制限されます。このユーザは、sudo の実行者である guestshell Linux ユーザとして Linux コンテナへのアクセスを許可され、すべてのルート操作を実行できます。ゲストシェルから実行されるコマンドは、ユーザが Cisco IOS 端末にログインしたときと同じ特権で実行されます。

ゲスト シェル プロンプトでは、標準的な Linux コマンドを実行できます。

管理ポートを介してのゲスト シェルへのアクセス

ゲストシェルは、デフォルトで、アプリケーションによる管理ネットワークへのアクセスを許可します。ユーザは、ゲスト シェル内から管理 VRF のネットワーキング設定を変更することはできません。

（注） 管理ポートがないプラットフォームの場合、VirtualPortGroup を Cisco IOS 設定内のゲストシェルに関連付けることができます。詳細については、「VirtualPortGroup の設定例」の項を参照してください。

Cisco Catalyst 9200 シリーズ スイッチ、Cisco Catalyst 9300 シリーズ スイッチ、および Cisco Catalyst 9400 シリーズ スイッチは、ゲストシェルにアクセスするために AppGigabitEthernet インターフェイスおよび管理インターフェイス（mgmt-if）をサポートします。

Catalyst 9500 シリーズ スイッチ、Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ、および Catalyst 9600 シリーズ スイッチでは、AppGigabitEthernet インターフェイスはサポートされません。

（注） Cisco Catalyst 9200L SKU はゲストシェルをサポートしていません。

前面パネルポートまたは光ファイバアップリンクを使用した デイ ゼロ ゲスト シェル プロビジョニング

デイゼロでは、デバイスに管理接続がなく、唯一の接続が前面パネルポートまたはファイバアップリンクポートのいずれかを介して行われる場合、ゲストシェルは使用可能なポートを使用するように内部的に設定されます。AppGigabitEthernet インターフェイスは、ゲストシェルをサーバに接続します。

ゲストシェルがサーバに接続されると、デバイスは構成スクリプトをダウンロードし、デバイスを設定します。この設定には、仮想マシン（VM）のダウンロード、設定、起動も含まれます。デイゼロ設定が完了すると、設定に基づいてシステムがリブートする場合があります。システムがユーザ固有の設定のみで起動することを確認します。

USB ポートを使用したゲストシェル接続

デバイスは、シリアルアダプタを使用して複数の他のデバイスに接続します。このシリアルアダプタは、デバイスの前面パネルにある USB ポートを介して接続されます。

VM はシリアルアダプタを制御し、VM の実行中に USB インターフェイスにアタッチされている接続済みデバイスに変更があると、VM に通知されます。

ゲスト シェルでのスタッキング

ゲストシェルがインストールされている場合、フラッシュのファイルシステムには、ディレクトリが自動的に作成されます。このディレクトリは、スタックメンバー間で同期されます。切り替え時には、このディレクトリの内容のみが、すべてのスタックメンバー間で同期されます。ハイ アベイラビリティでの切り替えの際にデータを保持するには、このディレクトリにデータを格納します。

ハイアベイラビリティでの切り替えの際には、新しいアクティブデバイスは、それぞれのゲストシェルインストールを作成し、ゲストシェルを同期状態に復元します。古いファイルシステムは維持されません。ゲストシェルの状態は、すべてのスタックメンバー間で内部的に同期されます。

Cisco IOx の概要

Cisco IOx（IOs＋linuX）はエンドツーエンドアプリケーション フレームワークであり、Cisco ネットワーク プラットフォーム上のさまざまなタイプのアプリケーションに対し、アプリケーション ホスティング機能を提供します。Cisco ゲストシェルは特殊なコンテナ展開であり、システムの開発に役立つアプリケーションの 1 つです。

Cisco IOx は、構築済みアプリケーションをパッケージ化し、それらをターゲットデバイス上にホストする開発者の作業を支援する一連のサービスを提供することにより、アプリケーションのライフサイクル管理とデータ交換を容易にします。IOx のライフサイクル管理には、アプリケーションおよびデータの配布、展開、ホスティング、開始、停止（管理）、およびモニタが含まれます。IOx サービスにはアプリケーションの配布および管理ツールも含まれており、ユーザがアプリケーションを発見して IOx フレームワークに展開するのに役立ちます。

Cisco IOx アプリケーション ホスティングは、次の機能を提供します。

- ネットワークの不均質性の遮蔽。
- デバイス上にホストされているアプリケーションのライフサイクルをリモートで管理する Cisco IOx アプリケーション プログラミング インターフェイス（API）。
- 一元化されたアプリケーションのライフ サイクル管理。
- クラウド ベースの開発。

IOx のトレースとロギングの概要

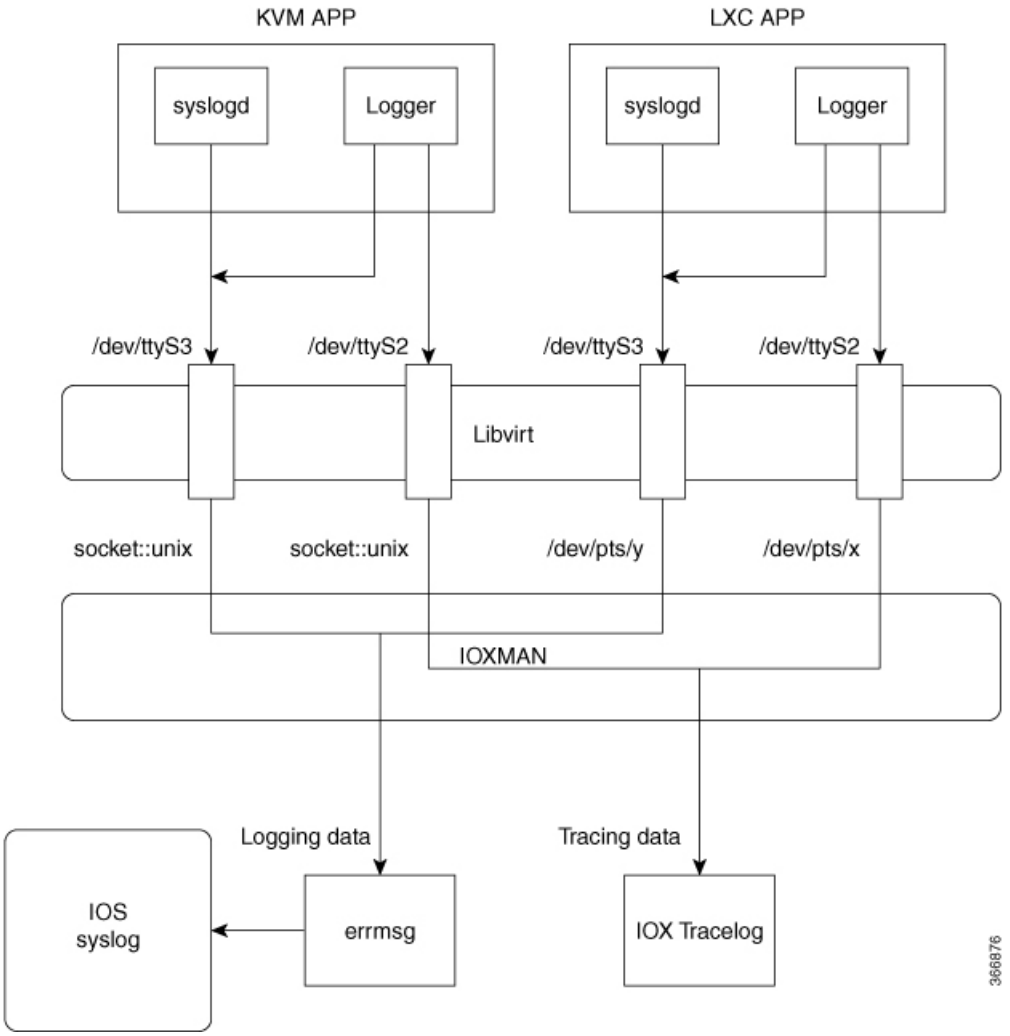
IOx のトレースとロギングの機能を使用すると、ホスト デバイスでゲスト アプリケーションを個別に実行できます。これにより、ホストへのデータのロギングとトレースをレポートするのに役立ちます。トレースデータはIOxトレースログに保存され、ロギングデータはホストデバイスの Cisco IOS syslog に保存されます。

トレース データをホスト デバイス上の適切なストレージ デバイスにリダイレクトすると、ゲスト アプリケーションのデバッグに役立ちます。

IOXMAN 構造体

ゲストアプリケーション、システムLXC、またはKVMインスタンスはそれぞれ独自のsyslogおよびログファイルを使用して設定されます。これらのファイルは表示可能なファイルシステム内に保存され、ホスト デバイスからはアクセスできません。Cisco IOS syslog へのデータのロギングとホスト上のIOxトレースログへのデータのトレースをサポートするため、次の図に示すように、ホストにデータを配信するための 2 つのシリアルデバイス（**/dev/ttyS2** と **/dev/ttyS3**）がゲストアプリケーションで指定されています。

図 *2 : IOXMAN* 構造体



IOXMANは、トレースインフラストラクチャを確立してロギングサービスまたはトレースサービス（シリアルデバイスをエミュレートするLibvirtを除く）を提供するプロセスです。IOXMANは、ゲストアプリケーションのライフサイクルに基づいて、トレースサービスを有効または無効にし、ロギングデータを Cisco IOS syslog に送信し、トレースデータを IOx トレースログに保存し、各ゲストアプリケーションの IOx トレースログを維持します。

ゲストシェルからの NETCONF アクセス

NETCONF-YANG にはゲストシェル内からアクセスできるため、ユーザーは Python スクリプトを実行し、NETCONF プロトコルを使用してシスコカスタムパッケージ CLI を呼び出すことができます。

ゲストシェルアプリケーションは、ユーザー名として guestshell を使用することで、ローカルホストおよび NETCONF ポートへのパスワードレス SSH 接続を行わずに SSH 接続を確立します。このユーザー名は、デバイスに設定されている実際のユーザーに対応していません。デバ

イスに guestshell ユーザーが設定されている場合でも、このパスワードレスアクセスへの接続はありません。PRIV15 権限レベルを持つユーザーのみがゲストシェル内から NETCONF にアクセスできます。

認証と認可はバイパスされません。代わりに、ゲストシェルにアクセスを許可するときに認証と許可が行われます。最大の権限を持つユーザーのみがこのアクセスを許可されます。

ユーザーは外部ポートを開かずにゲストシェルから NETCONF サービスにアクセスできます。デバイスの NETCONF-YANG サーバーに接続する前に、ゲストシェルで初期化コマンドを実行する必要があります。これらのコマンドは次のとおりです。

```
iosp_client -f netconf_enable guestshell <port-number> and
iosp_client -f netconf_enable_passwordless guestshell <username>
```

iosp_client -f netconf_enable guestshell *port-number* コマンドは、**netconf-yang ssh local-vrf guestshell** コマンドを設定し、NETCONF-YANG が稼働するまで接続をブロックします。

iosp_client -f netconf_enable_passwordless guestshell<username> コマンドは、ゲストシェルアクセスに必要な SSH キーを作成します。

ゲストシェルからの NETCONF-YANG アクセスを削除するには、次のコマンドを使用します。

```
iosp_client -f netconf_disable guestshell and
iosp_client -f netconf_disable_passwordless guestshell  <username>
```

iosp_client -f netconf_disable guestshell コマンドは、ゲストシェル内から NETCONF へのアクセスを無効にします。ただし、NETCONF-YANG の設定は引き続き存在します。NETCONF-YANG をシャットダウンするには、**no netconf-yang** コマンドを使用します。

iosp_client -f netconf_disable_passwordless guestshell *username* コマンドは、指定されたユーザーの SSH キーを削除します。ユーザーはパスワードなしで NETCONF にアクセスすることはできません。ただし、パスワードを使用すれば接続できます。

netconf_enable_guestshell python API は、iosp_client 関数、iosp_client -f netconf_enable guestshell 830 および iosp_client -f netconf_enable_passwordless guestshell guesthell の組み合わせを実行します。この API は、unfamiliar-to-user iosp_client 関数を隠蔽します。この関数が呼び出されると、すべてのコマンドが完了するまで応答を返しません。関数がエラーを返さない限り、NETCONF が確実に実行されて、パスワードレスのセットアップが完了しており、接続の作成を開始できます。

ロギングとトレースのシステム フロー

ここでは、IOx のロギングとトレースの仕組みについて説明します。

LXC のロギング

1. ゲスト OS が、ゲストアプリケーションで **/dev/ttyS2** を有効にします。
2. ゲスト アプリケーションが、**/dev/ttyS2** にデータを書き込みます。
3. Libvirt が、ホストで **/dev/pts/x** への **/dev/ttyS2** をエミュレートします。

4. IOXMAN が、エミュレートされたシリアル デバイス **/dev/pts/x** を XML ファイルから取得します。
5. IOXMAN が、使用可能なデータを **/dev/pts/x** からリッスンして読み取り、メッセージのシビラティ（重大度）を設定して、メッセージをフィルタ処理し、解析してキューに格納します。
6. **errmsg** を使用してホストの **/dev/log** デバイスにメッセージを送信するタイマーが開始されます。
7. データが Cisco IOS syslog に保存されます。

KVM のロギング

1. ゲスト OS が、ゲストアプリケーションで **/dev/ttyS2** を有効にします。
2. ゲスト アプリケーションが、**/dev/ttyS2** にデータを書き込みます。
3. Libvirt が、ホストで **/dev/pts/x** への **/dev/ttyS2** をエミュレートします。
4. IOXMAN が、エミュレートされた TCP パスを XML ファイルから取得します。
5. IOXMAN が、UNIX ソケットを開き、リモートソケットに接続します。
6. IOXMAN が、使用可能なデータをソケットから読み取り、メッセージのシビラティ（重大度）を設定して、メッセージをフィルタ処理し、解析して、キューに格納します。
7. **errmsg** を使用してホストの **/dev/log** デバイスにメッセージを送信するタイマーが開始されます。
8. データが Cisco IOS syslog に保存されます。

LXC のトレース

1. ゲスト OS が、ゲストアプリケーションで **/dev/ttyS3** を有効にします。
2. メッセージを **/dev/ttyS3** にコピーするように **syslogd** を設定します。
3. ゲスト アプリケーションが、**/dev/ttyS3** にデータを書き込みます。
4. Libvirt が、ホストで **/dev/pts/y** への **/dev/ttyS3** をエミュレートします。
5. IOXMAN が、エミュレートされたシリアル デバイス **/dev/pts/y** を XML ファイルから取得します。
6. IOXMAN が、使用可能なデータを **/dev/pts/y** からリッスンして読み取り、フィルタ処理し、解析して、メッセージを IOx トレースログに保存します。
7. IOx トレースログが満杯の場合は、IOXMAN がトレースログ ファイルを **/bootflash/tracelogs** にローテーションします。

KVM のトレース

1. ゲスト OS が、ゲストアプリケーションで **/dev/ttyS3** を有効にします。
2. メッセージを **/dev/ttyS3** にコピーするように syslog を設定します。
3. ゲスト アプリケーションが、**/dev/ttyS3** にデータを書き込みます。
4. Libvirt が、ホストで TCP パスへの **/dev/ttyS3** をエミュレートします。
5. IOXMAN が、エミュレートされた TCP パスを XML ファイルから取得します。
6. IOXMAN が、UNIX ソケットを開き、リモートソケットに接続します。
7. IOXMANが、使用可能なデータをソケットから読み取り、メッセージのシビラティ（重大度）レベルを設定して、メッセージをフィルタ処理し、解析して、IOx トレースログに格納します。
8. IOx トレースログが満杯の場合は、IOXMANがトレースログ ファイルを **/bootflash/tracelogs** にローテーションします。

メッセージのロギングとトレース

ここでは、Cisco IOS syslog へのメッセージのロギングとトレースについて説明します。

Cisco IOS Syslog でのメッセージのロギング

ゲストアプリケーションから受信したどのロギングメッセージでも、IOXMAN はメッセージのシビラティ（重大度）をデフォルトで NOTICE に設定してから Cisco IOS syslog に送信します。IOSd で受信されたメッセージはコンソールに表示され、次のメッセージ形式で syslog に保存されます。

***Apr 7 00:48:21.911: %IM-5-IOX_INST_NOTICE:ioxman: IOX SERVICE guestshell LOG: Guestshell test**

Cisco IOS syslog に準拠するために、IOXMAN はロギングメッセージのシビラティ（重大度）レベルをサポートしています。シビラティ（重大度）のあるロギングメッセージを報告するには、ゲストアプリケーションでメッセージの先頭にヘッダーを追加する必要があります。

```
[a123b234,version,severity]

a123b234 is magic number.
Version:        severity support version.  Current version is 1.
Severity:       CRIT is 2
                ERR is 3
                WARN is 4
                NOTICE is 5
                INFO is 6
                DEBUG is 7
```

次に、メッセージログの例を示します。

```
echo "[a123b234,1,2]Guestshell failed" > /dev/ttyS2
```

ゲストアプリケーションから Cisco IOS syslog にロギングデータを報告するには、次の手順を実行します。

1. C プログラミングを使用している場合は、**write()** を使用してロギングデータをホストに送信します。

```
#define SYSLOG_TEST       “syslog test”
     int fd;
     fd = open("/dev/ttyS2", O_WRONLY);
     write(fd, SYSLOG_TEST, strlen(SYSLOG_TEST));
     close(fd);
```

2. シェルコンソールを使用している場合は、**echo** を使用してロギングデータをホストに送信します。

```
echo “syslog test” > /dev/ttyS2
```

IOx トレースログへのメッセージのトレース

ゲストアプリケーションから IOx トレースログにトレースメッセージを報告するには、次の手順を実行します。

1. C プログラミングを使用している場合は、**write()** を使用してトレースメッセージをホストに送信します。

```
#define SYSLOG_TEST       “tracelog test”
     int fd;
     fd = open("/dev/ttyS3", O_WRONLY);
     write(fd, SYSLOG_TEST, strlen(SYSLOG_TEST));
     close(fd);
```

2. C プログラミングを使用している場合は、**syslog()** を使用してトレースメッセージをホストに送信します。

```
#define SYSLOG_TEST       “tracelog test”

syslog(LOG_INFO, “%s\n”, SYSLOG_TEST);
```

3. シェルコンソールを使用している場合は、**echo** を使用してトレースデータをホストに送信します。

```
echo “tracelog test” > /dev/ttyS3
   or
logger “tracelog test”
```

ゲスト シェルを有効にする方法

IOx の管理

始める前に

IOxは開始まで最長で2分かかります。ゲストシェルを正常に有効にするには、CAF、IOXman、および Libvirtd サービスが実行している必要があります。

手順の概要

1. **enable**
2. **configure terminal**
3. **iox**
4. **exit**
5. **show iox-service**
6. **show app-hosting list**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**iox** 例： `Device(config)# iox`	IOx サービスを設定します。
ステップ 4	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 5	**show iox-service** 例： `Device# show iox-service`	IOx サービスのステータスを表示します。
ステップ 6	**show app-hosting list** 例：	デバイスに対して有効になっている app-hosting サービスのリストを表示します。

	コマンドまたはアクション	目的
	Device# show app-hosting list	

例

次に、**show iox-service** コマンドの出力例を示します。

```
Device# show iox-service

IOx Infrastructure Summary:
---------------------------
IOx service (CAF) 1.10.0.0 : Running
IOx service (HA)           : Running
IOx service (IOxman)       : Running
IOx service (Sec storage) : Not Running
Libvirtd   1.3.4           : Running
Dockerd    18.03.0         : Running
Application DB Sync Info  : Available
Sync Status              : Disabled
```

次に、**show app-hosting list** コマンドの出力例を示します。

```
Device# show app-hosting list

App id                                 State
---------------------------------------------------------
guestshell                             RUNNING
```

ゲスト シェルの管理

（注） VirtualPortGroups はルーティング プラットフォームでのみサポートされています。

始める前に

ゲスト シェルのアクセスが機能するには、IOx が構成されて実行している必要があります。IOx が構成されていない場合は、IOx の構成を求めるメッセージが表示されます。IOx を削除すると、ゲスト シェルにもアクセスできなくなります。ただし rootfs は影響を受けません。

ゲストシェルを有効にして操作するように、アプリケーションまたは管理インターフェイスも設定する必要があります。ゲストシェルのインターフェイスを有効にする方法の詳細については、「Configuring the AppGigabitEthernet Interface for Guest Shell」および「Enabling Guest Shell on the Management Interface」のセクションを参照してください。

手順の概要

1. **enable**

2. **guestshell enable**
3. **guestshell run** *linux-executable*
4. **guestshell run bash**
5. **guestshell disable**
6. **guestshell destroy**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**guestshell enable** 例： `Device# guestshell enable`	ゲスト シェル サービスの有効化。 （注） • **guestshell enable** コマンドは、ネットワーキングに管理 Virtual Routing and Forwarding（VRF）インスタンスを使用します。 • フロント パネル ネットワー キングに VirtualPortGroups（VPG）を使用している場合は、まず VPG を構成する必要があります。 • ゲスト IP アドレスとゲートウェイ IP アドレスは同じサブネット内にある必要があります。

ステップ 3	**guestshell run** *linux-executable*
	例：
	`Device# guestshell run python`
	または
	`Device# guestshell run python3`
		ゲスト シェルで Linux プログラムを実行します。
		（注） Cisco IOS XE Amsterdam 17.3.1 以降のリリースでは、Python バージョン 3 のみがサポートされます。

ステップ 4	**guestshell run bash** 例： `Device# guestshell run bash`	Bash シェルを開始して、ゲスト シェルにアクセスします。
ステップ 5	**guestshell disable** 例： `Device# guestshell disable`	ゲスト シェル サービスを無効化します。
ステップ 6	**guestshell destroy** 例：	ゲスト シェル サービスを非アクティブ化して、アンインストールします。

	コマンドまたはアクション	目的
	`Device# guestshell destroy`	

アプリケーション ホスティングを使用したゲストシェルの管理

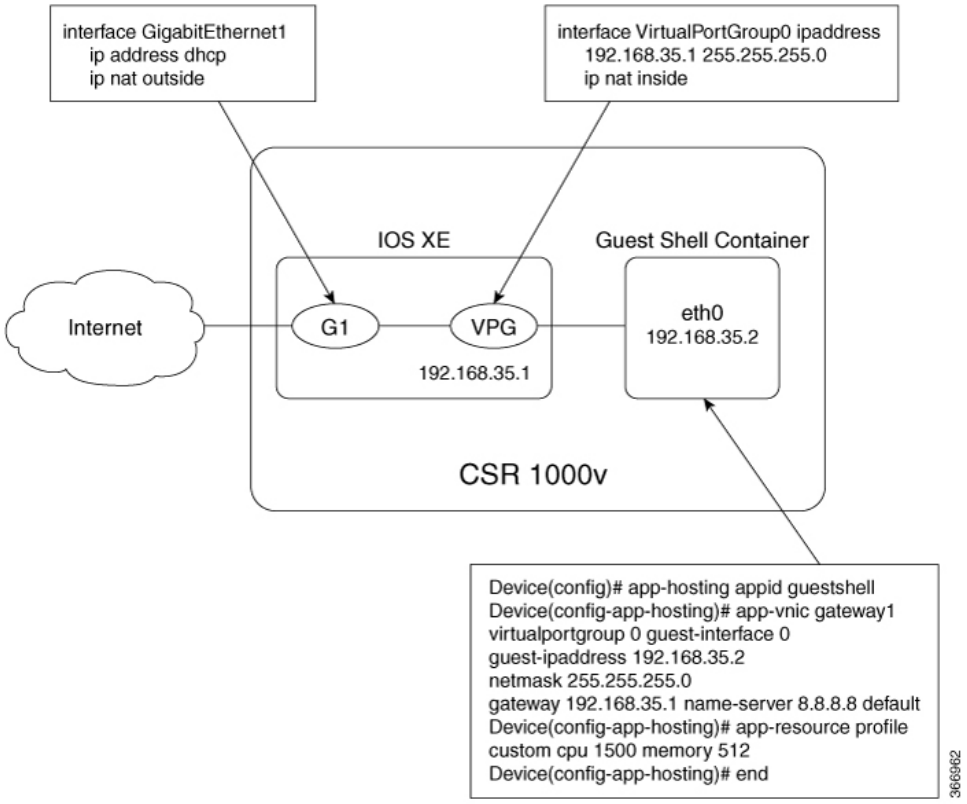
（注） この項は、シスコ ルーティング プラットフォームに適用されます。VirtualPortGroups は、Cisco Catalyst スイッチング プラットフォームではサポートされていません。

ゲスト シェルのアクセスが機能するには、IOx が構成されて実行している必要があります。IOx が構成されていない場合は、IOx の構成を求めるメッセージが表示されます。IOx を削除すると、ゲスト シェルにもアクセスできなくなります。ただし rootfs は影響を受けません。

（注） この手順（アプリケーションホスティングを使用したゲストシェルの管理）を使用して、Cisco IOS XE Fuji 16.7.1 以降のリリースのゲストシェルを有効にします。Cisco IOS XE Everest 16.6.x 以前では、ゲスト シェルの管理 （109 ページ） の手順を使用します。

```
Device(config)# interface GigabitEthernet1
Device(config-if)# ip address dhcp
Device(config-if)# ip nat outside
Device(config-if)# exit

Device(config-if)# interface VirtualPortGroup0
Device(config-if)# ip address 192.168.35.1 255.255.255.0
Device(config-if)# ip nat inside
Device(config-if)# exit

Device(config)# ip nat inside source list GS_NAT_ACL interface GigabitEthernet1 overload
Device(config)# ip access-list standard GS_NAT_ACL
Device(config)# permit 192.168.0.0 0.0.255.255

Device(config)# app-hosting appid guestshell
Device(config-app-hosting)# app-vnic gateway1 virtualportgroup 0 guest-interface 0
guest-ipaddress 192.168.35.2
netmask 255.255.255.0 gateway 192.168.35.1 name-server 8.8.8.8 default
Device(config-app-hosting)# app-resource profile custom cpu 1500 memory 512
Device(config-app-hosting)# end

Device# guestshell enable
Device# guestshell run python
```

図 *3:* アプリケーション ホスティングを使用したゲスト シェルの管理



前面パネルのネットワーキングでは、GigabitEthernet インターフェイスと VirtualPortGroup インターフェイスを上の図に示すように設定する必要があります。ゲストシェルは Virtualportgroup を送信元インターフェイスとして使用し、NAT を通じて外部ネットワークに接続します。

内部 NAT の設定には、次のコマンドを使用します。これにより、ゲスト シェルがインターネットに到達し、たとえば、Linux ソフトウェア更新プログラムを取得できるようになります。

```
ip nat inside source list
ip access-list standard
permit
```

上の例の **guestshell run** コマンドは Python 実行可能ファイルを実行します。また、**guestshell run** コマンドを使用して他の Linux 実行可能ファイルを実行することもできます。たとえば、**guestshell run bash** コマンドは bash シェルを起動し、**guestshell disable** コマンドはゲストシェルをシャットダウンして無効にします。後でシステムをリロードしても、ゲストシェルは無効のままになります。

ゲストシェルの AppGigabitEthernet インターフェイスの設定

（注） 次のタスクは、AppGigabitEthernet インターフェイスを持つ Catalyst スイッチにのみ適用されます。他のすべての Catalyst スイッチは、管理ポートを使用します。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface AppGigabitEthernet** *interface-number*
4. **switchport mode trunk**
5. **exit**
6. **app-hosting appid** *name*
7. **app-vnic AppGigabitEthernet trunk**
8. **vlan** *vlan-ID* **guest-interface** *guest-interface-number*
9. **guest-ipaddress** *ip-address* **netmask** *netmask*
10. **exit**
11. **exit**
12. **app-default-gateway** *ip-address* **guest-interface** *network-interface*
13. **nameserver#** *ip-address*
14. **end**
15. **guestshell enable**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**interface AppGigabitEthernet** *interface-number* 例： `Device(config)# interface AppGigabitEthernet 1/0/1`	AppGigabitEthernet インターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**switchport mode trunk** 例： `Device(config-if)# switchport mode trunk`	インターフェイスを永続的なトランキングモードに設定して、ネイバーリンクのトランクリンクへの変換をネゴシエートします。
ステップ 5	**exit** 例： `Device(config-if)# exit`	インターフェイス コンフィギュレーション モードを終了し、グローバルコンフィギュレーションモードに戻ります。
ステップ 6	**app-hosting appid** *name* 例： `Device(config)# app-hosting appid guestshell`	アプリケーションを設定し、アプリケーション ホスティング コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 7	**app-vnic AppGigabitEthernet trunk** 例： `Device(config-app-hosting)# app-vnic AppGigabitEthernet trunk`	トランクポートをアプリケーション ホスティングの前面パネルポートとして設定し、アプリケーション ホスティング トランク コンフィギュレーション モードを開始します。
ステップ 8	**vlan** *vlan-ID* **guest-interface** *guest-interface-number* 例： `Device(config-config-app-hosting-trunk)# vlan 4094 guest-interface 0`	VLAN ゲストインターフェイスを設定し、アプリケーション ホスティング VLAN アクセス IP コンフィギュレーション モードを開始します。
ステップ 9	**guest-ipaddress** *ip-address* **netmask** *netmask* 例： `Device(config-config-app-hosting-vlan-access-ip)# guest-ipaddress 192.168.2.2 netmask 255.255.255.0`	（オプション）静的 IP を設定します。
ステップ 10	**exit** 例： `Device(config-config-app-hosting-vlan-access-ip)# exit`	アプリケーション ホスティング VLAN アクセス IP コンフィギュレーション モードを終了し、アプリケーション ホスティング トランク コンフィギュレーション モードに戻ります。
ステップ 11	**exit** 例： `Device(config-config-app-hosting-trunk)# exit`	アプリケーション ホスティング トランク コンフィギュレーション モードを終了し、アプリケーション ホスティング コンフィギュレーション モードに戻ります。
ステップ 12	**app-default-gateway** *ip-address* **guest-interface** *network-interface* 例： `Device(config-app-hosting)# app-default-gateway 192.168.2.1 guest-interface 0`	デフォルトの管理ゲートウェイを設定します。
ステップ 13	**nameserver#** *ip-address* 例： `Device(config-app-hosting)# name-server0 172.16.0.1`	ドメインネームシステム（DNS）サーバを設定します。
ステップ 14	**end** 例： `Device(config-app-hosting)# end`	アプリケーション ホスティング コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 15	**guestshell enable** 例： `Device# guestshell enable`	ゲスト シェル サービスの有効化。

管理インターフェイスでのゲストシェルの有効化

（注） このタスクは、Cisco Catalyst 9200 シリーズ スイッチ、Cisco Catalyst 9300 シリーズ スイッチ、Cisco Catalyst 9400 シリーズ スイッチ、Cisco Catalyst 9500 シリーズ スイッチ、Cisco Catalyst 9600 シリーズ スイッチに適用できます。

手順の概要

1. **enable**
2. **configure terminal**
3. **app-hosting appid** *name*
4. **app-vnic management guest-interface** *interface-number*
5. **end**
6. **show app-hosting list**
7. **guestshell enable**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**app-hosting appid** *name* 例： `Device(config)# app-hosting appid guestshell`	アプリケーションを設定し、アプリケーション ホスティング コンフィギュレーション モードを開始します。
ステップ 4	**app-vnic management guest-interface** *interface-number* 例： `Device(config-app-hosting)# app-vnic management guest-interface 0`	仮想ネットワーク インターフェイスおよびゲスト インターフェイスの管理ゲートウェイを設定し、アプリケーションホスティング ゲートウェイ コンフィギュレーション モードを開始します。
ステップ 5	**end** 例： `Device(config-app-hosting-mgmt-gateway)# end`	アプリケーションホスティング管理ゲートウェイ コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 6	**show app-hosting list** 例：	インストールされているアプリケーションの現在のステータスを表示します。

	コマンドまたはアクション	目的
	`Device# show app-hosting list`	（注） ゲストシェルは、インストールされている場合にのみ、アプリケーションのリストに表示されます。
ステップ 7	**guestshell enable** 例： `Device# guestshell enable`	ゲスト シェル サービスの有効化。

ゲストシェルからの NETCONF アクセスの有効化と無効化

始める前に

ゲストシェル内から次のコマンドを初期化して、NETCONF-YANGアクセスを初期化します。

手順の概要

1. **iosp_client -f netconf_enable guestshell** *port-number*
2. **iosp_client -f netconf_enable_passwordless guestshell** *username*
3. **iosp_client -f netconf_disable guestshell**
4. **iosp_client -f netconf_disable_passwordless guestshell** *username*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**iosp_client -f netconf_enable guestshell** *port-number* 例： `Guest Shell: iosp_client -f netconf_enable guestshell 3`	**netconf-yang ssh local-vrf guestshell** コマンドを設定し、NETCONF-YANG が稼働するまで接続をブロックします。
ステップ 2	**iosp_client -f netconf_enable_passwordless guestshell** *username* 例： `Guest Shell: iosp_client -f netconf_enable guestshell guestshell`	ゲストシェルアクセスに必要なSSHキーを作成します。
ステップ 3	**iosp_client -f netconf_disable guestshell** 例： `GuestShell: iosp_client -f netconf_disable guestshell`	ゲストシェル内からNETCONFへのアクセスを削除します。 • NETCONF-YANG 設定は引き続き存在します。NETCONF-YANG をシャットダウンするには、**no netconf-yang** コマンドを使用します。
ステップ 4	**iosp_client -f netconf_disable_passwordless guestshell** *username*	指定したユーザーのアクセスキーを削除します。

	コマンドまたはアクション	目的
	例： `Guest Shell: iosp_client -f netconf_disable_passwordless guestshell guestshell`	• NETCONF アクセスはユーザーに対して引き続き有効です。ただし、ユーザーは NETCONF に接続するためにパスワードを使用する必要があります。

例

Python インタープリタのアクセス

Python はインタラクティブに使用できますが、Python スクリプトをゲスト シェルで実行することもできます。**guestshell run python** コマンドを使用してゲスト シェルで Python インタープリタを起動し、Python 端末を開きます。

（注） Cisco IOS XE Amsterdam 17.3.1 より前のリリースでは、Python V2 がデフォルトです。Cisco IOS XE Amsterdam 17.1.1 および Cisco IOS XE Amsterdam 17.2.1 では、Python V3 がサポートされています。Cisco IOS XE Amsterdam 17.3.1 以降のリリースでは、Python V3 がデフォルトです。

Cisco IOS XE Amsterdam 17.3.1 より前のリリース

guestshell run コマンドは、Linux 実行可能ファイルの実行に相当する Cisco IOS であり、Cisco IOS からの Python スクリプトの実行時に絶対パスを指定します。次の例は、コマンドの絶対パスを指定する方法を示しています。

```
Guestshell run python /flash/guest-share/sample_script.py parameter1 parameter2
```

次に、Cisco Catalyst 3650 シリーズ スイッチまたは Cisco Catalyst 3850 シリーズ スイッチで Python を有効にする例を示します。

```
Device# guestshell run python

Python 2.7.11 (default, March 16 2017, 16:50:55)
[GCC 4.7.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>>>
```

次の例は、Cisco ISR 4000 シリーズ サービス統合型ルータで Python を有効にする方法を示しています。

```
Device# guestshell run python

Python 2.7.5 (default, Jun 17 2014, 18:11:42)
[GCC 4.8.2 20140120 (Red Hat 4.8.2-16)] on linux2
```

```
Type "help", "copyright", "credits" or "license" for more information.
>>>>>
```

Cisco IOS XE Amsterdam 17.3.1 以降のリリース

次の例は、Cisco Catalyst 9000 シリーズ スイッチ上で Python を有効にする方法を示しています。

```
Device# guestshell run python3

Python 3.6.8 (default, Nov 21 2019, 22:10:21)
[GCC 8.3.1 20190507 (Red Hat 8.3.1-4)] on linux
Type "help", "copyright", "credits" or "license" for more information.>>>>>
```

ゲスト シェルの設定例

例：ゲスト シェルの管理

Cisco IOS XE Amsterdam 17.1.x から Cisco IOS XE Amsterdam 17.2.x

次の例は、ゲストシェルを有効にする方法を示しています。Cisco IOS XE Amsterdam 17.1.x および Cisco IOS XE Amsterdam 17.2.x では、Python V2.7 および Python V3.6 がサポートされています。ただし、これらのリリースでは Python V2.7 がデフォルトです。

```
Device> enable
Device# guestshell enable

Management Interface will be selected if configured
Please wait for completion
Guestshell enabled successfully

Device# guestshell run python
or
Device# guestshell run python3

Python 2.7.5 (default, Jun 17 2014, 18:11:42)
[GCC 4.8.2 20140120 (Red Hat 4.8.2-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>>>


Device# guestshell run bash

[guestshell@guestshell ~]$

Device# guestshell disable

Guestshell disabled successfully

Device# guestshell destroy
```

```
Guestshell destroyed successfully
```

Cisco IOS XE Amsterdam 17.3.1 以降のリリース

次の例は、ゲストシェルを有効にする方法を示しています。Cisco IOS XE Amsterdam 17.3.1 以降のリリースでは、Python V3.6 のみがサポートされます。

```
Device> enable
Device# guestshell enable

Management Interface will be selected if configured
Please wait for completion
Guestshell enabled successfully

Device# guestshell run python3

Python 3.6.8 (default, Nov 21 2019, 22:10:21)
[GCC 8.3.1 20190507 (Red Hat 8.3.1-4)] on linux
Type "help", "copyright", "credits" or "license" for more information.>>>>>

>>>>>

Device# guestshell run bash

[guestshell@guestshell ~]$

Device# guestshell disable

Guestshell disabled successfully

Device# guestshell destroy

Guestshell destroyed successfully
```

VirtualPortGroup 設定の例

（注） VirtualPortGroups は Cisco ルーティング プラットフォームでのみサポートされています。

ゲストシェル ネットワーキングに VirtualPortGroup インターフェイスを使用する場合、VirtualPortGroup インターフェイスには設定済みの静的 IP アドレスが必要です。フロントポートインターフェイスはインターネットに接続されている必要があり、ネットワーク アドレス変換（NAT）は VirtualPortGroup とフロント パネル ポートの間で設定されている必要があります。

次に示すのは、VirtualPortGroup の設定例です。

```
Device> enable
Device# configure terminal
```

```
Device(config)# interface VirtualPortGroup 0
Device(config-if)# ip address 192.168.35.1 255.255.255.0
Device(config-if)# ip nat inside
Device(config-if)# no mop enabled
Device(config-if)# no mop sysid
Device(config-if)# exit
Device(config)# interface GigabitEthernet 0/0/3
Device(config-if)# ip address 10.0.12.19 255.255.0.0
Device(config-if)# ip nat outside
Device(config-if)# negotiation auto
Device(config-if)# exit
Device(config)# ip route 0.0.0.0 0.0.0.0 10.0.0.1
Device(config)# ip route 10.0.0.0 255.0.0.0 10.0.0.1
!Port forwarding to use ports for SSH and so on.
Device(config)# ip nat inside source static tcp 192.168.35.2 7023 10.0.12.19 7023
extendable
Device(config)# ip nat outside source list NAT_ACL interface GigabitEthernet 0/0/3
overload
Device(config)# ip access-list standard NAT_ACL
Device(config-std-nacl)# permit 192.168.0.0 0.0.255.255
Device(config-std-nacl)# exit

! App-hosting configuration
Device(config)# app-hosting appid guestshell
Device(config-app-hosting)# app-vnic gateway1 virtualportgroup 0 guest-interface 0
guest-ipaddress 192.168.35.2
netmask 255.255.255.0 gateway 192.168.35.1 name-server 8.8.8.8 default
Device(config-app-hosting)# app-resource profile custom cpu 1500 memory 512
Device(config-app-hosting)# end

Device# guestshell enable
Device# guestshell run python
```

例：ゲストシェルの AppGigabitEthernet インターフェイスの設定

（注） 次のタスクは、AppGigabitEthernet インターフェイスを持つ Catalyst スイッチにのみ適用されます。他のすべての Catalyst スイッチは、管理ポートを使用します。

次の例は、ゲストシェルの AppGigabitEthernet インターフェイスを設定する方法を示しています。ここでは、VLAN 4094 がネットワークアドレス変換（NAT）を作成します。これはゲストシェルに使用されます。VLAN 1 は外部インターフェイスです。

```
Device> enable
Device# configure terminal
Device(config)# ip nat inside source list NAT_ACL interface vlan 1 overload
Device(config)# ip access-list standard NAT_ACL
Device(config-std-nacl)# permit 192.168.0.0 0.0.255.255
Device(config-std-nacl)# exit
Device(config)# vlan 4094
Device(config-vlan)# exit
Device(config)# interface vlan 4094
Device(config-if)# ip address 192.168.2.1 255.255.255.0
```

```
Device(config-if)# ip nat inside
Device(config-if)# exit
Device(config)# interface vlan 1
Device(config-if)# ip nat outside
Device(config-if)# exit
Device(config)# ip routing
Device(config)# ip route 0.0.0.0 0.0.0.0 209.165.201.1
Device(config)# interface AppGigabitEthernet 1/0/1
Device(config-if)# switchport mode trunk
Device(config-if)# exit
Device(config)# app-hosting appid guestshell
Device(config-app-hosting)# app-vnic AppGigEthernet trunk
Device(config-config-app-hosting-trunk)# vlan 4094 guest-interface 0
Device(config-config-app-hosting-vlan-access-ip)# guest-ipaddress 192.168.2.2 netmask
255.255.255.0
Device(config-config-app-hosting-vlan-access-ip)# exit
Device(config-config-app-hosting-trunk)# exit
Device(config-app-hosting)# app-default-gateway 192.168.2.1 guest-interface 0
Device(config-app-hosting)# name-server0 172.16.0.1
Device(config-app-hosting)# name-server1 198.51.100.1
Device(config-app-hosting)# end
Device# guestshell enable
```

例：管理インターフェイスでのゲストシェルの有効化

この例は、Cisco Catalyst 9200 シリーズ スイッチ、Cisco Catalyst 9300 シリーズ スイッチ、Cisco Catalyst 9400 シリーズ スイッチ、Cisco Catalyst 9500 シリーズ スイッチ、Cisco Catalyst 9600 シリーズ スイッチに適用できます。

```
Device> enable
Device# configure terminal
Device(config)# app-hosting appid guestshell
Device(config-app-hosting)# app-vnic management guest-interface 0
Device(config-app-hosting-mgmt-gateway)# end
Device# guestshell enable
```

例：ゲスト シェルの使用

ゲスト シェル プロンプトから Linux のコマンドを実行できます。次の例は、一部の Linux コマンドの使用法を示しています。

```
[guestshell@guestshell~]$ pwd
/home/guestshell

[guestshell@guestshell~]$ whoami
guestshell

[guestshell@guestshell~]$ uname -a
Linux guestshell 5.4.85 #1 SMP Tue Dec 22 10:50:44 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
```

Cisco 4000 シリーズ サービス統合型ルータは、CentOS Linux リリース 7.1.1503 で提供される **dohost** を使用します。

（注） **dohost** コマンドには、**ip http server** コマンドがデバイス上で設定されていることが必要です。

例：ゲスト シェルのネットワーキング設定

ゲスト シェルのネットワーキングでは、次の設定が必要です。

- ドメイン ネーム システム（DNS）の設定
- プロキシの設定
- プロキシの設定を使用するための YUM または PIP の設定

ゲスト シェルの DNS 設定の例

ゲスト シェルのサンプル DNS 構成は次のとおりです。

```
[guestshell@guestshell ~]$ cat/etc/resolv.conf
nameserver 192.0.2.1

Other Options:
[guestshell@guestshell ~]$ cat/etc/resolv.conf
domain cisco.com
search cisco.com
nameserver 192.0.2.1
search cisco.com
nameserver 198.51.100.1
nameserver 172.16.0.6
domain cisco.com
nameserver 192.0.2.1
nameserver 172.16.0.6
nameserver 192.168.255.254
```

例：プロキシ環境変数の設定

ネットワークがプロキシの背後にある場合は、Linux でプロキシ変数を設定します。必要な場合は、環境にこれらの変数を追加します。

次の例は、プロキシ変数を設定する方法を示しています。

```
[guestshell@guestshell ~]$cat /bootflash/proxy_vars.sh
export http_proxy=http://proxy.example.com:80/
export https_proxy=http://proxy.example.com:80/
```

```
export ftp_proxy=http://proxy.example.com:80/
export no_proxy=example.com
export HTTP_PROXY=http://proxy.example.com:80/
export HTTPS_PROXY=http://proxy.example.com:80/
export FTP_PROXY=http://proxy.example.com:80/
guestshell ~] source /bootflash/proxy_vars.sh
```

例：プロキシ設定用の Yum および PIP の構成

次の例は、プロキシ環境変数の設定に Yum を使用する方法を示しています。

```
cat /etc/yum.conf | grep proxy
[guestshell@guestshell~]$ cat/bootflash/yum.conf | grep proxy
proxy=http://proxy.example.com:80/
```

PIP のインストールでは、プロキシ設定に使用される環境変数が選択されます。PIP インストールには -E オプションを指定した sudo を使用します。環境変数が設定されていない場合は、次の例に示すように PIP コマンドでそれらを明示的に定義します。

```
sudo pip --proxy http://proxy.example.com:80/install requests
sudo pip install --trusted-bost pypi.example.com --index-url
http://pypi.example.com/simple requests
```

次の例では、Python の PIP インストールを使用する方法を示します。

```
Sudo -E pip install requests
[guestshell@guestshell ~]$ python
Python 2.17.11 (default, Feb 3 2017, 19:43:44)
[GCC 4.7.0] on linux2
Type "help", "copyright", "credits" or "license" for more information
>>>import requests
```

ゲスト シェルに関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
Python モジュール	『CLI Python モジュール』
ゼロ タッチ プロビジョニング	『ゼロ タッチ プロビジョニング』

MIB

MB	MIB のリンク
	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

ゲスト シェルの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *8 :* ゲスト シェルの機能情報

機能名	リリース	機能情報
ゲスト シェル	Cisco IOS XE Everest 16.5.1a Cisco IOS XE Everest 16.5.1b	ゲスト シェルは、お客様がシスコ スイッチの自動制御および管理のためのカスタム Python アプリケーションを実行できる、埋め込み Linux 環境であるセキュア コンテナです。システムの自動化されたプロビジョニングも含まれます。このコンテナ シェルは、ホスト デバイスから分離された安全な環境を提供します。ユーザはそこで、スクリプトまたはソフトウェア パッケージをインストールし、実行することができます。 Cisco IOS XE Everest 16.5.1a では、この機能は次のプラットフォームに実装されていました。 • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS Everest 16.5.1b では、この機能は次のプラットフォームに実装されていました。 • Cisco 4000 シリーズ サービス統合型ルータ
	Cisco IOS XE Everest 16.6.2	この機能は、Cisco IOS XE Everest 16.6.2 で、Cisco Catalyst 9400 シリーズ スイッチに実装されました。

機能名	リリース	機能情報
	Cisco IOS XE Fuji 16.7.1	Cisco IOS XE Fuji 16.7.1 では、この機能は次のプラットフォームに実装されていました。 • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco Cloud Services Router 1000V シリーズ Cisco IOS XE Fuji 16.7.1 では、ゲスト シェル機能の場合、ロギングとトレーシング サポートが Cisco ASR 1000 アグリゲーション サービス ルータに実装されました。
	Cisco IOS XE Fuji 16.8.1	Cisco IOS XE Fuji 16.8.1 では、この機能は Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチに実装されていました。
	Cisco IOS XE Fuji 16.9.1	Cisco IOS XE Fuji 16.9.1 では、この機能は Cisco 1000 シリーズ サービス統合型ルータに実装されていました。
	Cisco IOS XE Gibraltar 16.11.1b	Cisco IOS XE Gibraltar 16.11.1b では、この機能は次のプラットフォームに実装されていました。 • Cisco Catalyst 9800-40 ワイヤレスコントローラ • Cisco Catalyst 9800-80 ワイヤレスコントローラ
	Cisco IOS XE Gibraltar 16.12.1	

機能名	リリース	機能情報
		この機能は、Cisco IOS XE Gibraltar 16.12.1 で次のプラットフォームに実装されました。 • Cisco Catalyst 9200 シリーズ スイッチ （注） この機能は C9200L SKU ではサポートされていません。 • Cisco Catalyst 9300L SKU • Cisco Catalyst 9600 シリーズ スイッチ
	Cisco IOS XE Amsterdam 17.3.1	この機能は、Cisco IOS XE Amsterdam 17.3.1 で次のプラットフォームに実装されました。 • Cisco Catalyst 8200 シリーズエッジプラットフォーム • Cisco Catalyst 8300 シリーズエッジプラットフォーム • Cisco Catalyst 8500 および 8500L シリーズ エッジ プラットフォーム

機能名	リリース	機能情報
ゲストシェルからのNETCONFアクセス	Cisco IOS XE Bengaluru 17.6.1	NETCONF にはゲストシェル内からアクセスできるため、ユーザーは Python スクリプトを実行し、NETCONF プロトコルを使用してシスコカスタムパッケージ CLI を呼び出すことができます。 この機能は、17.6.1 で次のプラットフォームに実装されました。 • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 および 9300L シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 および 9500 ハイ パフォーマンス シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ
ゲストシェルでの Python 3 のサポート	Cisco IOS XE Amsterdam 17.1.1	Python バージョン 3.6 は、ゲストシェルでサポートされています。Python バージョン 3.6 は、すべてのサポート対象プラットフォームで使用できます。

netconf-yang ssh local-vrf guestshell

ゲストシェル内から SSH 接続を介した NETCONF-YANG アクセスを有効にするには、グローバル コンフィギュレーション モードで **netconf-yang ssh local-vrf guestshell** コマンドを使用します。NETCONF-YANG アクセスを無効にするには、このコマンドの **no** 形式を使用します。

netconf-yang ssh local-vrf guestshell *port-number*
no netconf-yang ssh local-vrf guestshell *port-number*

構文の説明

port-number	NETCONFアクセス用のポート番号。

コマンド デフォルト ゲストシェルからの NETCONF アクセスが無効化されます。

コマンド モード グローバル コンフィギュレーション（config）

コマンド履歴

リリース	変更内容
Cisco IOS XE Bengaluru 17.6.1	このコマンドが導入されました。

使用上のガイドライン ゲストシェル内からの NETCONF-YANG アクセスを有効にするには、ゲストシェルプロンプトで次のコマンドを実行する必要があります。

- **iosp_client -f netconf_enable guestshell** *port-number*
- **iosp_client -f netconf_enable_passwordless guestshell** *username*

iosp_client -f netconf_enable guestshell *port-number* コマンドは、**netconf-yang ssh local-vrf guestshell** コマンドを設定し、NETCONF-YANG が使用可能になるまで接続をブロックします。**iosp_client -f netconf_enable_passwordless guestshell** *username* コマンドは、ゲストシェルアクセス用の SSH キーを生成します。

例

次の例は、ゲストシェルからの NETCONF-YANG アクセスを有効にする方法を示しています。

```
Device> enable
Device# configure terminal
Device(config)# netconf-yang ssh local-vrf guestshell 803
```

netconf-yang ssh port disable

NETCONF-YANG のすべての外部接続を無効にするには、グローバル コンフィギュレーション モードで **netconf-yang ssh port disable** コマンドを使用します。

netconf-yang ssh port disable

このコマンドには引数またはキーワードはありません。

コマンド デフォルト 外部ポートは有効です。

コマンド モード グローバル コンフィギュレーション（config）

コマンド履歴

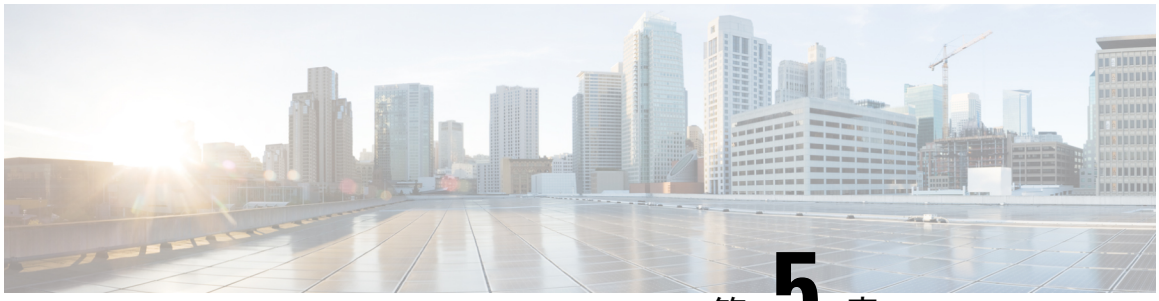
リリース	変更内容
Cisco IOS XE Bengaluru 17.6.1	このコマンドが導入されました。

使用上のガイドライン このコマンドは、外部ポートを閉じます。ゲストシェルに使用される接続などの内部接続のみが開いたままになります。

例

次に、NETCONF-YANG の外部接続を無効にする例を示します。

```
Device> enable
Device# configure terminal
Device(config)# netconf-yang ssh port-disable
```



第 5 章

Python API

Python プログラマビリティは、Python API をサポートしています。

• Python の使用 （131 ページ）

Python の使用

Cisco Python モジュール

シスコが提供する Python モジュールでは、EXEC および設定コマンドを実行するアクセス権が提供されます。 **help()** コマンドを入力すると、Cisco Python モジュールの詳細が表示されます。 **help()** コマンドは Cisco CLI モジュールのプロパティを表示します。

次の例は、Cisco Python モジュールに関する情報を示します。

```
Device# guestshell run python

Python 2.7.5 (default, Jun 17 2014, 18:11:42)
[GCC 4.8.2 20140120 (Red Hat 4.8.2-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> >>> from cli import cli,clip,configure,configurep, execute, executep
>>> help(configure)
Help on function configure in module cli:

configure(configuration)
Apply a configuration (set of Cisco IOS CLI config-mode commands) to the device
and return a list of results.

configuration = '''interface gigabitEthernet 0/0
no shutdown'''

# push it through the Cisco IOS CLI.
try:
results = cli.configure(configuration)
print "Success!"
except CLIConfigurationError as e:
print "Failed configurations:"
for failure in e.failed:
print failure
```

```
Args:
configuration (str or iterable): Configuration commands, separated by newlines.

Returns:
list(ConfigResult): A list of results, one for each line.

Raises:
CLISyntaxError: If there is a syntax error in the configuration.
```

```
>>> help(configurep)
Help on function configurep in module cli:

configurep(configuration)
Apply a configuration (set of Cisco IOS CLI config-mode commands) to the device
and prints the result.

configuration = '''interface gigabitEthernet 0/0
no shutdown'''

# push it through the Cisco IOS CLI.
configurep(configuration)

Args:
configuration (str or iterable): Configuration commands, separated by newlines.
```

```
>>> help(execute)
Help on function execute in module cli:

execute(command)
Execute Cisco IOS CLI exec-mode command and return the result.

command_output = execute("show version")

Args:
command (str): The exec-mode command to run.

Returns:
str: The output of the command.

Raises:
CLISyntaxError: If there is a syntax error in the command.
```

```
>>> help(executep)
Help on function executep in module cli:

executep(command)
Execute Cisco IOS CLI exec-mode command and print the result.

executep("show version")

Args:
command (str): The exec-mode command to run.
```

```
>>> help(cli)
Help on function cli in module cli:

cli(command)
    Execute Cisco IOS CLI command(s) and return the result.

    A single command or a delimited batch of commands may be run. The
    delimiter is a space and a semicolon, " ;". Configuration commands must be
    in fully qualified form.
```

```
    output = cli("show version")
    output = cli("show version ; show ip interface brief")
    output = cli("configure terminal ; interface gigabitEthernet 0/0 ; no shutdown")

    Args:
        command (str): The exec or config CLI command(s) to be run.

    Returns:
        string: CLI output for show commands and an empty string for
            configuration commands.

    Raises:
        errors.cli_syntax_error: if the command is not valid.
        errors.cli_exec_error: if the execution of command is not successful.


>>> help(clip)
Help on function clip in module cli:

clip(command)
    Execute Cisco IOS CLI command(s) and print the result.

    A single command or a delimited batch of commands may be run. The
    delimiter is a space and a semicolon, " ;". Configuration commands must be
    in fully qualified form.

    clip("show version")
    clip("show version ; show ip interface brief")
    clip("configure terminal ; interface gigabitEthernet 0/0 ; no shutdown")

    Args:
        command (str): The exec or config CLI command(s) to be run.
```

IOS CLI コマンドを実行するための Cisco Python モジュール

（注） Python を実行するには、ゲスト シェルが有効である必要があります。詳細については、「ゲスト シェル」の章を参照してください。

Python プログラミング言語は CLI コマンドを実行できる 6 つの関数を使用します。これらの関数は、Python CLI モジュールから利用できます。これらの関数を使用するには、**import cli** コマンドを実行します。これらの関数が機能するには、**ip http server** コマンドが有効になっている必要があります。

これらの関数の引数は CLI コマンドの文字列です。Python インタープリタ経由で CLI コマンドを実行するには、次の 6 つの関数のいずれかの引数文字列として CLI コマンドを入力します。

- **cli.cli(command)**：この関数は IOS コマンドを引数として取り、IOS パーサーからコマンドを実行し、結果のテキストを返します。このコマンドの形式が正しくない場合、Python の例外が発生します。次に、**cli.cli(command)** 関数の出力例を示します。

```
>>> import cli
>>> cli.clip('configure terminal; interface loopback 10; ip address
```

```
 10.10.10.10 255.255.255.255')
*Mar 13 18:39:48.518: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback10,
changed state to up
>>> cli.clip('show clock')
'\n*18:11:53.989 UTC Mon Mar 13 2017\n'
>>> output=cli.cli('show clock')
>>> print(output)
*18:12:04.705 UTC Mon Mar 13 2017
```

- **cli.clip(command)**：この関数は **cli.cli(command)** 関数と機能はまったく同じです。ただし結果のテキストを（返すのではなく）stdout に出力する点が異なります。次に、**cli.clip(command)** 関数の出力例を示します。

```
>>> cli
>>> cli.clip('configure terminal; interface loopback 11; ip address
 10.11.11.11 255.255.255.255')
*Mar 13 18:42:35.954: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback11,
changed state to up
*Mar 13 18:42:35.954: %LINK-3-UPDOWN: Interface Loopback11, changed state to up
>>> cli.clip('show clock')
*18:13:35.313 UTC Mon Mar 13 2017
>>> output=cli.clip('show clock')
*18:19:26.824 UTC Mon Mar 13 2017
>>> print (output)
None
```

- **cli.execute(command)**：この関数は単一の EXEC コマンドを実行して出力を返します。ただし結果のテキストは出力しません。このコマンドの一部としてセミコロンまたは改行を使用することは許可されません。この関数を複数回実行するには、for-loop が指定された Python リストを使用します。次に、 **cli.execute(command)**

  関数の出力例を示します。

```
>>> cli.execute("show clock")
'15:11:20.816 UTC Thu Jun 8 2017'
>>>
>>> cli.execute('show clock'; 'show ip interface brief')
  File "<stdin>", line 1
    cli.execute('show clock'; 'show ip interface brief')
                            ^
SyntaxError: invalid syntax
>>>
```

- **cli.executep(command)**：この関数は単一のコマンドを実行して、結果のテキストを（返すのではなく）stdout に出力します。次に、**cli.executep(command)** 関数の出力例を示します。

```
>>> cli.executep('show clock')
*18:46:28.796 UTC Mon Mar 13 2017
>>> output=cli.executep('show clock')
*18:46:36.399 UTC Mon Mar 13 2017
>>> print(output)
```

```
None
```

- **cli.configure(command)**：この関数は、コマンドで使用できる設定によりデバイスを設定します。これは次に示すように、コマンドとその結果が含まれる名前付きタプルのリストを返します。

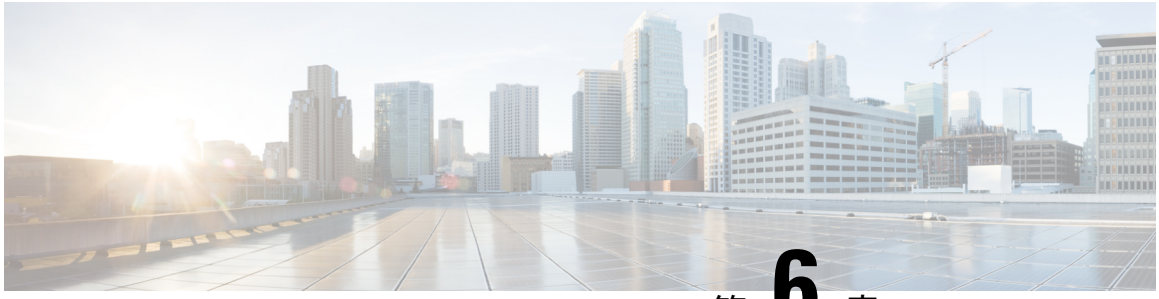
```
[Think: result = (bool(success), original_command, error_information)]
```

コマンド パラメータは複数行に入力することができ、**show running-config** コマンドの出力に表示されているのと同じ形式にすることができます。次に、**cli.configure(command)** 関数の出力例を示します。

```
>>>cli.configure(["interface GigabitEthernet1/0/7", "no shutdown",
"end"])
[ConfigResult(success=True, command='interface GigabitEthernet1/0/7',
line=1, output='', notes=None), ConfigResult(success=True, command='no shutdown',
line=2, output='', notes=None), ConfigResult(success=True, command='end',
line=3, output='', notes=None)]
```

- **cli.configurep(command)**：この関数は **cli.configure(command)** 関数と機能はまったく同じです。ただし結果のテキストを（返すのではなく）stdout に出力する点が異なります。次に、**cli.configurep(command)** 関数の出力例を示します。

```
>>> cli.configurep(["interface GigabitEthernet1/0/7", "no shutdown",
 "end"])
Line 1 SUCCESS: interface GigabitEthernet1/0/7
Line 2 SUCCESS: no shut
Line 3 SUCCESS: end
```



第 6 章

CLI Python モジュール

Python プログラマビリティでは、CLI を使用して IOS と対話できる Python モジュールを提供しています。

Python CLI モジュールについて

Python について

Cisco IOS XE デバイスは、ゲスト シェル内でインタラクティブおよび非インタラクティブ（スクリプト）の両方のモードで Python バージョン 2.7 をサポートします。Python スクリプト機能により、デバイスの CLI にプログラムを使用してアクセスして、さまざまなタスク、およびゼロ タッチ プロビジョニングまたは Embedded Event Manager（EEM）アクションを実行することができます。

Python スクリプトの概要

Python は、仮想化された Linux ベースの環境であるゲスト シェルで実行されます。詳細については、「ゲスト シェル」の章を参照してください。シスコが提供する Python モジュールは、ユーザの Python スクリプトがホスト デバイス上で IOS CLI コマンドを実行することを可能にします。

対話形式の Python プロンプト

デバイス上で **guestshell run python** コマンドを実行すると、ゲスト シェル内で、対話形式の Python プロンプトが開きます。Python の対話モードでは、Cisco Python CLI モジュールから Python 機能を実行してデバイスを設定することができます。

次の例は、対話形式の Python プロンプトを有効にする方法を示しています。

```
Device# guestshell run python

Python 2.7.5 (default, Jun 17 2014, 18:11:42)
[GCC 4.8.2 20140120 (Red Hat 4.8.2-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>

Device#
```

Python スクリプト

Python スクリプト名を引数として Python コマンドで使用することで、Python スクリプトを非インタラクティブ モードで実行できます。Python スクリプトは、ゲスト シェル内からアクセス可能である必要があります。ゲスト シェルから Python スクリプトにアクセスするには、ゲスト シェル内にマウントされているブートフラッシュまたはフラッシュにスクリプトを保存します。

（注） Python で **import cli** が機能するように、**ip http server** コマンドを設定する必要があります。

次の Python スクリプトの例は、さまざまな CLI 関数を使用して **show** コマンドを設定および出力します。

```
Device# more flash:sample_script.py


import sys
import cli

intf= sys.argv[1:]
intf = ''.join(intf[0])

print "\n\n *** Configuring interface %s with 'configurep' function  *** \n\n" %intf
cli.configurep(["interface loopback55","ip address 10.55.55.55 255.255.255.0","no
shut","end"])

print "\n\n *** Configuring interface %s with 'configure' function  *** \n\n"
cmd='interface %s,logging event link-status ,end' % intf
cli.configure(cmd.split(','))

print "\n\n *** Printing show cmd with 'executep' function  *** \n\n"
cli.executep('show ip interface brief')

print "\n\n *** Printing show cmd with 'execute' function  *** \n\n"
output= cli.execute('show run interface %s' %intf)
print (output)

print "\n\n *** Configuring interface %s with 'cli' function  *** \n\n"
cli.cli('config terminal; interface %s; spanning-tree portfast edge default' %intf)

print "\n\n *** Printing show cmd with 'clip' function  *** \n\n"
cli.clip('show run interface %s' %intf)


To run a Python script from the Guest Shell, execute the guestshell run python
```

```
/flash/script.py command
at the device prompt.
The following example shows how to run a Python script from the Guest Shell:
```

次の例は、ゲスト シェルから Python スクリプトを実行する方法を示しています。

```
Device# guestshell run python /flash/sample_script.py loop55

 *** Configuring interface loop55 with 'configurep' function  ***

Line 1 SUCCESS: interface loopback55
Line 2 SUCCESS: ip address 10.55.55.55 255.255.255.0
Line 3 SUCCESS: no shut
Line 4 SUCCESS: end

 *** Configuring interface %s with 'configure' function  ***


 *** Printing show cmd with 'executep' function  ***

Interface              IP-Address      OK? Method Status                Protocol
Vlan1                  unassigned      YES NVRAM  administratively down down
GigabitEthernet0/0     192.0.2.1       YES NVRAM  up                    up
GigabitEthernet1/0/1   unassigned      YES unset  down                  down
GigabitEthernet1/0/2   unassigned      YES unset  down                  down
GigabitEthernet1/0/3   unassigned      YES unset  down                  down
    :
         :
         :
Te1/1/4                unassigned      YES unset  down                  down
Loopback55             10.55.55.55     YES TFTP   up                    up
Loopback66             unassigned      YES manual up                    up


 *** Printing show cmd with 'execute' function  ***

Building configuration...
Current configuration : 93 bytes
!
interface Loopback55
 ip address 10.55.55.55 255.255.255.0
 logging event link-status
end

 *** Configuring interface %s with 'cli' function  ***

 *** Printing show cmd with 'clip' function  ***

Building configuration...
Current configuration : 93 bytes
!
interface Loopback55
 ip address 10.55.55.55 255.255.255.0
 logging event link-status
end
```

サポートされる Python のバージョン

ゲスト シェルは、Python バージョン 2.7 をプリインストールしています。ゲスト シェルは、仮想化された Linux ベースの環境であり、Cisco デバイスの自動制御と管理のための Python アプリケーションを含む、カスタム Linux アプリケーションを実行するように設計されています。Montavista CGE7 がインストールされたプラットフォームは Python バージョン 2.7.11 をサポートし、CentOS 7 がインストールされたプラットフォームは Python バージョン 2.7.5 をサポートします。

次の表は、Python の各バージョンおよびサポート対象のプラットフォームに関する情報を示しています。

表 *9 : Python* バージョン サポート

Python のバージョン	**プラットフォーム**
Python バージョン 2.7.5	Cisco Catalyst 3650 シリーズ スイッチおよび Cisco Catalyst 3850 シリーズ スイッチを除くすべてのサポート対象プラットフォーム。
Python バージョン 2.7.11	• Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ
Python バージョン 3.6	Cisco IOS XE Amsterdam 17.1.1 以降のリリースでサポートされています。 Cisco IOS XE Amsterdam 17.1.1 および Cisco IOS XE Amsterdam 17.2.1 では、Python V2 がデフォルトです。ただし、Cisco IOS XE Amsterdam 17.3.1 以降のリリースでは、Python V3 がデフォルトです。 （注） Cisco Catalyst 9200 シリーズ スイッチは、Cisco IOS XE Amsterdam 17.1.1 および Cisco IOS XE Amsterdam 17.2.1 で Python Version 3.6 をサポートしていません。Cisco Catalyst 9200 シリーズ スイッチは、Cisco IOS XE Amsterdam 17.3.1 以降のリリースで Python V3 をサポートしています。 （注） Cisco Catalyst 3650 シリーズ スイッチおよび Cisco Catalyst 3850 シリーズ スイッチではサポートされていません。

CentOS 7 がインストールされたプラットフォームは、オープン ソース リポジトリからの Redhat Package Manager（RPM）のインストールをサポートします。

Cisco CLI Python モジュールの更新

Cisco CLI Python モジュールおよび EEM モジュールは、デバイスにインストール済みです。ただし、Yum または事前にパッケージ化されているバイナリのいずれかを使用して Python のバージョンを更新する場合は、シスコが提供する CLI モジュールも更新する必要があります。

（注） Python バージョン 2 がすでにあるデバイスで Python バージョン 3 への更新を行うと、デバイス上には両方のバージョンの Python が存在するようになります。Python を実行するには、次の IOS コマンドのいずれかを使用します。

- **guestshell run python2** コマンドは、Python バージョン 2 を有効化します。
- **guestshell run python3** コマンドは、Python バージョン 3 を有効化します。
- **guestshell run python** コマンドは、Python バージョン 2 を有効化します。

Python のバージョンを更新するには、次の方法のいずれかを使用します。

- スタンドアロン tarball のインストール
- CLI モジュールのための PIP のインストール

CLI Python モジュールに関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
ゲスト シェル	ゲスト シェル
EEM Python モジュール	EEM の Python スクリプト

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

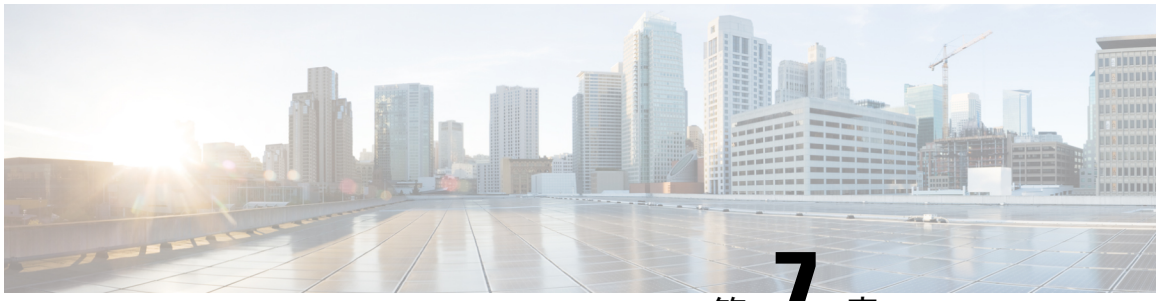
CLI Python モジュールの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *10 : CLI Python* モジュールの機能情報

機能名	リリース	機能情報
CLI Python モジュール	Cisco IOS XE Everest 16.5.1a	Python プログラマビリティでは、ユーザが CLI を使用して IOS と対話できるようにする Python モジュールが提供されます。 Cisco IOS XE Everest 16.5.1a では、この機能は次のプラットフォームに実装されていました。 • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.5.1b では、この機能は次のプラットフォームに実装されていました。 • Cisco 4000 シリーズ サービス統合型ルータ
	Cisco IOS XE Everest 16.6.2	この機能は、Cisco Catalyst 9400 シリーズ スイッチに実装されました。
	Cisco IOS XE Fuji 16.7.1	この機能は、次のプラットフォームに実装されていました。 • Cisco ASR 1000 アグリゲーション サービス ルータ • Cisco CSR 1000v シリーズ クラウド サービス ルータ



第 7 章

EEM Python モジュール

組み込みイベント マネージャ（EEM）ポリシーは、Python スクリプトをサポートします。Python スクリプトは、EEM アプレットで EEM アクションの一部として実行できます。

EEM Python モジュールの前提条件

ゲスト シェルは、コンテナ内で機能する必要があります。ゲスト シェルは、デフォルトでは有効になっていません。詳細については、ゲスト シェル機能の説明を参照してください。

EEM Python モジュールについて

EEM の Python スクリプト

組み込みイベント マネージャ（EEM）ポリシーは、Python スクリプトをサポートします。Python スクリプトを EEM ポリシーとして登録し、対応するイベントが発生したときに、登録済みの Python スクリプトを実行することができます。EEM Python スクリプトには、EEM TCL ポリシーと同じイベント仕様の構文があります。

設定済みの EEM ポリシーは、ゲスト シェル内で実行します。ゲスト シェルは、仮想化された Linux ベースの環境であり、Cisco デバイスの自動制御と管理のための Python アプリケーションを含む、カスタム Linux アプリケーションを実行するように設計されています。ゲスト シェル コンテナは、Python インタープリタを提供します。

EEM Python パッケージ

EEM Python パッケージを Python スクリプトにインポートすると、EEM に固有の拡張機能を実行できます。

（注） EEM Python パッケージは、EEM Python スクリプト内でのみ使用できます（パッケージは EEM に登録でき、スクリプトの最初の行に EEM イベント仕様が記載されます）。標準的な Python スクリプト（Python スクリプト名を使用して実行される）では使用できません。

Python パッケージには、次のアプリケーションプログラミングインターフェイス（API）が含まれています。

- アクション API：EEM アクションを実行するもので、デフォルトのパラメータがあります。
- CLI 実行 API：IOS コマンドを実行し、出力を返します。次に、CLI 実行 API のリストを示します。
  - eem_cli_open()
  - eem_cli_exec()
  - eem_cli_read()
  - eem_cli_read_line()
  - eem_cli_run()
  - eem_cli_run_interactive()
  - eem_cli_read_pattern()
  - eem_cli_write()
  - eem_cli_close()
- 環境変数にアクセスする API：組み込みまたはユーザ定義の変数のリストを取得します。次に、環境変数にアクセスする API を示します。
  - eem_event_reqinfo ()：組み込み変数のリストを返します。
  - eem_user_variables()：引数の現在の値を返します。

Python がサポートする EEM アクション

Python パッケージ（EEM スクリプト内でのみ使用可能で、標準的な Python スクリプトでは使用不可）では、次の EEM アクションをサポートしています。

- Syslog メッセージの印刷
- SNMP トラップの送信

- ボックスのリロード
- スタンバイ デバイスへの切り替え
- ポリシーの実行
- トラック オブジェクトの読み取り
- トラック オブジェクト セット
- Cisco ネットワーキング サービスのイベントの生成

EEM Python パッケージは、EEM アクションを実行するため、インターフェイスを公開します。これらのアクションは Python スクリプトを使用して呼び出すことができ、Cisco Plug N Play（PnP）経由で Python パッケージからアクション ハンドラに転送されます。

EEM 変数

EEM ポリシーは、次の種類の変数を持つことができます。

- イベント固有の組み込み変数：ポリシーをトリガーしたイベントの詳細が設定される事前定義の変数のセット。eem_event_reqinfo () API は、組み込み変数のリストを返します。これらの変数は、ローカルマシンに保存してローカル変数として使用することができます。ローカル変数に対する変更は、組み込み変数に反映されません。
- ユーザ定義の変数：定義およびポリシーでの使用が可能な変数。これらの変数の値は、Python スクリプト内で参照できます。スクリプトを実行する際に、変数の最新の値が使用可能であることを確認してください。eem_user_variables() API は、API で入力された引数の現在の値を返します。

EEM CLI ライブラリのコマンド拡張

EEM 内では、Python スクリプトを動作させるため、次の CLI ライブラリ コマンドを使用できます。

- eem_cli_close()：EXEC プロセスをクローズし、コマンドに接続された、VTY および指定されたチャネル ハンドラをリリースします。
- eem_cli_exec：指定されたチャネル ハンドラにコマンドを記述し、コマンドを実行します。次に、チャネルからコマンドの出力を読み取り、出力を返します。
- eem_cli_open：VTY を割り当て、EXEC CLI セッションを作成し、VTY をチャネル ハンドラに接続します。チャネル ハンドラを含む配列を返します。
- eem_cli_read()：読み取られている内容でデバイス プロンプトのパターンが発生するまで、指定された CLI のチャネル ハンドラからコマンド出力を読み取ります。一致するまで、読み取られたすべての内容を返します。
- eem_cli_read_line()：指定された CLI のチャネル ハンドラから、コマンド出力の 1 行を読み取ります。読み取られた行を返します。

- eem_cli_read_pattern()：読み取られている内容でパターンが発生するまで、指定された CLI のチャネル ハンドラからコマンド出力を読み取ります。一致するまで、読み取られたすべての内容を返します。
- eem_cli_run()：clist にある項目を繰り返し、それぞれが、イネーブル モードで実行されるコマンドであることを前提とします。正常に実行されると、実行されたすべてのコマンドの出力を返します。失敗すると、エラーを返します。
- eem_cli_run_interactive()：3 つの項目がある clist のサブリストを用意します。正常に実行されると、実行されたすべてのコマンドの出力を返します。失敗すると、エラーを返します。可能な場合には、配列も使用します。予測と応答を別々に保持することによって、より簡単に後で読み取ることができます。
- eem_cli_write()：指定された CLI チャネル ハンドラに対して実行されるコマンドを書き込みます。CLI チャネル ハンドラによって、コマンドが実行されます。

EEM Python ポリシーの設定方法

Python スクリプトが動作できるようにするには、ゲスト シェルを有効化する必要があります。詳細については、「ゲスト シェル」の章を参照してください。

Python ポリシーの登録

手順の概要

1. **enable**
2. **configure terminal**
3. **event manager directory user policy** *path*
4. **event manager policy** *policy-filename*
5. **exit**
6. **show event manager policy registered**
7. **show event manager history events**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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。

	コマンドまたはアクション	目的
ステップ 3	**event manager directory user policy** *path* 例： `Device(config)# event manager directory user policy flash:/user_library`	ユーザ ライブラリ ファイルまたはユーザ定義 EEM ポリシーの保存に使用するディレクトリを指定します。 （注） 指定されたパスにポリシーが必要です。たとえば、この手順では、eem_script.py ポリシーが flash:/user_library フォルダーまたはパスで使用できます。
ステップ 4	**event manager policy** *policy-filename* 例： `Device(config)# event manager policy eem_script.py`	EEM ポリシーを EEM に登録します。 • ポリシーは、ファイル拡張子に基づいて解析されます。ファイル拡張子は .py で、ポリシーは Python ポリシーとして登録されます。 • EEMは、ポリシーそのものに含まれるイベント仕様に基づいてポリシーをスケジューリングし、実行します。**event manager policy** コマンドが呼び出されると、EEMはポリシーを確認し、指定されたイベントが発生した場合に実行されるように登録します。
ステップ 5	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 6	**show event manager policy registered** 例： `Device# show event manager policy registered`	保留 EEM ポリシーを表示します。
ステップ 7	**show event manager history events** 例： `Device# show event manager history events`	トリガーされた EEM イベントを表示します。

例

次に、**show event manager policy registered** コマンドの出力例を示します。

```
Device# show event manager policy registered

No.  Class     Type    Event Type          Trap  Time Registered           Name
1    script    user    multiple            Off   Tue Aug 2 22:12:15 2016   multi_1.py
 1: syslog: pattern {COUNTER}
 2: none: policyname {multi_1.py} sync {yes}
 trigger delay 10.000
  correlate event 1 or event 2
  attribute tag 1 occurs 1
```

```
 nice 0 queue-priority normal maxrun 100.000 scheduler rp_primary Secu none


2    script    user    multiple            Off   Tue Aug 2 22:12:20 2016   multi_2.py
 1: syslog: pattern {COUNTER}
 2: none: policyname {multi_2.py} sync {yes}
 trigger
  correlate event 1 or event 2
 nice 0 queue-priority normal maxrun 100.000 scheduler rp_primary Secu none


3    script    user    multiple            Off   Tue Aug 2 22:13:31 2016   multi.tcl
 1: syslog: pattern {COUNTER}
 2: none: policyname {multi.tcl} sync {yes}
 trigger
  correlate event 1 or event 2
  attribute tag 1 occurs 1
 nice 0 queue-priority normal maxrun 100.000 scheduler rp_primary Secu none
```

EEM アプレット アクションの一部としての Python スクリプトの実行

Python スクリプト：eem_script.py

アクション コマンドを使用することで、EEM アプレットに Python スクリプトを含めることができます。この例では、ユーザは標準 Python スクリプトを EEM アクションの一部として実行しようとしています。ただし、EEM Python パッケージは標準 Python スクリプトでは使用できません。IOS の標準 Python スクリプトには from cli import cli,clip という名前のパッケージがあり、そのパッケージは IOS コマンドを実行するために使用できます。

```
import sys
from cli import cli,clip,execute,executep,configure,configurep

intf= sys.argv[1:]
intf = ''.join(intf[0])


print ('This script is going to unshut interface %s and then print show ip interface
brief'%intf)

if intf == 'loopback55':
configurep(["interface loopback55","no shutdown","end"])
else :
cmd='int %s,no shut ,end' % intf
configurep(cmd.split(','))

executep('show ip interface brief')
```

次に、**guestshell run python** コマンドの出力例を示します。

```
Device#  guestshell run python /flash/eem_script.py loop55

This script is going to unshut interface loop55 and then print show ip interface brief
Line 1 SUCCESS: int loop55
Line 2 SUCCESS: no shut
```

```
Line 3 SUCCESS: end
Interface IP-Address OK? Method Status Protocol
Vlan1 unassigned YES NVRAM administratively down down
GigabitEthernet0/0 5.30.15.37 YES NVRAM up up
GigabitEthernet1/0/1 unassigned YES unset down down
GigabitEthernet1/0/2 unassigned YES unset down down
GigabitEthernet1/0/3 unassigned YES unset down down
GigabitEthernet1/0/4 unassigned YES unset up up
GigabitEthernet1/0/5 unassigned YES unset down down
GigabitEthernet1/0/6 unassigned YES unset down down
GigabitEthernet1/0/7 unassigned YES unset down down
GigabitEthernet1/0/8 unassigned YES unset down down
GigabitEthernet1/0/9 unassigned YES unset down down
GigabitEthernet1/0/10 unassigned YES unset down down
GigabitEthernet1/0/11 unassigned YES unset down down
GigabitEthernet1/0/12 unassigned YES unset down down
GigabitEthernet1/0/13 unassigned YES unset down down
GigabitEthernet1/0/14 unassigned YES unset down down
GigabitEthernet1/0/15 unassigned YES unset down down
GigabitEthernet1/0/16 unassigned YES unset down down
GigabitEthernet1/0/17 unassigned YES unset down down
GigabitEthernet1/0/18 unassigned YES unset down down
GigabitEthernet1/0/19 unassigned YES unset down down
GigabitEthernet1/0/20 unassigned YES unset down down
GigabitEthernet1/0/21 unassigned YES unset down down
GigabitEthernet1/0/22 unassigned YES unset down down
GigabitEthernet1/0/23 unassigned YES unset up up
GigabitEthernet1/0/24 unassigned YES unset down down
GigabitEthernet1/1/1 unassigned YES unset down down
GigabitEthernet1/1/2 unassigned YES unset down down
GigabitEthernet1/1/3 unassigned YES unset down down
GigabitEthernet1/1/4 unassigned YES unset down down
Te1/1/1 unassigned YES unset down down
Te1/1/2 unassigned YES unset down down
Te1/1/3 unassigned YES unset down down
Te1/1/4 unassigned YES unset down down
Loopback55 10.55.55.55 YES manual up up

Device#
Jun 7 12:51:20.549: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback55,
changed state to up
Jun 7 12:51:20.549: %LINK-3-UPDOWN: Interface Loopback55, changed state to up
```

次に示すのは、syslog へのメッセージ出力のサンプル スクリプトです。このスクリプトは、ファイルに保存され、デバイス上のファイル システムにコピーされ、イベント マネージャのポリシー ファイルを使用して登録される必要があります。

```
::cisco::eem::event_register_syslog tag "1"    pattern     COUNTER    maxrun 200

import eem
import time

eem.action_syslog("SAMPLE SYSLOG MESSAGE","6","TEST")
```

次に示すのは、EEM 環境変数を出力するサンプル スクリプトです。このスクリプトは、ファイルに保存され、デバイス上のファイル システムにコピーされ、イベント マネージャのポリシー ファイルを使用して登録される必要があります。

```
::cisco::eem::event_register_syslog tag "1"    pattern     COUNTER    maxrun 200
```

```
import eem
import time

c = eem.env_reqinfo()


print "EEM Environment Variables"
for k,v in c.iteritems():
      print "KEY : " + k + str(" ---> ") + v

print "Built in Variables"
for i,j in a.iteritems() :
      print "KEY : " + i + str(" ---> ") + j
```

EEM アプレットでの Python スクリプトの追加

手順の概要

1. **enable**
2. **configure terminal**
3. **event manager applet** *applet-name*
4. **event** [**tag** *event-tag*] **syslog pattern** *regular-expression*
5. **action** *label* **cli command** *cli-string*
6. **action** *label* **cli command** *cli-string* [**pattern** *pattern-string*]
7. **end**
8. **show event manager policy active**
9. **show event manager history events**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**event manager applet** *applet-name* 例： `Device(config)# event manager applet interface_Shutdown`	Embedded Event Manager（EEM）にアプレットを登録し、アプレット コンフィギュレーション モードを開始します。
ステップ 4	**event** [**tag** *event-tag*] **syslog pattern** *regular-expression* 例：	syslog メッセージのパターン一致を実行する正規表現を指定します。

	コマンドまたはアクション	目的
	`Device(config-applet)# event syslog pattern "Interface Loopback55, changed state to administratively down"`	
ステップ 5	**action** *label* **cli command** *cli-string* 例： `Device(config-applet)# action 0.0 cli command "en"`	EEMアプレットがトリガーされたときに実行されるIOS コマンドを指定します。
ステップ 6	**action** *label* **cli command** *cli-string* [**pattern** *pattern-string*] 例： `Device(config-applet)# action 1.0 cli command "guestshell run python3 /bootflash/eem_script.py loop55"`	**pattern** キーワードで指定されるアクションを指定します。 • 次の要請プロンプトに一致する正規表現パターン文字列を指定します。
ステップ 7	**end** 例： `Device(config-applet)# end`	アプレット コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 8	**show event manager policy active** 例： `Device# show event manager policy active`	実行している EEM ポリシーを表示します。
ステップ 9	**show event manager history events** 例： `Device# show event manager history events`	トリガーされた EEM イベントを表示します。

次のタスク

次の例では、タスクに設定されているPythonスクリプトをトリガーする方法を示しています。

```
Device(config)# interface loopback 55
Device(config-if)# shutdown
Device(config-if)# end
Device#

Mar 13 10:53:22.358 EDT: %SYS-5-CONFIG_I: Configured from console by console
Mar 13 10:53:24.156 EDT: %LINK-5-CHANGED:  Line protocol on Interface Loopback55, changed
 state to down
Mar 13 10:53:27.319 EDT: %LINK-3-UPDOWN:  Interface Loopback55, changed state to
administratively down
Enter configuration commands, one per line.  End with CNTL/Z.
Mar 13 10:53:35.38 EDT:  %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback55,
changed state to up
*Mar 13 10:53:35.39 EDT %LINK-3-UPDOWN: Interface Loopback55, changed state to up
+++ 10:54:33 edi37(default) exec +++
show ip interface br
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0/0   unassigned      YES unset  down                  down
GigabitEthernet0/0/1   unassigned      YES unset  down                  down
GigabitEthernet0/0/2   10.1.1.31       YES DHCP   up                    up
```

```
GigabitEthernet0/0/3   unassigned      YES unset  down                  down
GigabitEthernet0       192.0.2.1       YES manual up                    up
Loopback55             198.51.100.1    YES manual up                    up
Loopback66             172.16.0.1      YES manual up                    up
Loopback77             192.168.0.1     YES manual up                    up
Loopback88             203.0.113.1     YES manual up                    up
```

EEM Python モジュールに関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
EEM 設定	『*Embedded Event Manager Configuration Guide*』
EEM コマンド	『*Embedded Event Manager Command Reference*』
ゲスト シェル設定	『ゲスト シェル』

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

EEM Python モジュールの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

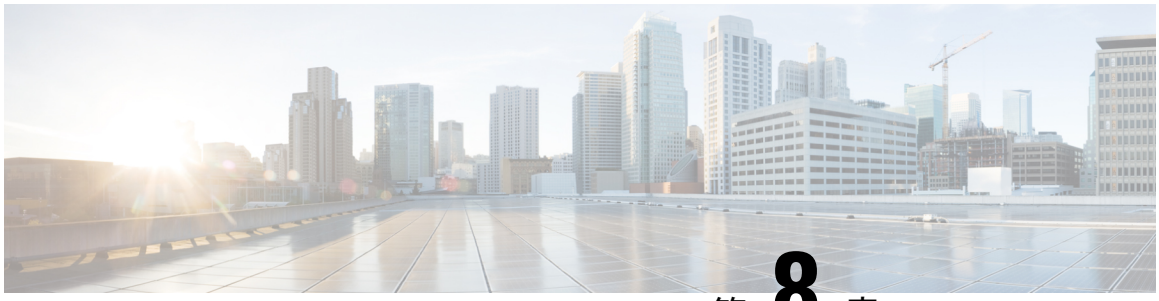
プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *11 : EEM Python* モジュールの機能情報

機能名	リリース	機能情報
EEM Python モジュール	Cisco IOS XE Everest 16.5.1a Cisco IOS XE Everest 16.5.1b	この機能は、EEM ポリシーとして Python スクリプトをサポートします。 追加された新規コマンドはありません。 Cisco IOS XE Everest 16.5.1a では、この機能は次のプラットフォームに実装されていました。 • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco IOS XE Everest 16.5.1b では、この機能は次のプラットフォームに実装されていました。 • Cisco ISR 4000 シリーズ サービス統合型ルータ
	Cisco IOS XE Everest 16.6.2	この機能は、Cisco IOS XE Everest 16.6.2 で、Cisco Catalyst 9400 シリーズ スイッチに実装されました。
	Cisco IOS XE Fuji 16.8.1a	Cisco IOS XE Fuji 16.8.1a では、この機能は Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチに実装されていました。

第 III 部

モデル駆動型プログラマビリティ



第 8 章

NETCONF プロトコル

NETCONF プロトコルの概要

データモデルの概要：プログラムによる設定と各種の標準規格に準拠した設定

ネットワーク デバイスを管理する従来の方法は、階層的データ（設定コマンド）および運用データ（show コマンド）用のコマンドライン インターフェイス（CLI）を使用することです。ネットワーク管理の場合、特にさまざまなネットワーク デバイス間で管理情報を交換するために、Simple Network Management Protocol（SNMP）が広く使用されています。頻繁に使用されている CLI と SNMP ですが、これにはいくつかの制約事項があります。CLI は非常に独自的であり、テキストベースの仕様を理解し、解釈するには人間の介入が必要です。SNMP は、階層的データと運用データを区別しません。

これを解決するには、手作業で設定作業を行うのではなく、プログラムを使用したり、各種の標準規格に準拠してネットワーク デバイスの設定を記述します。Cisco IOS XE で動作するネットワーク デバイスは、データ モデルを使用するネットワーク上の複数のデバイスの設定の自動化をサポートしています。データ モデルは、業界で定義された標準的な言語で開発され、ネットワークの設定とステータス情報を定義できます。

Cisco IOS XE は、Yet Another Next Generation（YANG）データ モデリング言語をサポートしています。YANG をネットワーク設定プロトコル（NETCONF）で使用すると、自動化されたプログラミング可能なネットワーク操作の望ましいソリューションが実現します。NETCONF（RFC 6241）は、クライアント アプリケーションがデバイスからの情報を要求してデバイス

に設定変更を加えるために使用するXMLベースのプロトコルです。YANGは主に、NETCONF操作で使用される設定とステート データをモデル化するために使用されます。

Cisco IOS XE では、モデル ベースのインターフェイスは、既存のデバイス CLI、Syslog、およびSNMPインターフェイスと相互運用します。必要に応じて、これらのインターフェイスは、ネットワーク デバイスからノースバウンドに公開されます。YANG は、RFC 6020 に基づいて各プロトコルをモデル化するために使用されます。

（注） 開発者に分かりやすい方法で Cisco YANG モデルにアクセスするには、GitHub リポジトリを複製し、vendor/cisco サブディレクトリに移動します。ここでは、IOS XE、IOS-XR、および NX-OS プラットフォームのさまざまなリリースのモデルを使用できます。

NETCONF

NETCONFは、ネットワーク デバイスの設定をインストール、操作、削除するためのメカニズムです。

コンフィギュレーション データとプロトコル メッセージにExtensible Markup Language（XML）ベースのデータ符号化を使用します。

NETCONFはシンプルなリモート プロシージャ コール（RPC）ベースのメカニズムを使用してクライアントとサーバ間の通信を促進します。クライアントはネットワーク マネージャの一部として実行されているスクリプトやアプリケーションです。通常、サーバはネットワーク デバイス（スイッチまたはルータ）です。サーバは、ネットワーク デバイス全体のトランスポート層としてセキュア シェル（SSH）を使用します。SSH ポート番号830をデフォルトのポートとして使用します。ポート番号は、設定可能なオプションです。

NETCONFは、機能の検出およびモデルのダウンロードもサポートしています。サポート対象のモデルは、ietf-netconf-monitoring モデルを使用して検出されます。各モデルに対する改定日付は、機能の応答に示されています。データ モデルは、get-schema RPC を使用して、デバイスからオプションのダウンロードとして入手できます。これらのYANGモデルを使用して、データ モデルを理解したりエクスポートしたりできます。NETCONFの詳細については、RFC 6241を参照してください。

Cisco IOS XE Fuji 16.8.1 よりも前のリリースでは、運用データ マネージャ（ポーリングに基づく）が個別に有効になっていました。Cisco IOS XE Fuji 16.8.1 以降のリリースでは、運用データは、NETCONFを実行しているプラットフォームで動作し（設定データの仕組みと同様）、デフォルトで有効になっています。運用データのクエリまたはストリーミングに対応するコンポーネントの詳細については、GitHub リポジトリで命名規則の *-oper を参照してください。

NETCONF プロトコルの制約事項

- NETCONF 機能は、デュアル IOSd 設定またはソフトウェア冗長性を実行中のデバイスではサポートされていません。

- **no ip pim rp-address** コマンドを使用して NETCONF データストアから RP アドレスを削除すると、パーサーの制限により、データストアに不整合が生じる可能性があります。NETCONF データストアから RP アドレスエントリを削除するには、RPC を使用します。

NETCONF RESTCONF IPv6 のサポート

データ モデル インターフェイス（DMI）は IPv6 プロトコルの使用をサポートしています。DMI による IPv6 のサポートは、クライアント アプリケーションが、IPv6 アドレスを使用するサービスと通信する場合に役に立ちます。外部向けインターフェイスは、IPv4 と IPv6 の両方についてデュアルスタックをサポートします。

DMI は、ネットワーク要素の管理を容易にする一連のサービスです。NETCONF や RESTCONF などのアプリケーション層プロトコルは、ネットワークを介してこれらの DMI にアクセスします。

IPv6 アドレスが設定されていない場合でも、外部向けアプリケーションは IPv6 ソケットをリッスンし続けますが、これらのソケットは到達不能になります。

NETCONF グローバル セッションのロック

NETCONF プロトコルは、デバイス設定を管理し、デバイスの状態情報を取得するための一連の操作を提供します。NETCONF はグローバル ロックをサポートしており、NETCONF では応答しなくなったセッションを kill する機能が導入されています。

複数の同時セッションの全体にわたって一貫性を確保し、設定の競合を防ぐために、セッションのオーナーは NETCONF セッションをロックできます。NETCONF lock RPC は、コンフィギュレーション パーサーと実行コンフィギュレーション データベースをロックします。その他のすべての NETCONF セッション（ロックを所有していない）は、編集操作を実行できません。ただし、読み取り操作は実行できます。これらのロックは存続時間が短いことを意図しており、オーナーは、他の NETCONF クライアント、NETCONF 以外のクライアント（SNMP、CLI スクリプトなど）、および人間のユーザとやり取りをせずに変更を加えることができます。

アクティブ セッションによって保持されているグローバル ロックは、関連付けられたセッションが kill されたときに無効になります。ロックによって、ロックを保持しているセッションが、設定に対して排他的な書き込みアクセスを行えるようになります。グローバル ロックにより設定の変更が拒否された場合は、エラー メッセージによって、NETCONF グローバル ロックが原因で設定の変更が拒否されたことが示されます。

<lock> 操作は必須パラメータ <target> を受け取ります。これは、ロックしようとするコンフィギュレーション データストアの名前です。ロックがアクティブな場合、<edit-config> 操作と <copy-config> 操作は許可されません。

NETCONF のグローバル ロックの保持中に **clear configuration lock** コマンドが指定された場合は、設定の完全な同期がスケジュールされ、警告の syslog メッセージが生成されます。このコマンドは、パーサー コンフィギュレーション ロックのみをクリアします。

次に、<lock> 操作を示す RPC の例を示します。

```
<rpc message-id="101"
          xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
       <lock>
         <target>
           <running/>
         </target>
       </lock>
</rpc>
```

NETCONF Kill セッション

セッションの競合時、またはクライアントによるグローバル ロックの誤用が生じたときは、**show netconf-yang sessions** コマンドを使用して NETCONF セッションをモニタできます。また、**clear netconf-yang session** コマンドを使用して応答しなくなったセッションをクリアすることもできます。**clear netconf-yang session** コマンドは、NETCONF ロックとコンフィギュレーション ロックの両方をクリアします。

<kill-session> 要求は、NETCONF セッションを強制的に終了します。NETCONF エンティティは、オープン セッションの <kill-session> 要求を受信すると、プロセス内のすべての操作を停止し、セッションに関連付けられているすべてのロックとリソースを解放して、関連付けられた接続をすべて閉じます。

<kill-session> 要求には、終了する NETCONF セッションのセッション ID が必要です。セッション ID の値が現在のセッション ID と同じ場合は、無効な値を示すエラーが返されます。NETCONF セッションのトランザクションがまだ進行中に NETCONF セッションが終了した場合は、データ モデル インフラストラクチャによってロールバックが要求され、ネットワーク要素にロールバックが適用されて、すべての YANG モデルの同期がトリガーされます。

セッションの kill が失敗し、グローバル ロックが保持されている場合は、コンソールまたは vty を使用して **clear configuration lock** コマンドを入力します。この時点で、データ モデルを停止して再起動することができます。

NETCONF-YANG SSH サーバのサポート

NETCONF-YANG はパスワードベースの認証に代わる方法として、IOS セキュアシェル（SSH）リベスト、シャミア、エーデマル（RSA）公開キーを使用したユーザの認証をサポートします。

NETCONF-YANG で公開キー認証を機能させるには、IOS SSH サーバを設定する必要があります。SSH サーバに対してユーザを認証するには、**ip ssh pubkey-chain** および **user** コマンドを使用して設定された RSA キーのいずれかを使用します。

NACM は、グループベースのアクセス制御メカニズムです。ユーザが認証されると、設定された権限レベルに基づいて、NACM 権限グループに自動的に配置されます。ユーザを他のユーザ定義グループに手動で配置することもできます。デフォルトの特権レベルは 1 です。PRIV00 ー PRIV15 の 16 の特権レベルがあります。

ユーザが公開キーを介して認証する場合、対応する認証、許可、アカウンティング（AAA）設定がないと、このユーザは拒否されます。ユーザが公開キーを介して認証する場合、NETCONF

の AAA 設定がローカル以外の AAA ソースを使用していると、このユーザも拒否されます。ローカルおよび TACACS + AAA 認証がサポートされます。

トークンベースの RESTCONF 認証はサポートされていません。SSH ユーザ証明書はサポートされていません。

候補コンフィギュレーションのサポート

候補コンフィギュレーション機能を使用すると、シンプルなコミットオプションを使用して RFC 6241 を実装することによって、候補機能をサポートできます。

候補データストアは、デバイスの実行コンフィギュレーションのコピーを保存する一時的な作業領域となります。実行コンフィギュレーションをデバイスにコミットする前に、実行コンフィギュレーションを作成して変更することができます。候補機能は、NETCONF 機能 urn:ietf:params:netconf:capability:candidate:1.0 により示されます。この NETCONF 機能は、デバイスが候補データストアをサポートしていることを示します。

ユーザはこの共有データストアを使用して、デバイスの実行コンフィギュレーションに影響を与えることなく、デバイスのコンフィギュレーションを作成、追加、削除、変更できます。コミット操作では、デバイスのコンフィギュレーションが候補から実行のコンフィギュレーションにプッシュされます。候補データストアが有効になっていると、実行のデータストアには NETCONF セッションを介して書き込むことができず、すべてのコンフィギュレーションは候補を通じてのみコミットされます。つまり、稼働中の設定を直接変更できる NETCONF 機能は、候補コンフィギュレーションでは有効になりません。

（注） 候補データストアは共有データストアであることに留意してください。複数の NETCONF セッションが内容を同時に変更する可能性があります。したがって、内容を変更する前にデータストアをロックして、コミットが競合しないようにし、最終的にコンフィギュレーションの変更が失われる可能性を防ぐことが重要になります。

候補の NETCONF 操作

候補データストアでは次の操作を実行できます。

（注） この項の情報は RFC 6241 の 8.3.4 項を参考にしています。詳細と正確な RPC については、RFC を参照してください。

ロック

<lock> RPC は、ターゲットのデータストアをロックするために使用します。これにより、他のユーザはロックされたデータストアのコンフィギュレーションを変更できなくなります。ロック操作では候補データと実行データの両方をロックできます。

（注） 候補データストアのロックは、Cisco IOS のコンフィギュレーションのロックや実行コンフィギュレーションのロックに影響を与えません。逆も同様です。

コミット

<commit> RPC は、候補コンフィギュレーションをデバイスの実行コンフィギュレーションにコピーします。「コミット」操作は、候補コンフィギュレーションを更新してコンフィギュレーションをデバイスにプッシュした後に実行する必要があります。

実行または候補のデータストアのいずれかが別の NETCONF セッションによってロックされている場合、<commit> RPC は RPC エラー応答で失敗します。<error-tag> は <in-use> となり、<error-info> にはロックを保持している NETCONF セッションのセッション ID が示されます。conf t lock モードに移行してグローバルロックを使用し、「実行」コンフィギュレーションをロックすることもできますが、コミット操作は RPC エラー応答で失敗し、error-tag の値は <in-use>、セッション ID は「0」になります。

コンフィギュレーション編集

候補コンフィギュレーションは、コンフィギュレーションを変更するための edit-config（コンフィギュレーション編集）操作のターゲットとして使用できます。デバイスの実行コンフィギュレーションに影響を与えることなく、候補コンフィギュレーションを変更できます。

廃棄

候補コンフィギュレーションに加えられた変更を削除するには、discard（廃棄）操作を実行して候補コンフィギュレーションを実行コンフィギュレーションに戻します。

たとえば、NETCONF セッション A によって候補データストアの内容が変更されている場合、セッション B が候補データストアをロックしようとするとロックは失敗します。NETCONF セッション B では候補をロックする前に、他の NETCONF セッションから候補データストアの未解決のコンフィギュレーションの変更を削除するために <discard> 操作を実行する必要があります。

ロック解除

ロック、edit-config（コンフィギュレーション編集）、コミットなどで候補コンフィギュレーションを操作した後、ロック解除 RPC でターゲットとして candidate を指定することによって、データストアをロック解除できます。これで、他のセッションでのすべての操作に候補データストアを使用できるようになります。

候補データストアに対する未解決の変更で不具合が発生した場合、コンフィギュレーションの回復が困難になり、他のセッションで問題が生じる可能性があります。問題を回避するため、未解決の変更は、「NETCONF セッションの障害」で暗黙的にロックが解除されたとき、またはロック解除操作を使用して明示的にロックが解除されたときに廃棄される必要があります。

コンフィギュレーション取得、コンフィギュレーションコピー、コンフィギュレーション検証

候補データストアは、get-config（コンフィギュレーション取得）、copy-config（コンフィギュレーション コピー）、または validate（コンフィギュレーション検証）のどの操作でも、ソースまたはターゲットとして使用できます。候補データストアの変更をデバイスにコミットせずに、コンフィギュレーションの検証のみを行う場合は、<validate> RPC の後に discard の操作を付けることで使用できます。

候補データストアの変更

次の図は、候補データストアを介してデバイスコンフィギュレーションを変更する場合に推奨されるベスト プラクティスを示しています。

図 *4*: 候補データストアの変更手順

1. 実行データストアをロックします。
2. 候補データストアをロックします。
3. edit-config RPC とターゲットの候補を使用して、候補コンフィギュレーションを変更します。
4. 候補コンフィギュレーションを、実行コンフィギュレーションにコミットします。
5. 候補データストアと実行データストアをロック解除します。

確認済み候補コンフィギュレーションのコミット

候補コンフィギュレーションは、confirmed-commit 機能をサポートします。この実装では、confirmed-commit 機能に関する RFC 6241 で指定されているとおり、発行されると、実行コンフィギュレーションが候補コンフィギュレーションの現在の内容に設定され、confirmed-commit タイマーが開始されます。commit がタイムアウト期間内に発行されない場合、confirmed-commit 操作はロールバックされます。デフォルトのタイムアウト期間は 600 秒（10分）です。

候補コンフィギュレーションをコミットする場合、コミットを永続的にするための明示的な確認を要求できます。確認済みコミット操作は、コンフィギュレーションの変更が正しく機能し、デバイスへの管理アクセスを妨げないことを確認するのに役立ちます。変更によってアクセスが妨げられたり、その他のエラーが発生したりすると、ロールバックの期限が過ぎた後、以前のコンフィギュレーションへの自動ロールバックによってアクセスが復元されます。指定した時間内にコミットが確認されない場合、デバイスはデフォルトで、以前にコミットされたコンフィギュレーションを自動的に取得してコミット（ロールバック）します。

（注） RESTCONF は確認済みコミットをサポートしていません。

NETCONFセッションでは、候補コンフィギュレーションをコミットし、コミットが永続的になることを明示的に確認するために、クライアントアプリケーションは空の<confirmed/>タグを <commit> および <rpc> タグ要素内に囲みます。

```
<rpc>
  <commit>
    <confirmed/>
  </commit>
</rpc>
```

次に、デフォルトのロールバックタイマーを変更する RPC の例を示します。

```
<rpc>
    <commit>
        <confirmed/>
        <confirm-timeout>nnn</confirm-timeout> !nnn is the rollback-delay in seconds.
    </commit>
</rpc>
```

次のサンプルRPCは、NETCONFサーバが候補コンフィギュレーションが一時的にコミットされたことを確認することを示しています。

```
<rpc-reply xmlns="URN" xmlns:junos="URL">
    <ok/>
</rpc-reply>
```

NETCONFサーバが候補コンフィギュレーションをコミットできない場合、<rpc-reply>要素で失敗の理由を説明する <rpc-error> 要素を囲みます。最も一般的な原因は、候補コンフィギュレーションのセマンティックまたは構文エラーです。

ロールバックを現在のロールバックタイマーよりも後の時間に遅らせるために、クライアントアプリケーションは、期限が切れる前に <commit> タグ要素内にある <confirmed/> タグを再度送信します。オプションで、<confirm-timeout>要素を含めることで、次のロールバックを遅らせる時間を指定します。クライアントアプリケーションは、<confirmed/>タグを繰り返し送信することでロールバックを無制限に遅らせることができます。

コンフィギュレーションを永続的にコミットするには、ロールバック期限が過ぎる前に、クライアントアプリケーションが <rpc> タグ要素で囲まれた <commit/> タグを送信します。ロールバックがキャンセルされ、候補コンフィギュレーションがただちにコミットされます。候補コンフィギュレーションが、一時的にコミットされたコンフィギュレーションと同じ場合、一時的にコミットされたコンフィギュレーションが再コミットされます。

別のアプリケーションが <kill-session/> タグ要素を使用して、確認済みコミットが保留中の間にこのアプリケーションのセッションを終了する場合（このアプリケーションは変更をコミットしましたが、まだ確認していません）、このセッションを使用している NETCONF サーバは、確認済みコミット命令が発行される前の状態にコンフィギュレーションを復元します。

候補データストアは、 **no netconf-yang feature candidate-datastore** コマンドを使用することで無効になります。候補データストアが有効の場合に候補データストアの確認済みコミットが有効になるため、候補データストアが無効の場合は確認済みコミットが無効になります。進行中のすべてのセッションが終了し、confd プログラムが再起動されます。

候補サポートの設定

候補データストア機能は、 **netconf-yang feature candidate-datastore** コマンドを使用して有効にすることができます。データストアの状態が「実行」から「候補」、またはその逆に変わると、変更を有効にするために NETCONF または RESTCONF の再起動が行われることをユーザに通知する警告メッセージが表示されます。

NETCONF-YANG または RESTCONF confd プロセスの開始時に候補または実行のデータストアの選択がコンフィギュレーションで指定されている場合は、次のような警告が表示されます。

```
Device(config)# netconf-yang feature candidate-datastore

netconf-yang initialization in progress - datastore transition not allowed, please try
again after 30 seconds
```

NETCONF-YANG または RESTCONF confd プロセスの開始後に候補または実行の選択が行われた場合は、次のように適用されます。

- **netconf-yang feature candidate-datastore** コマンドが設定されている場合は、コマンドによって候補データストアが有効になり、次の警告が出力されます。

  ```
  “netconf-yang and/or restconf is transitioning from running to candidate netconf-yang
   and/or restconf will now be restarted,
  and any sessions in progress will be terminated”.
  ```

- **netconf-yang feature candidate-datastore** コマンドが削除された場合は、コマンドによって候補データストアが無効になり、実行データストアが有効になり、次の警告が出力されます。

  ```
  netconf-yang and/or restconf is transitioning from candidate to running netconf-yang
   and/or restconf will now be restarted,
  and any sessions in progress will be terminated”.
  ```

- NETCONF-YANG または RESTCONF が再起動すると、進行中のセッションは失われます。

コンフィギュレーション データベースの副次的同期

データ モデル インターフェイス（DMI）の設定変更中に、コマンドまたは RPC の設定時にトリガーされる変更の部分的な同期が行われます。これは副次的同期と呼ばれ、同期時間と NETCONF のダウンタイムを短縮します。副次的同期の前に、コンフィギュレーションデータベースの時間のかかる完全な同期をトリガーするため、設定変更が使用されます。

副次的同期は、**netconf-yang feature side-effect-sync** コマンドによって有効になります。

一部のコマンドは、設定されると、すでに設定されている一部のコマンドの変更をトリガーします。たとえば、次に NETCONF edit-config RPC が設定される前のデバイスの設定を示します。

```
hostname device123
```

NETCONF edit-config RPC：

```
<native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
 <hostname xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" nc:operation="delete"/>
</native>
```

次に、NETCONF edit-config RPC を設定した後のデバイスの設定を示します。

```
hostname Switch
```

ここで、NETCONF edit-config RPC の副作用は、RPC が直接意図していない実行コンフィギュレーションへの変更です。edit-config 要求はホスト名を削除することになっていますが、ホスト名は削除されずに Switch に戻ります。副次的同期では、設定全体を同期することなく、この設定変更を NETCONF データベースと同期するため、パフォーマンスが向上します。

副次的同期はCLIモードツリーの概念に基づいており、コマンドはモードとサブモード構造で維持されます。このCLIモードツリーのデータ構造は、次の3つのメインノードで構成されています。

- 同じレベルのノード：このノードは、同じ親に属し、同じレベルにあるCLIノードのリストを指します。
- 親ノード：このノードは、CLIノードの親、そのモード、およびサブモードノードを指します。
- 子ノード：このノードは子CLI（現在のモードまたはサブモードでのCLI）を指します。ノードに複数の子ノードがある場合、それらの子ノードは同じレベルのノードポインタの一部としてリンクされます。

NETCONF プロトコルの設定方法

NETCONF-YANG は、デバイスのプライマリ トラストポイントを使用します。トラストポイントが存在しない場合に NETCONF-YANG が設定されると、自己署名トラストポイントが作成されます。詳細については、『Public Key Infrastructure Configuration Guide, Cisco IOS XE Gibraltar 16.10.x』を参照してください。

NETCONF を使用するための権限アクセスの提供

NETCONF API の使用を開始するには、権限レベル 15 を持つユーザである必要があります。

手順の概要

1. **enable**
2. **configure terminal**
3. **username** *name* **privilege** *level* **password** *password*
4. **aaa new-model**
5. **aaa authentication login default local**

6. **aaa authorization exec default local**
7. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device# enable`	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**username** *name* **privilege** *level* **password** *password* 例： `Device(config)# username example-name privilege 15 password example_password`	ユーザ名をベースとした認証システムを確立します。次のキーワードを設定します。 • **privilege** *level*：ユーザの権限レベルを設定します。NETCONF プロトコルの場合は、15 にする必要があります。 • **password** *password*：CLI ビューにアクセスするためのパスワードを設定します。
ステップ 4	**aaa new-model** 例： `Device(config)# aaa new-model`	（任意）許可、認証、アカウンティング（AAA）を有効にします。 **aaa new-model** コマンドを設定する場合は、AAA 認証および許可が必要です。
ステップ 5	**aaa authentication login default local** 例： `Device(config)# aaa authentication login default local`	ローカル ユーザ名データベースを使用するログイン認証を設定します。 （注） NETCONF プロトコルでは、デフォルトの AAA 認証ログイン方式のみがサポートされます。 • リモート AAA サーバの場合は、local を AAA サーバに置き換えます。 **default** キーワードにより、ローカルユーザデータベース認証がすべてのポートに適用されます。
ステップ 6	**aaa authorization exec default local** 例： `Device(config)# aaa authorization exec default local`	ユーザの AAA 許可を設定し、ローカルデータベースを確認して、そのユーザに EXEC シェルの実行を許可します。

	コマンドまたはアクション	目的
		• リモート AAA サーバの場合は、local を AAA サーバに置き換えます。 • **default** キーワードにより、ローカルユーザデータベース認証がすべてのポートに適用されます。
ステップ 7	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

NETCONF-YANG の設定

レガシー NETCONF プロトコルがデバイスで有効になっている場合、RFC 準拠の NETCONF プロトコルは機能しません。**no netconf legacy** コマンドを使用してレガシー NETCONF プロトコルを無効にしてください。

手順の概要

1. **enable**
2. **configure terminal**
3. **netconf-yang**
4. **netconf-yang feature candidate-datastore**
5. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**netconf-yang** 例：	ネットワークデバイスでNETCONFインターフェイスを有効にします。

	コマンドまたはアクション	目的
	`Device (config)# netconf-yang`	（注） CLIによる最初のイネーブル化の後、ネットワーク デバイスをモデル ベースのインターフェイスを通じて管理できるようになります。モデル ベースのインターフェイス プロセスの完全なアクティベーションには、最大 90 秒かかることがあります。
ステップ 4	**netconf-yang feature candidate-datastore** 例： `Device(config)# netconf-yang feature candidate-datastore`	候補データストアを有効にします。
ステップ 5	**exit** 例： `Device (config)# exit`	グローバル コンフィギュレーション モードを終了します。

NETCONF オプションの設定

SNMP の設定

NETCONF を有効にして、サポートされている MIB から生成された YANG モデルを使用して SNMP MIB データにアクセスしたり、IOS でサポートされている SNMP トラップを有効にして、サポートされているトラップから NETCONF 通知を受信するには、IOS で SNMP サーバを有効にします。

次の操作を行ってください。

手順の概要

1. IOS で SNMP 機能を有効にします。
2. NETCONF-YANG が起動した後、次の RPC <edit-config> メッセージを NETCONF-YANG ポートに送信して、SNMP トラップのサポートを有効にします。
3. 次の RPC メッセージを NETCONF-YANG ポートに送信して、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。

手順の詳細

ステップ 1 IOS で SNMP 機能を有効にします。

例：

```
configure terminal
logging history debugging
logging snmp-trap emergencies
logging snmp-trap alerts
```

```
logging snmp-trap critical
logging snmp-trap errors
logging snmp-trap warnings
logging snmp-trap notifications
logging snmp-trap informational
logging snmp-trap debugging
!
snmp-server community public RW
snmp-server trap link ietf
snmp-server enable traps snmp authentication linkdown linkup
snmp-server enable traps syslog
snmp-server manager
exit
```

ステップ 2 NETCONF-YANG が起動した後、次の RPC <edit-config> メッセージを NETCONF-YANG ポートに送信して、SNMP トラップのサポートを有効にします。

例：

```
<?xml version="1.0" encoding="utf-8"?>
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <netconf-yang xmlns="http://cisco.com/yang/cisco-self-mgmt">
        <cisco-ia xmlns="http://cisco.com/yang/cisco-ia">
          <snmp-trap-control>
            <trap-list>
              <trap-oid>1.3.6.1.4.1.9.9.41.2.0.1</trap-oid>
            </trap-list>
            <trap-list>
              <trap-oid>1.3.6.1.6.3.1.1.5.3</trap-oid>
            </trap-list>
            <trap-list>
              <trap-oid>1.3.6.1.6.3.1.1.5.4</trap-oid>
            </trap-list>
          </snmp-trap-control>
        </cisco-ia>
      </netconf-yang>
    </config>
  </edit-config>
</rpc>
```

ステップ 3 次の RPC メッセージを NETCONF-YANG ポートに送信して、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。

例：

```
<?xml version="1.0" encoding="utf-8"?>
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="">
  <cisco-ia:save-config xmlns:cisco-ia="http://cisco.com/yang/cisco-ia"/>
</rpc>
```

RSA ベースのユーザ認証を実行するための SSH サーバの設定

NETCONF-YANG がユーザを認証するための SSH 公開キーを設定するには、次の作業を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip ssh pubkey-chain**
4. **username** *username*
5. **key-string**
6. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**ip ssh pubkey-chain** 例： `Device(config)# ip ssh pubkey-chain`	SSH サーバ上のユーザおよびサーバ認証用に SSH-RSA キーを設定し、公開キー コンフィギュレーション モードを開始します。 • サーバに保存されている RSA 公開キーが、クライアントに保存されている公開キーと秘密キーのペアを使用して検証されると、ユーザ認証は成功です。
ステップ 4	**username** *username* 例： `Device(conf-ssh-pubkey)# username user1`	SSH ユーザ名を設定し、公開キー ユーザ コンフィギュレーション モードを開始します。
ステップ 5	**key-string** 例： `Device(conf-ssh-pubkey-user)# key-string`	リモート ピアの RSA 公開キーを指定し、公開キー データ コンフィギュレーション モードを開始します。 （注） オープン SSH クライアントから（言い換えると .ssh/id_rsa.pub ファイルから）公開キー値を取得できます。
ステップ 6	**end** 例： `Device(conf-ssh-pubkey-data)# end`	公開キー データ コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。 • デフォルト ホストに戻るには、**no hostname** コマンドを使用します。

CLI を使用した NETCONF プロトコルのコンフィギュレーションの確認

NETCONF コンフィギュレーションを確認するには次のコマンドを使用します。

手順の概要

1. **show netconf-yang datastores**
2. **show netconf-yang sessions**
3. **show netconf-yang sessions detail**
4. **show netconf-yang diagnostics summary**
5. **show netconf-yang statistics**
6. **show platform software yang-management process**

手順の詳細

ステップ 1 show netconf-yang datastores

NETCONF-YANG データストアに関する情報を表示します。

例：

```
Device# show netconf-yang datastores

Device# show netconf-yang datastores
Datastore Name : running
Globally Locked By Session : 42
Globally Locked Time : 2018-01-15T14:25:14-05:00
```

ステップ 2 show netconf-yang sessions

NETCONF-YANG セッションに関する情報を表示します。

例：

```
Device# show netconf-yang sessions

R: Global-lock on running datastore
C: Global-lock on candidate datastore
S: Global-lock on startup datastore
Number of sessions : 10
session-id transport username source-host global-lock
------------------------------------------------------------------------------
40 netconf-ssh admin 10.85.70.224 None
42 netconf-ssh admin 10.85.70.224 None
44 netconf-ssh admin 10.85.70.224 None
46 netconf-ssh admin 10.85.70.224 None
48 netconf-ssh admin 10.85.70.224 None
50 netconf-ssh admin 10.85.70.224 None
52 netconf-ssh admin 10.85.70.224 None
54 netconf-ssh admin 10.85.70.224 None
56 netconf-ssh admin 10.85.70.224 None
```

```
58 netconf-ssh admin 10.85.70.224 None
```

ステップ 3 show netconf-yang sessions detail

NETCONF-YANG セッションに関する詳細情報を表示します。

例：

```
Device# show netconf-yang sessions detail

R: Global-lock on running datastore
C: Global-lock on candidate datastore
S: Global-lock on startup datastore

Number of sessions     : 1

session-id             : 19
transport              : netconf-ssh
username               : admin
source-host            : 2001:db8::1
login-time             : 2018-10-26T12:37:22+00:00
in-rpcs                : 0
in-bad-rpcs            : 0
out-rpc-errors         : 0
out-notifications      : 0
global-lock            : None
```

ステップ 4 show netconf-yang diagnostics summary

NETCONF-YANG 診断情報の概要を表示します。

例：

```
Device# show netconf-yang diagnostics summary

Diagnostic Debugging is ON
Diagnostic Debugging Level: Maximum
Total Log Size (bytes): 20097
Total Transactions: 1
message username session-id transaction-id start-time        end-time          log size
-------------------------------------------------------------------------------------------
1       admin    35          53             03/12/21 14:31:03 03/12/21 14:31:04 20097
```

ステップ 5 show netconf-yang statistics

NETCONF-YANG 統計に関する情報を表示します。

例：

```
Device# show netconf-yang statistics

netconf-start-time : 2018-01-15T12:51:14-05:00
in-rpcs : 0
in-bad-rpcs : 0
out-rpc-errors : 0
out-notifications : 0
in-sessions : 10
dropped-sessions : 0
in-bad-hellos : 0
```

ステップ 6　show platform software yang-management process

NETCONF-YANG のサポートに必要なソフトウェア プロセスのステータスを表示します。

例：

```
Device# show platform software yang-management process

confd            : Running
nesd             : Running
syncfd           : Running
ncsshd           : Running
dmiauthd         : Running
vtyserverutild   : Running
opdatamgrd       : Running
nginx            : Running
ndbmand          : Running
```

（注）　プロセス nginx は、**ip http secure-server** または **ip http server** がデバイスで設定されている場合に実行されます。このプロセスが「実行」状態でなくても NETCONF は正常に機能します。ただし、RESTCONF には nginx プロセスが必要です。

表 *12 : show platform software yang-management process* のフィールドの説明

フィールド	説明
confd	コンフィギュレーション デーモン
nesd	ネットワーク要素シンクロナイザ デーモン
syncfd	デーモンからの同期
ncsshd	NETCONF セキュア シェル（SSH）デーモン
dmiauthd	デバイス管理インターフェイス（DMI）認証デーモン
vtyserverutild	VTY サーバ ユーティリティ デーモン
opdatamgrd	運用データ マネージャ デーモン
nginx	NGINX Web サーバ
ndbmand	NETCONF データベース マネージャ

RPC による NETCONF-YANG 診断の表示

show netconf-yang diagnostics コマンドまたは次の RPC を使用して、診断情報を表示できます。

次に、NETCONF-YANG 診断を有効にする RPC の例と、ホストから受信した RPC 応答を示します。

```
#308
<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
    message-id="urn:uuid:b0f45ac0-3fe2-4e1d-a3a1-f57985965be6">
  <enable-netconf-diag xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-netconf-diag-rpc">
    <diag-level>diag-maximum</diag-level>
  </enable-netconf-diag>
</nc:rpc>

##

Received message from host
<?xml version="1.0" ?>
<rpc-reply message-id="urn:uuid:b0f45ac0-3fe2-4e1d-a3a1-f57985965be6"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
    xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

次に、現在のステータスを示す RPC の例と、ホストから受信した RPC 応答を示します。

```
#294
<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
    message-id="urn:uuid:c6c986ac-fc44-45e2-9390-f8a5968dc8d4">
  <nc:get>
    <nc:filter>
      <netconf-diag-oper-data
xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-netconf-diag-oper"/>
    </nc:filter>
  </nc:get>
</nc:rpc>

#

Received message from host
<?xml version="1.0" ?>
<rpc-reply message-id="urn:uuid:c6c986ac-fc44-45e2-9390-f8a5968dc8d4"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
    xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <netconf-diag-oper-data
xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-netconf-diag-oper">
      <diag-summary>
        <level>diag-maximum</level>
        <log-size>0</log-size>
        <trans-count>0</trans-count>
      </diag-summary>
    </netconf-diag-oper-data>
  </data>
</rpc-reply>
```

次に、ホスト名を変更するための RPC の例と、ホストから受信した RPC 応答を示します。

```
#
#364
<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
    message-id="urn:uuid:f3005ee6-8a11-4146-b616-dd95a92b97d1">
  <nc:edit-config>
    <nc:target>
      <nc:running/>
    </nc:target>
    <nc:config>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <hostname>new-ott-c9300-35</hostname>
      </native>
    </nc:config>
  </nc:edit-config>
</nc:rpc>

##

Received message from host
<?xml version="1.0" ?>
<rpc-reply message-id="urn:uuid:f3005ee6-8a11-4146-b616-dd95a92b97d1"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
    xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

次に、現在のステータスを表示するための RPC の例と、ホストから受信した RPC 応答を示します。

```
 #294
<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
    message-id="urn:uuid:9bffb8d5-3866-48ef-b59d-0486e508fbc4">
  <nc:get>
    <nc:filter>
      <netconf-diag-oper-data
xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-netconf-diag-oper"/>
    </nc:filter>
  </nc:get>
</nc:rpc>

##

Received message from host
<?xml version="1.0" ?>
<rpc-reply message-id="urn:uuid:9bffb8d5-3866-48ef-b59d-0486e508fbc4"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
    xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <netconf-diag-oper-data
xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-netconf-diag-oper">
      <diag-summary>
        <level>diag-maximum</level>
        <log-size>20775</log-size>
        <trans-count>1</trans-count>
      </diag-summary>
      <diag-trans>
        <message>1</message>
        <username>lab</username>
        <session-id>31</session-id>
        <trans-id>50</trans-id>
        <start-time>2021-03-12T14:08:26.830334+00:00</start-time>
```

```
        <end-time>2021-03-12T14:08:28.279414+00:00</end-time>
        <log-size>20775</log-size>
      </diag-trans>
    </netconf-diag-oper-data>
  </data>
</rpc-reply>
```

次に、収集されたシステムエラーメッセージをアーカイブするための PRC の例と、ホストから受信した RPC 応答を示します。

```
#
#256
<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
    message-id="urn:uuid:1dbc795c-f594-4194-a89b-fd4d88446b69">
  <archive-netconf-diag-logs
xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-netconf-diag-rpc"/>
</nc:rpc>

##

Received message from host
<?xml version="1.0" ?>
<rpc-reply message-id="urn:uuid:1dbc795c-f594-4194-a89b-fd4d88446b69"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
    xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <log-file xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-netconf-diag-rpc">
    bootflash:netconf-yang-diag.20210312141009.tar.gz</log-file>


</rpc-reply>
```

次に、NETCONF-YANG 診断を無効にする RPC の例と、ホストから受信した RPC 応答を示します。

```
#309
<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
    message-id="urn:uuid:d253a313-4aec-42bc-80a2-672e9bb9ad56">
  <enable-netconf-diag xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-netconf-diag-rpc">
    <diag-level>diag-disabled</diag-level>
  </enable-netconf-diag>
</nc:rpc>

##

Received message from host
<?xml version="1.0" ?>
<rpc-reply message-id="urn:uuid:d253a313-4aec-42bc-80a2-672e9bb9ad56"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
    xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

NETCONF プロトコルの関連資料

関連資料

関連項目	マニュアル タイトル
IOS-XE、IOS-XR、および NX-OS プラットフォームのさまざまなリリースの YANG データ モデル	開発者に分かりやすい方法で Cisco YANG モデルにアクセスするには、*GitHub リポジトリ*を複製し、*vendor/cisco* サブディレクトリに移動します。ここでは、IOS XE、IOS-XR、および NX-OS プラットフォームのさまざまなリリースのモデルを使用できます。

標準および RFC

標準/RFC	タイトル
RFC 6020	*YANG：Network Configuration Protocol（NETCONF）向けデータ モデリング言語*
RFC 6241	ネットワーク設定プロトコル（*NETCONF*）
RFC 6536	ネットワーク設定プロトコル（*NETCONF*）アクセス制御モデル
RFC 8040	*RESTCONF プロトコル*

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

NETCONF プロトコルの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *13 : NETCONF* プロトコルの機能情報

機能名	リリース	機能情報
NETCONF プロトコル	Cisco IOS XE Denali 16.3.1	NETCONFプロトコル機能によって、プログラムによる各種の標準規格に準拠した方法で、設定の記述やネットワーク デバイスからの運用データの読み取りが容易になります。 次のコマンドが導入されました：**netconf-yang** この機能は、次のプラットフォームに実装されていました。 • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco Cloud Services Router 1000V シリーズ
	Cisco IOS XE Everest 16.5.1a	この機能は、次のプラットフォームに実装されていました。 • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ
	Cisco IOS XE Everest 16.6.2	この機能は、次のプラットフォームに実装されていました。 • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ
	Cisco IOS XE Fuji 16.8.1a	

機能名	リリース	機能情報
		Cisco IOS XE Fuji 16.8.1a では、この機能は次のプラットフォームに実装されていました。 • Cisco 1000 シリーズ サービス統合型ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ • Cisco CBR-8 シリーズ ルータ • Cisco Network Convergence System 4200 シリーズ
	Cisco IOS XE Fuji 16.9.2	この機能は、次のプラットフォームに実装されていました。 • Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300L SKU
	Cisco IOS XE Gibraltar 16.10.1	Cisco IOS XE Gibraltar 16.10.1 では、この機能は次のプラットフォームに実装されていました。 • Cisco Catalyst 9800-40 ワイヤレスコントローラ • Cisco Catalyst 9800-80 ワイヤレスコントローラ • Cisco Network Convergence System 520 シリーズ
	Cisco IOS XE Gibraltar 16.11.1	Cisco IOS XE Gibraltar 16.11.1 では、この機能は Cisco Catalyst 9600 シリーズ スイッチに実装されていました。
	Cisco IOS XE Gibraltar 16.12.1	この機能は、Cisco IOS XE Gibraltar 16.12.1 で、Cisco Catalyst 9800-L ワイヤレスコントローラに実装されました。

機能名	リリース	機能情報
	Cisco IOS XE Amsterdam 17.3.1	この機能は、Cisco IOS XE Amsterdam 17.3.1 で次のプラットフォームに実装されました。 • Cisco Catalyst 8200 シリーズ エッジ プラットフォーム • Cisco Catalyst 8300 シリーズ エッジ プラットフォーム • Cisco Catalyst 8500 および 8500L シリーズ エッジ プラットフォーム

機能名	リリース	機能情報
NETCONF および RESTCONF IPv6 のサポート	Cisco IOS XE Fuji 16.8.1a	NETCONF および RESTCONF プロトコルの IPv6 のサポート。この機能は、次のプラットフォームに実装されていました。 • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco CBR-8 シリーズ ルータ • Cisco Cloud Services Router 1000V シリーズ
	Cisco IOS XE Gibraltar 16.11.1	Cisco IOS XE Gibraltar 16.11.1 では、この機能は Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチに実装されていました。

機能名	リリース	機能情報
NETCONF グローバル ロックおよびセッションの kill	Cisco IOS XE Fuji 16.8.1a	NETCONF プロトコルは、グローバル ロックおよび応答しなくなったセッションを kill する機能をサポートしています。この機能は、次のプラットフォームに実装されています。 • Cisco 1100 シリーズ サービス統合型ルータ • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco CBR-8 シリーズ ルータ • Cisco Cloud Services Router 1000V シリーズ

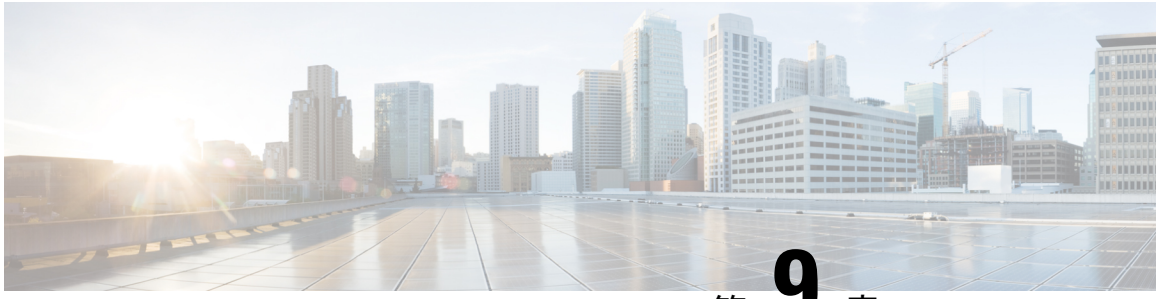
機能名	リリース	機能情報
NETCONF：候補コンフィギュレーション サポート	Cisco IOS XE Fuji 16.9.1	候補コンフィギュレーション サポート機能を使用すると、シンプルなコミットオプションを使用して RFC 6241 を実装することによって、候補機能をサポートできます。 この機能は、次のプラットフォームに実装されていました。 • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco CBR-8 シリーズ ルータ • Cisco Cloud Services Router 1000V シリーズ 次のコマンドが導入されました：**netconf-yang feature candidate-datastore**

機能名	リリース	機能情報
NETCONF：候補コンフィギュレーションのコミットの確認	Cisco IOS XE Amsterdam 17.1.1	候補コンフィギュレーションは、confirmed-commit 機能をサポートします。 この機能は、次のプラットフォームに実装されていました。 • Cisco 1000 シリーズ サービス統合型ルータ • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco ASR 900 シリーズアグリゲーション サービス ルータ • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco cBR-8 コンバージドブロードバンド ルータ • Cisco Cloud Services Router 1000V シリーズ • Cisco Network Convergence System 520 シリーズ • Cisco Network Convergence System 4200 シリーズ

機能名	リリース	機能情報
NETCONF-YANG SSH サーバのサポート	Cisco IOS XE Gibraltar 16.12.1	

機能名	リリース	機能情報
		この機能は、次のプラットフォームに実装されていました。 • Cisco 1000 シリーズ サービス統合型ルータ • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 900 シリーズアグリゲーション サービス ルータ • Cisco ASR 920 シリーズアグリゲーション サービス ルータ • Cisco ASR 1000 アグリゲーション サービス ルータ • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ • Cisco Catalyst 9800 シリーズ ワイヤレス コントローラ • Cisco cBR-8 コンバージドブロードバンド ルータ • Cisco Cloud Services Router 1000V シリーズ • Cisco Network Convergence System 520 シリーズ

機能名	リリース	機能情報
		• Cisco Network Convergence System 4200 シリーズ
コンフィギュレーションデータベースの副次的同期	Cisco IOS XE Bengaluru 17.4.1	DMI の設定変更中に、コマンドまたは RPC の設定時にトリガーされる変更の部分的な同期が行われます。これは副次的同期と呼ばれ、同期時間と NETCONF のダウンタイムを短縮します。 この機能は、次のプラットフォームに実装されていました。 • Cisco ASR 1000 アグリゲーションサービス ルータ • Cisco Catalyst 8500 および 8500L シリーズ エッジ プラットフォーム • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ



第 9 章

RESTCONF プロトコル

この章では、HTTP ベースの Representational State Transfer コンフィギュレーション プロトコル（RESTCONF）を設定する方法を説明します。RESTCONF は、設定データ、状態データ、データ モデルに固有のリモート プロシージャ コール（RPC）操作、および YANG モデルで定義されているイベントにアクセスするための、標準的なメカニズムに基づく、プログラミングが可能なインターフェイスを提供します。

RESTCONF プロトコルの前提条件

- RESTCONF に対して Cisco IOS-HTTP サービスを有効にします。詳細については、『RESTCONF RPC の例』を参照してください。

RESTCONF プロトコルの制約事項

RESTCONF プロトコルには、次の制約事項が適用されます。

- 通知およびイベント ストリーム
- YANG パッチ
- フィルタ、開始時、停止時、再生、アクションなどのオプションのクエリ パラメータ
- RESTCONF 機能は、デュアル IOSd 設定またはソフトウェア冗長性を実行しているデバイスではサポートされていません。

RESTCONF プロトコルに関する情報

RESTCONF の概要

このセクションでは、構成をネットワーク デバイスにプログラムを使用して書き込めるようにする、プロトコルおよびモデリング言語について説明します。

- RESTCONF：構造化データ（XML または JSON）および YANG を使用して REST ライクな API を提供します。これによりさまざまなネットワーク デバイスにプログラムを使用してアクセスできます。RESTCONF API は HTTPs メソッドを使用します。
- YANG：モデル構成および操作機能に使用されるデータ モデリング言語。YANG は、NETCONF および RESTCONF API によって実行できる関数の有効範囲と種類を決定します。

Cisco IOS XE Fuji 16.8.1 よりも前のリリースでは、運用データ マネージャ（ポーリングに基づく）が個別に有効になっていました。Cisco IOS XE Fuji 16.8.1 以降のリリースでは、運用データは、NETCONF を実行しているプラットフォームで動作し（設定データの仕組みと同様）、デフォルトで有効になっています。運用データのクエリまたはストリーミングに対応するコンポーネントの詳細については、GitHub リポジトリで命名規則の *-oper* を参照してください。

HTTPs メソッド

ステートレス プロトコルである HTTPS ベースの RESTCONF プロトコル（RFC 8040）は、セキュアなHTTP メソッドを使用して、YANG 定義データが含まれる概念データストア（NETCONF データストアを実装するサーバと互換性がある）で CREATE、READ、UPDATE、および DELETE（CRUD）操作を提供します。

次の表では、RESTCONF 操作に NETCONF プロトコル操作を関連付ける方法を示しています。

オプション	サポートされているメソッド
GET	読み取り
PATCH	更新
PUT	作成または置換
POST	作成または操作（リロード、デフォルト）
DELETE	ターゲット リソースの削除
HEAD	ヘッダー メタデータ（応答本文なし）

RESTCONF ルート リソース

- RESTCONF デバイスは、RESTCONF 属性を含むリンク要素である /.well-known/host-meta リソースにより、RESTCONF API のルートを決定します。
- RESTCONF デバイスは、要求 URI のパスの最初の部分として RESTCONF API ルート リソースを使用します。

例：

```
Example returning /restconf:

   The client might send the following:

      GET /.well-known/host-meta HTTP/1.1
      Host: example.com
      Accept: application/xrd+xml

   The server might respond as follows:

      HTTP/1.1 200 OK
      Content-Type: application/xrd+xml
      Content-Length: nnn

      <XRD xmlns='http://docs.oasis-open.org/ns/xri/xrd-1.0'>
          <Link rel='restconf' href='/restconf'/>
      </XRD>
```

URI の例：

- GigabitEthernet0/0/2：
  https://10.104.50.97/restconf/data/Cisco-IOS-XE-native:native/interface/GigabitEthernet=0%2F0%2F2
- fields=name：
  https://10.104.50.97/restconf/data/Cisco-IOS-XE-native:native/interface/GigabitEthernet=0%2F0%2F2?fields=name
- depth=1：
  https://10.85.116.59/restconf/data/Cisco-IOS-XE-native:native/interface/GigabitEthernet?depth=1
- Name と IP：
  https://10.85.116.59/restconf/data/Cisco-IOS-XE-native:native/interface?fields=GigabitEthernet/ip/address/primary;name
- MTU（フィールド）：
  https://10.104.50.97/restconf/data/Cisco-IOS-XE-native:native/interface?fields=GigabitEthernet(mtu)
- MTU：
  https://10.85.116.59/restconf/data/Cisco-IOS-XE-native:native/interface/GigabitEthernet=3/mtu
- ポートチャネル：
  https://10.85.116.59/restconf/data/Cisco-IOS-XE-native:native/interface/Port-channel
- 「Char」から「Hex」への変換チャート：http://www.columbia.edu/kermit/ascii.html

バージョン情報の表示

Cisco-IOS-XE-install-operモジュールには、バージョン情報を表示するさまざまなノードがあります。

次のサンプル RPC は、Cisco-IOS-XE-install-oper モジュールのサポートされているノードの一部と、メジャーおよびマイナーリリースバージョンを含むホストからの応答を示しています。

```
<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
        message-id="urn:uuid:7d0908d8-0d5f-4521-9d7b-380b81304776">
  <nc:get>
    <nc:filter>
      <install-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-install-oper">
        <install-location-information>
          <install-version-info>
            <version/>
            <version-extension/>
            <current/>
            <src-filename/>
          </install-version-info>
        </install-location-information>
      </install-oper-data>
    </nc:filter>
  </nc:get>
</nc:rpc>

##
Received message from host

<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
    xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
    message-id="urn:uuid:7d0908d8-0d5f-4521-9d7b-380b81304776">
  <data>
    <install-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-install-oper">
      <install-location-information>
        <install-version-info>
          <version>17.06.04.0.3870</version>
          <version-extension>1651661105</version-extension>
          <current>install-version-state-present</current>
          <src-filename/>
        </install-version-info>
        <install-version-info>
          <version>17.09.01.0.158212</version>
          <version-extension>1651125381</version-extension>
          <current>install-version-state-present</current>
          <src-filename/>
        </install-version-info>
        <install-version-info>
          <version>17.10.01.0.158658</version>
          <version-extension>1651754624</version-extension>
          <current>install-version-state-present</current>

<src-filename>/bootflash/c8000v-universalk9nic.2022-05-05_18.13.SSA.bin</src-filename>
        </install-version-info>
        <install-version-info>
          <version>17.10.01.0.160585</version>
          <version-extension>1656581638</version-extension>
          <current>install-version-state-provisioned-committed</current>

<src-filename>/bootflash/c8000v-universalk9.2022-06-30_15.03.SSA.bin</src-filename>
        </install-version-info>
```

```
        <install-version-info>
          <version>17.10.01.0.162616</version>
          <version-extension>1657120419</version-extension>
          <current>install-version-state-present</current>
          <src-filename>/bootflash/c8000v-universalk9.BLD_POLARIS_DEV_LATEST_20220706_
            143733.SSA.bin</src-filename>
        </install-version-info>
      </install-location-information>
    </install-oper-data>
  </data>
</rpc-reply>
```

プロトコル、gNMI、NETCONF、またはRESTCONFを使用する場合、Cisco-IOS-XE-native:version モジュールは、メジャーリリースバージョンのみを表示します。

RESTCONF API リソース

APIリソースは、+restconfに位置する上位リソースです。これは次のメディアタイプをサポートします。

（注） メディアは、RESTCONF サーバ（XML または JSON）に送信される YANG 形式 RPC のタイプです。

- application/yang-data+xml または application/yang-data+json
- API リソースには、RESTCONF DATASTORE および OPERATION リソースの RESTCONF ルート リソースが含まれます。次に例を示します。

```
The client may then retrieve the top-level API resource, using the
   root resource "/restconf".

      GET /restconf HTTP/1.1
      Host: example.com
      Accept: application/yang-data+json

   The server might respond as follows:

      HTTP/1.1 200 OK
      Date: Thu, 26 Jan 2017 20:56:30 GMT
      Server: example-server
      Content-Type: application/yang-data+json

      {
        "ietf-restconf:restconf" : {
          "data" : {},
          "operations" : {},
          "yang-library-version" : "2016-06-21"
        }
      }
```

詳細については、RFC 3986 を参照してください

メソッド

メソッドは、ターゲット リソースで実行される HTTPS 操作（GET/PATCH/POST/DELETE/OPTIONS/PUT）です。YANG 形式 RPC は、RESTCONF サーバに存在するターゲット YANG モデルに関連する指定のリソースに対して、特定のメソッドを呼び出します。Uniform Resource Identifier（URI）は指定されたリソースのロケーション ID として機能するため、クライアントの RESTCONF メソッドは、その特定のリソースを探して、HTTPS のメソッドまたはプロパティで指定されたアクションを実行することができます。

詳細については、「RFC 8040：RESTCONF プロトコル」を参照してください。

RESTCONF YANG パッチのサポート

RESTCONF は、RFC 8072 で指定されている YANG パッチメディアタイプをサポートしています。YANG パッチは、RESTCONF サーバによってターゲットデータストアに適用される編集の順序付きリストです。YANG パッチ操作は、メディアタイプ *application/yang-patch+xml* または *application/yang-patch+json* のいずれかを使用した表現でパッチメソッド要求を送信することによって RESTCONF クライアントにより呼び出されます。

YANG パッチは一意のパッチ ID で識別されます。パッチは編集の順序付けられたコレクションであり、各編集は編集 ID によって識別されます。ターゲットリソースに適用される編集操作（「作成」、「削除（delete）」、「挿入」、「マージ」、「移動」、「置換」、「削除（remove）」）があります。

RESTCONF YANG パッチがサポートされているかどうかを確認するには、次の RESTCONF Get 要求を発行します。

```
$ curl -k -s -u admin:DMIdmi1! --location-trusted
"https://10.1.1.1/restconf/data/ietf-restconf-monitoring:restconf-state/capabilities"
-X GET

<capabilities xmlns="urn:ietf:params:xml:ns:yang:ietf-restconf-monitoring"
xmlns:rcmon="urn:ietf:params:xml:ns:yang:ietf-restconf-monitoring">

<capability>urn:ietf:params:restconf:capability:defaults:1.0?basic-mode=explicit</capability>

  <capability>urn:ietf:params:restconf:capability:depth:1.0</capability>
  <capability>urn:ietf:params:restconf:capability:fields:1.0</capability>
  <capability>urn:ietf:params:restconf:capability:with-defaults:1.0</capability>
  <capability>urn:ietf:params:restconf:capability:filter:1.0</capability>
  <capability>urn:ietf:params:restconf:capability:replay:1.0</capability>

<capability>urn:ietf:params:restconf:capability:yang-patch:1.0</capability>

  <capability>http://tail-f.com/ns/restconf/collection/1.0</capability>
  <capability>http://tail-f.com/ns/restconf/query-api/1.0</capability>
</capabilities>
```

このセクションでは、いくつかの RESTCONF YANG パッチの例を示します。

リソースの追加エラー

ファイルを編集しようとしているときに、最初の編集がすでに存在し、エラーが報告されます。最初の編集が失敗したため、残りの編集は試行されません。この例では、XML エンコーディングが使用されています。

次の例は、RESTCONF クライアントからのリソース追加要求を示しています。

```
<yang-patch xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-patch">
        <patch-id>add-hostname-patch</patch-id>
        <edit>
          <edit-id>edit1</edit-id>
          <operation>create</operation>
          <target>/hostname</target>
          <value>
              <hostname
xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">Cat9K-test</hostname>
          </value>
        </edit>
        <edit>
          <edit-id>edit2</edit-id>
          <operation>create</operation>
          <target>/interface/Loopback=1</target>
          <value>
              <interface xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
              <Loopback>
                <name>1</name>
              </Loopback>
              </interface>
          </value>
        </edit>
</yang-patch>
```

次の例は、RESTCONF サーバからの JSON 応答を示しています。

```
Device:/nobackup/folder1/confd_6313/bin $ curl -k -s -u admin:DMIdmi1! --location-trusted

"https://10.1.1.1/restconf/data/Cisco-IOS-XE-native:native" -X PATCH -H "Accept:
application/yang-data+json" -d
'@yang_patch_create_hostname' -H "Content-type: application/yang-patch+xml"
{
  "ietf-yang-patch:yang-patch-status": {
    "patch-id": "add-hostname-patch",
    "edit-status": {
      "edit": [
        {
          "edit-id": "edit1",
          "errors": {
            "error": [
              {
                "error-type": "application",
                "error-tag": "data-exists",
                "error-path": "/Cisco-IOS-XE-native:native/hostname",
                "error-message": "object already exists: /ios:native/ios:hostname"
              }
            ]
          }
        }
      ]
    }
  }
}
```

次の例は、RESTCONF サーバからの XML 応答を示しています。

```
Device:/nobackup/folder1/confd_6313/bin $ curl -k -s -u admin:DMIdmi1! --location-trusted

"https://10.1.1.1/restconf/data/Cisco-IOS-XE-native:native" -X PATCH -H "Accept:
application/yang-data+xml" -d
'@yang_patch_create_hostname' -H "Content-type: application/yang-patch+xml"

 <yang-patch-status xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-patch">
   <patch-id>add-hostname-patch</patch-id>
   <edit-status>
     <edit>
       <edit-id>edit1</edit-id>
       <errors>
         <error>
           <error-type>application</error-type>
           <error-tag>data-exists</error-tag>
           <error-path
xmlns:ios="http://cisco.com/ns/yang/Cisco-IOS-XE-native">/ios:native/ios:hostname</error-path>

          <error-message>object already exists: /ios:native/ios:hostname</error-message>

         </error>
       </errors>
     </edit>
   </edit-status>
 </yang-patch-status>device:/nobackup/folder1/confd_6313/bin $
```

リソースの追加成功

次の例は、編集要求を示しています。

```
<yang-patch xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-patch">
        <patch-id>add-Loopback-patch</patch-id>
        <edit>
          <edit-id>edit1</edit-id>
          <operation>create</operation>
          <target>/Loopback=1</target>
          <value>
              <Loopback xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
                <name>1</name>
              </Loopback
          </value>
        </edit>
</yang-patch>
```

次の例は、編集要求が成功したことを示しています。

```
Device:/nobackup/folder1/confd_6313/bin $ curl -k -s -u admin:DMIdmi1! --location-trusted

"https://10.1.1.1/restconf/data/Cisco-IOS-XE-native:native/interface" -X PATCH -H "Accept:
 application/yang-data+json"
-d '@yang_patch_create_Loopback_interface' -H "Content-type: application/yang-patch+xml"
Device:/nobackup/folder1/confd_6313/bin
{
  "ietf-yang-patch:yang-patch-status": {
    "patch-id": "add-Loopback-patch",
    "ok" : [null]
  }
}
```

リストエントリの挿入

次に、ループバック 1 がループバック 0 の後に挿入される例を示します。

```
<yang-patch xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-patch">
        <patch-id>insert-Loopback-patch</patch-id>
        <edit>
          <edit-id>edit1</edit-id>
          <operation>insert</operation>
          <target>/Loopback=1</target>
          <point>/Loopback=0</point>
          <where>after</where>
          <value>
              <Loopback xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
                <name>1</name>
              </Loopback>
          </value>
        </edit>
</yang-patch>
```

次の例は、リストの挿入要求が成功したことを示しています。

```
Device:/nobackup/folder1/confd_6313/bin $ curl -k -s -u admin:DMIdmi1! --location-trusted

"https://10.1.1.1/restconf/data/Cisco-IOS-XE-native:native/interface" -X PATCH -H "Accept:
 application/yang-data+json" -d
'@yang_patch_create_Loopback_interface' -H "Content-type: application/yang-patch+xml"
Device:/nobackup/folder1/confd_6313/bin
{
  "ietf-yang-patch:yang-patch-status": {
    "patch-id": "insert-Loopback-patch",
    "ok" : [null]
  }
}
```

リストエントリの移動

次に、ループバック 1 がループバック 0 の前に移動される例を示します。

```
<yang-patch xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-patch">
        <patch-id>move-Loopback-patch</patch-id>
        <edit>
          <edit-id>edit1</edit-id>
          <operation>move</operation>
          <target>/Loopback=1</target>
          <point>/Loopback=0</point>
          <where>before</where>
          <value>
              <Loopback xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
                <name>1</name>
              </Loopback>
          </value>
        </edit>
</yang-patch>
```

次の例は、移動要求が成功したことを示しています。

```
Device:/nobackup/folder1/confd_6313/bin $ curl -k -s -u admin:DMIdmi1! --location-trusted

"https://10.1.1.1/restconf/data/Cisco-IOS-XE-native:native/interface" -X PATCH -H "Accept:
```

```
 application/yang-data+json" -d
'@yang_patch_create_Loopback_interface' -H "Content-type: application/yang-patch+xml"
Device:/nobackup/folder1/confd_6313/bin
{
  "ietf-yang-patch:yang-patch-status": {
    "patch-id": "move-Loopback-patch",
    "ok" : [null]
  }
}
```

RESTCONF プロトコルの設定方法

AAA を使用した NETCONF/RESTCONF の認証

始める前に

NETCONF 接続と RESTCONF 接続は、認証、許可、およびアカウンティング（AAA）を使用して認証する必要があります。その結果、権限レベル 15 のアクセスで定義された RADIUS または TACACS + ユーザに、システムへのアクセスが許可されます。

手順の概要

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa group server radius** *server-name*
5. **server-private** *ip-address* **key** *key-name*
6. **ip vrf forwarding** *vrf-name*
7. **exit**
8. **aaa authentication login default group** *group-name***local**
9. **aaa authentication login** *list-name* **none**
10. **aaa authorization exec default group** *group-name***local**
11. **aaa session-id common**
12. **line console** *number*
13. **login authentication** *authentication-list*
14. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**aaa new-model** 例： `Device(config)# aaa new-model`	AAA をイネーブルにします。
ステップ 4	**aaa group server radius** *server-name* 例： `Device(config)# aaa group server radius ISE`	RADIUS サーバを追加し、サーバ グループ RADIUS コンフィギュレーション モードを開始します。 • server-name 引数には、RADIUS サーバ グループ名を指定します。
ステップ 5	**server-private** *ip-address* **key** *key-name* 例： `Device(config-sg-radius)# server-private 172.25.73.76 key Cisco123`	プライベート RADIUS サーバの IP アドレスと暗号キーを設定します。
ステップ 6	**ip vrf forwarding** *vrf-name* 例： `Device(config-sg-radius)# ip vrf forwarding Mgmt-intf`	AAA RADIUS または TACACS+ サーバ グループの Virtual Route Forwarding（VRF）参照情報を設定します。
ステップ 7	**exit** 例： `Device(config-sg-radius)# exit`	サーバ グループ RADIUS コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードに戻ります。
ステップ 8	**aaa authentication login default group** *group-name***local** 例： `Device(config)# aaa authentication login default group ISE local`	ログイン時に、指定されたグループ名をデフォルトのローカル AAA 認証として設定します。
ステップ 9	**aaa authentication login** *list-name* **none** 例： `Device(config)# aaa authentication login NOAUTH none`	システムへのログイン中に認証が不要であることを指定します。
ステップ 10	**aaa authorization exec default group** *group-name***local** 例： `Device(config)# aaa authorization exec default group ISE local`	許可を実行して、EXEC シェルの実行がユーザに許可されているかどうかを確認します。
ステップ 11	**aaa session-id common** 例：	指定のコールに対して送信されたセッション ID 情報が同じになるようにします。

	コマンドまたはアクション	目的
	`Device(config)# aaa session-id common`	
ステップ 12	**line console** *number* 例： `Device(config)# line console 0`	設定する特定の回線を識別し、ライン コンフィギュレーション モードを開始します。
ステップ 13	**login authentication** *authentication-list* 例： `Device(config-line)# login authentication NOAUTH`	ログインに対する AAA 認証をイネーブルにします。
ステップ 14	**end** 例： `Device(config-line)# end`	回線コンフィギュレーションモードを終了します。続いて、特権 EXEC モードに戻ります。

RESTCONF の Cisco IOS HTTP サービスの有効化

RESTCONF インターフェイスを使用するには、次の作業を行います。

手順の概要

1. **enable**
2. **configure terminal**
3. **restconf**
4. **ip http secure-server**
5. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**restconf** 例： `Device(config)# restconf`	ネットワーク デバイスで RESTCONF インターフェイスを有効にします。
ステップ 4	**ip http secure-server** 例：	セキュア HTTP（HTTPS）サーバをイネーブルにします。

	コマンドまたはアクション	目的
	`Device(config)# ip http secure-server`	
ステップ 5	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードを開始します。

RESTCONF の設定の検証

スタートアップ コンフィギュレーションを使用してデバイスが起動すると、*nginx* プロセスが実行中になります。ただし、DMI プロセスは有効にはなりません。

次の **show platform software yang-management process monitor** コマンドの出力例は、nginx プロセスが実行中であることを示しています。

```
Device# show platform software yang-management process monitor

COMMAND           PID S    VSZ   RSS %CPU %MEM     ELAPSED
nginx           27026 S 332356 18428  0.0  0.4       01:34
nginx           27032 S 337852 13600  0.0  0.3       01:34
```

NGINX は、プロキシ Web サーバとして機能する内部 Web サーバで、Transport Layer Security（TLS）ベースの HTTPS を提供します。HTTPS を介して送信された RESTCONF 要求は、最初に NGINX プロキシ Web サービスによって受信され、さらに要求が構文/セマンティックチェックのために confd Web サーバに転送されます。

次の **show platform software yang-management process** コマンドの出力例は、スタートアップ コンフィギュレーションを使用してデバイスが起動されたときのすべてのプロセスのステータスを示しています。

```
Device# show platform software yang-management process

confd            : Not Running
nesd             : Not Running
syncfd           : Not Running
ncsshd           : Not Running
dmiauthd         : Not Running
nginx            : Running
ndbmand          : Not Running
pubd             : Not Running
```

restconf コマンドが設定されている場合、nginx プロセスが再起動され、DMI プロセスが起動されます。

次の **show platform software yang-management process** コマンドの出力例は、nginx プロセス と DMI プロセスが起動して実行中であることを示しています。

```
Device# show platform software yang-management process

confd            : Running
nesd             : Running
syncfd           : Running
```

```
ncsshd           : Not Running ! NETCONF-YANG is not configured, hence ncsshd process
is in not running.
dmiauthd         : Running
vtyserverutild   : Running
opdatamgrd       : Running
nginx            : Running ! nginx process is up due to the HTTP configuration, and it
is restarted when RESTCONF is enabled.
ndbmand          : Running
```

次の show platform software yang-management process monitor コマンドの出力例では、すべてのプロセスに関する詳細情報が表示されています。

```
Device#show platform software yang-management process monitor

COMMAND           PID S    VSZ   RSS %CPU %MEM     ELAPSED
confd           28728 S 860396 168496 42.2  4.2       00:12
confd-startup.s 28448 S  19664  4496  0.2  0.1       00:12
dmiauthd        29499 S 275356 23340  0.2  0.5       00:10
ndbmand         29321 S 567232 65564  2.1  1.6       00:11
nesd            29029 S 189952 14224  0.1  0.3       00:11
nginx           29711 S 332288 18420  0.6  0.4       00:09
nginx           29717 S 337636 12216  0.0  0.3       00:09
pubd            28237 S 631848 68624  2.1  1.7       00:13
syncfd          28776 S 189656 16744  0.2  0.4       00:12
```

AAA と RESTCONF インターフェイスが設定され、nginx プロセスと関連する DMI プロセスが実行中になった後、デバイスは RESTCONF 要求を受信できる状態になります。

NETCONF/RESTCONF セッションのステータスを表示するには、**show netconf-yang sessions** コマンドを使用します。

```
Device# show netconf-yang sessions

R: Global-lock on running datastore
C: Global-lock on candidate datastore
S: Global-lock on startup datastore

Number of sessions : 1

session-id  transport    username              source-host            global-lock
--------------------------------------------------------------------------------
19          netconf-ssh  admin                 2001:db8::1             None
```

NETCONF/RESTCONF セッションに関する詳細情報を表示するには、**show netconf-yang sessions detail** コマンドを使用します。

```
Device# show netconf-yang sessions detail

R: Global-lock on running datastore
C: Global-lock on candidate datastore
S: Global-lock on startup datastore

Number of sessions     : 1

session-id             : 19
transport              : netconf-ssh
username               : admin
source-host            : 2001:db8::1
```

```
login-time             : 2018-10-26T12:37:22+00:00
in-rpcs                : 0
in-bad-rpcs            : 0
out-rpc-errors         : 0
out-notifications      : 0
global-lock            : None
```

RESTCONF プロトコルの設定例

例：RESTCONF プロトコルの設定

RESTCONF 要求（HTTPS Verb）：

次に、ターゲット リソースで許可されている HTTPS Verb を示す RESTCONF 要求の例を示します。この例では **logging monitor** コマンドを使用しています。

```
root:~# curl -i -k -X "OPTIONS"
"https://10.85.116.30:443/restconf/data/Cisco-IOS-XE-native:native/logging/monitor/severity"
 \
>      -H 'Accept: application/yang-data+json' \
>      -u 'admin:admin'
HTTP/1.1 200 OK
Server: nginx
Date: Mon, 23 Apr 2018 15:27:57 GMT
Content-Type: text/html
Content-Length: 0
Connection: keep-alive
Allow: DELETE, GET, HEAD, PATCH, POST, PUT, OPTIONS    >>>>>>>>>>>    Allowed methods
Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
Accept-Patch: application/yang-data+xml, application/yang-data+json
Pragma: no-cache

root:~#
```

POST（作成）要求

POST 操作では、ターゲット デバイスに存在しないコンフィギュレーションが作成されます。

（注） 実行コンフィギュレーションで **logging monitor** コマンドを使用できないことを確認してください。

次の POST 要求の例では **logging monitor alerts** コマンドを使用しています。

```
Device:~# curl -i -k -X "POST"
"https://10.85.116.30:443/restconf/data/Cisco-IOS-XE-native:native/logging/monitor" \
>      -H 'Content-Type: application/yang-data+json' \
>      -H 'Accept: application/yang-data+json' \
>      -u 'admin:admin' \
```

```
>      -d $'{
>   "severity": "alerts"
> }'
HTTP/1.1 201 Created
Server: nginx
Date: Mon, 23 Apr 2018 14:53:51 GMT
Content-Type: text/html
Content-Length: 0
Location:
https://10.85.116.30/restconf/data/Cisco-IOS-XE-native:native/logging/monitor/severity
Connection: keep-alive
Last-Modified: Mon, 23 Apr 2018 14:53:51 GMT
Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
Etag: 1524-495231-97239
Pragma: no-cache

Device:~#
```

PUT：（作成または置換）要求：

指定されたコマンドがデバイスに存在しない場合は、POST 要求によって作成されます。ただし、実行コンフィギュレーションにすでに存在する場合は、この要求によってコマンドが置き換えられます。

次の PUT 要求の例では **logging monitor warnings** コマンドを使用しています。

```
Device:~# curl -i -k -X "PUT"
"https://10.85.116.30:443/restconf/data/Cisco-IOS-XE-native:native/logging/monitor/severity"
 \
>      -H 'Content-Type: application/yang-data+json' \
>      -H 'Accept: application/yang-data+json' \
>      -u 'admin:admin' \
>      -d $'{
>   "severity": "warnings"
> }'
HTTP/1.1 204 No Content
Server: nginx
Date: Mon, 23 Apr 2018 14:58:36 GMT
Content-Type: text/html
Content-Length: 0
Connection: keep-alive
Last-Modified: Mon, 23 Apr 2018 14:57:46 GMT
Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
Etag: 1524-495466-326956
Pragma: no-cache

Device:~#
```

PATCH：（更新）要求

次の PATCH 要求の例では **logging monitor informational** コマンドを使用しています。

```
Device:~# curl -i -k -X "PATCH"
"https://10.85.116.30:443/restconf/data/Cisco-IOS-XE-native:native" \
>      -H 'Content-Type: application/yang-data+json' \
>      -H 'Accept: application/yang-data+json' \
>      -u 'admin:admin' \
>      -d $'{
>   "native": {
```

```
>     "logging": {
>       "monitor": {
>         "severity": "informational"
>       }
>     }
>   }
> }'
HTTP/1.1 204 No Content
Server: nginx
Date: Mon, 23 Apr 2018 15:07:56 GMT
Content-Type: text/html
Content-Length: 0
Connection: keep-alive
Last-Modified: Mon, 23 Apr 2018 15:07:56 GMT
Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
Etag: 1524-496076-273016
Pragma: no-cache
Device:~#
```

GET 要求（読み取り）

次の GET 要求の例では **logging monitor informational** コマンドを使用しています。

```
Device:~# curl -i -k -X "GET"
"https://10.85.116.30:443/restconf/data/Cisco-IOS-XE-native:native/logging/monitor/severity"
 \
>      -H 'Accept: application/yang-data+json' \
>      -u 'admin:admin'
HTTP/1.1 200 OK
Server: nginx
Date: Mon, 23 Apr 2018 15:10:59 GMT
Content-Type: application/yang-data+json
Transfer-Encoding: chunked
Connection: keep-alive
Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
Pragma: no-cache

{
  "Cisco-IOS-XE-native:severity": "informational"
}
Device:~#
```

DELETE 要求（コンフィギュレーションの削除）

```
Device:~# curl -i -k -X "DELETE"
"https://10.85.116.30:443/restconf/data/Cisco-IOS-XE-native:native/logging/monitor/severity"
 \
>      -H 'Content-Type: application/yang-data+json' \
>      -H 'Accept: application/yang-data+json' \
>      -u 'admin:admin'
HTTP/1.1 204 No Content
Server: nginx
Date: Mon, 23 Apr 2018 15:26:05 GMT
Content-Type: text/html
Content-Length: 0
Connection: keep-alive
```

```
Last-Modified: Mon, 23 Apr 2018 15:26:05 GMT
Cache-Control: private, no-cache, must-revalidate, proxy-revalidate
Etag: 1524-497165-473206
Pragma: no-cache

linux_host:~#
```

RESTCONF プロトコルの関連資料

関連資料

関連項目	マニュアル タイトル
IOS-XE、IOS-XR、および NX-OS プラットフォームのさまざまなリリースの YANG データ モデル	開発者にわかりやすい方法で Cisco YANG モデルにアクセスするには、GitHub リポジトリを複製し、vendor/cisco サブディレクトリに移動します。ここでは、IOS XE、IOS-XR、および NX-OS プラットフォームのさまざまなリリースのモデルを使用できます。

標準および RFC

標準/RFC	タイトル
RFC 6020	*YANG：Network Configuration Protocol（NETCONF）向けデータ モデリング言語*
RFC 8040	*RESTCONF プロトコル*
RFC 8072	*YANG パッチメディアタイプ*

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	https://www.cisco.com/c/en/us/support/index.html

RESTCONF プロトコルの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

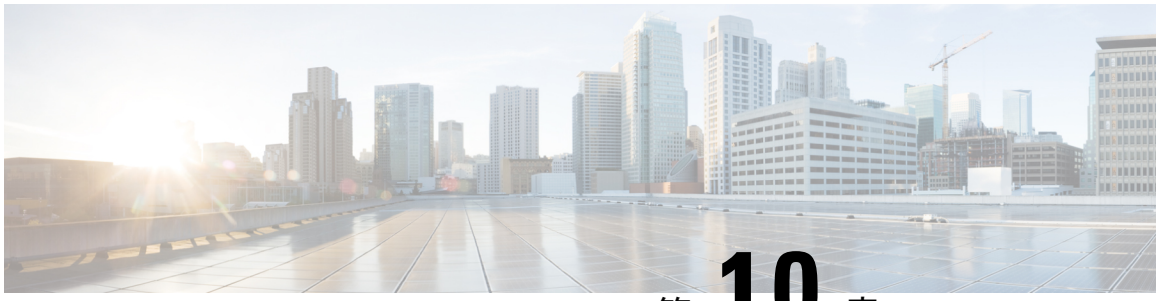
プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *14 : RESTCONF* プロトコルの機能情報

機能名	リリース	機能情報
RESTCONF プロトコル	Cisco IOS XE Everest 16.6.1	RESTCONF は、YANG モデルで定義されている設定データ、状態データ、データモデル固有の RPC の操作およびイベント通知にアクセスするための、標準メカニズムに基づくプログラマチック インターフェイスを提供します。 この機能が次のプラットフォームで追加されました。 • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 アグリゲーション サービス ルータ • Cisco Cloud Services Router 1000V シリーズ 次のコマンドが導入または変更されました：**ip http server** および **restconf**
	Cisco IOS XE Fuji 16.8.1a	Cisco IOS XE Fuji 16.8.1a では、この機能は次のプラットフォームに実装されていました。 • Cisco 1000 シリーズ サービス統合型ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 および 9500 ハイ パフォーマンス シリーズ スイッチ • Cisco cBR-8 コンバージド ブロードバンド ルータ • Cisco Network Convergence System 4200 シリーズ
	Cisco IOS XE Fuji 16.9.2	Cisco IOS XE Fuji 16.9.2 では、この機能は次のプラットフォームに実装されていました。 • Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300L SKU
	Cisco IOS XE Gibraltar 16.11.1	

機能名	リリース	機能情報
		Cisco IOS XE Gibraltar 16.11.1 では、この機能は次のプラットフォームに実装されていました。 • Cisco Catalyst 9600 シリーズ スイッチ • Cisco Catalyst 9800-CL ワイヤレスコントローラ • Cisco Catalyst 9800-40 ワイヤレスコントローラ • Cisco Catalyst 9800-80 ワイヤレスコントローラ • Cisco Network Convergence System 520 シリーズ
	Cisco IOS XE Gibraltar 16.12.1	この機能は、Cisco IOS XE Gibraltar 16.12.1 で、Cisco Catalyst 9800-L ワイヤレスコントローラに実装されました。
	Cisco IOS XE Amsterdam 17.3.1	この機能は、Cisco IOS XE Amsterdam 17.3.1 で次のプラットフォームに実装されました。 • Cisco Catalyst 8200 シリーズ エッジ プラットフォーム • Cisco Catalyst 8300 シリーズ エッジ プラットフォーム • Cisco Catalyst 8500 および 8500L シリーズ エッジ プラットフォーム
	Cisco IOS XE Bengaluru 17.4.1	この機能は、Cisco IOS XE Bengaluru 17.4.1 で、Cisco Catalyst 8000V Edge ソフトウェアに実装されました。

機能名	リリース	機能情報
RESTCONF YANG パッチのサポート	Cisco IOS XE Amsterdam 17.1.1	RESTCONF は、RFC 8072 で指定されている YANG パッチメディアタイプをサポートしています。 この機能は、次のプラットフォームに実装されていました。 • Cisco 1000 シリーズ サービス統合型ルータ • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 1000 アグリゲーション サービス ルータ（ASR1000-RP2、ASR1000-RP3、ASR1001-HX、ASR1001-X、ASR1002-HX、ASR1002-X） • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco cBR-8 コンバージド ブロードバンド ルータ • Cisco Cloud Services Router 1000V シリーズ • Cisco Network Convergence System 520 シリーズ • Cisco Network Convergence System 4200 シリーズ



第 10 章

NETCONF および RESTCONF のサービスレベル ACL

このモジュールでは、NETCONF および RESTCONF でサポートされるサービスレベル ACL とその設定方法について説明します。

NETCONF および RESTCONF のサービスレベル ACL に関する情報

NETCONF および RESTCONF のサービスレベル ACL の概要

NETCONF および RESTCONF セッションの IPv4 または IPv6 アクセス制御リスト（ACL）を設定できます。設定された ACL に準拠していないクライアントは、NETCONF または RESTCONF サブシステムへのアクセスを許可されません。サービスレベルの ACL が設定されている場合、NETCONF-YANG および RESTCONF 接続要求は送信元 IP アドレスに基づいてフィルタリングされます。

サービスレベルの ACL が設定されていない場合、すべての NETCONF-YANG および RESTCONF 接続要求がサブシステムに許可されます。

（注） 名前付き ACL のみがサポートされます。番号付き ACL はサポートされません。

NETCONF および RESTCONF のサービスレベル ACL の設定方法

NETCONF-YANG セッションの ACL の設定

NETCONF-YANG セッションの IP アクセスリストまたは IPv6 アクセスリストを設定できます。

手順の概要

1. **enable**
2. **configure terminal**
3. • **ip access-list** {**standard** | **extended**} *access-list-name*
   • **ipv6 access-list** *access-list-name*
4. **permit** {*host-address* | *host-name* | **any**} [*wildcard*]
5. **deny** {*host-address* | *host-name* | **any**} [*wildcard*]
6. **exit**
7. **netconf-yang ssh** {{**ipv4** | **ipv6** }**access-list name** *access-list-name*} | **port** *port-number*}
8. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	• **ip access-list** {**standard**	**extended**} *access-list-name*
		• **ipv6 access-list** *access-list-name*
		例：
		`Device(config)# ip access-list standard acl1_permit`
		`Device(config)# ipv6 access-list ipv6-acl1_permit`
			• 標準の IP アクセスリストを指定して、標準のアクセス リスト コンフィギュレーション モードを開始します。
			• IPv6 アクセスリストを指定して、標準の IPv6 アクセス リスト コンフィギュレーション モードを開始します。

ステップ 4	**permit** {*host-address*	*host-name*	**any**} [*wildcard*] 例：	パケットを許可する IP/IPv6 アクセスリストの条件を設定します。

	コマンドまたはアクション	目的
	`Device(config-std-nacl)# permit 192.168.255.0 0.0.0.255`	
ステップ 5	**deny** {*host-address* | *host-name* | **any**} [*wildcard*] 例： `Device(config-std-nacl)# deny any`	パケットを拒否する IP または IPv6 アクセスリストの条件を設定します。
ステップ 6	**exit** 例： `Device(config-std-nacl)# exit`	標準のアクセスリストコンフィギュレーションモードを終了し、グローバル コンフィギュレーション モードに戻ります。
ステップ 7	**netconf-yang ssh** {{**ipv4** | **ipv6** }**access-list name** *access-list-name*} | **port** *port-number*} 例： `Device(config)# netconf-yang ssh ipv4 access-list name acl1_permit`	NETCONF-YANG セッションの ACL を設定します。
ステップ 8	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

RESTCONF セッションの ACL の設定

RESTCONF セッションの IP アクセスリストまたは IPv6 アクセスリストを設定できます。

手順の概要

1. **enable**
2. **configure terminal**
3. • **ip access-list** {**standard** | **extended**} *access-list-name*
   • **ipv6 access-list** *access-list-name*
4. **permit** {*protocol-number* | *ipv6-source-address* | *ipv6-source-prefix* | *protocol*}**any**
5. **deny** {*protocol-number* | *ipv6-source-address* | *ipv6-source-prefix* | *protocol*}**any any**
6. **exit**
7. **restconf** {**ipv4** | **ipv6** }**access-list name** *access-list-name*
8. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	• **ip access-list** {**standard** | **extended**} *access-list-name*
	• **ipv6 access-list** *access-list-name*
	例：
	`Device(config)# ip access-list standard acl1_permit`
	`Device(config)# ipv6 access-list ipv6-acl1_permit`
		• 標準の IP アクセスリストを指定して、標準のアクセス リスト コンフィギュレーション モードを開始します。
		• IPv6 アクセスリストを指定して、標準の IPv6 アクセス リスト コンフィギュレーション モードを開始します。

ステップ 4	**permit** {*protocol-number* | *ipv6-source-address* | *ipv6-source-prefix* | *protocol*} **any** 例： `Device(config-ipv6-acl)# permit ipv6 2001:db8::1/32 any`	パケットを許可する IPv6 アクセスリストの条件を設定します。
ステップ 5	**deny** {*protocol-number* | *ipv6-source-address* | *ipv6-source-prefix* | *protocol*} **any any** 例： `Device(config-ipv6-acl)# deny ipv6 any any`	パケットを拒否する IPv6 アクセスリストの条件を設定します。
ステップ 6	**exit** 例： `Device(config-ipv6-acl)# exit`	IPv6 アクセス リスト コンフィギュレーション モードを終了して、グローバル コンフィギュレーション モードに戻ります。
ステップ 7	**restconf** {**ipv4** | **ipv6** } **access-list name** *access-list-name* 例： `Device(config)# restconf ipv6 access-list name ipv6-acl1_permit`	RESTCONF セッションの ACL を設定します。
ステップ 8	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

NETCONF および RESTCONF のサービスレベル ACL の設定例

例：NETCONF セッションの ACL の設定

```
Device# enable
Device# configure terminal
Device(config)# ip access-list standard acl1_permit
Device(config-std-nacl)# permit 192.168.255.0 0.0.0.255
Device(config-std-nacl)# deny any
Device(config-std-nacl)# exit
Device(config)# netconf-yang ssh ipv4 access-list name acl1_permit
Device(config)# end
```

例：RESTCONF セッションの ACL の設定

```
Device# enable
Device# configure terminal
Device(config)# ipv6 access-list ipv6-acl1_permit
Device(config-ipv6-acl)# permit ipv6 2001:db8::1/32 any
Device(config-ipv6-acl)# deny ipv6 any any
Device(config-ipv6-acl)# exit
Device(config)# restconf ipv6 access-list name ipv6-acl1_permit
Device(config)# end
```

NETCONF および RESTCONF のサービスレベル ACL に関するその他の資料

関連資料

関連項目	マニュアル タイトル
NETCONF-YANG	NETCONF プロトコル
RESTCONF	RESTCONF プロトコル
プログラマビリティ コマンド	プログラマビリティ コマンド リファレンス

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

NETCONF および RESTCONF のサービスレベル ACL の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *15 : NETCONF* および *RESTCONF* のサービスレベル *ACL* の機能情報

機能名	リリース	機能情報
NETCONF および RESTCONF のサービスレベル ACL	Cisco IOS XE Everest 16.11.1	NETCONF および RESTCONF セッションのアクセス制御リスト（ACL）を設定できます。設定された ACL に準拠していないクライアントは、NETCONF または RESTCONF サブシステムへのアクセスを許可されません。 次のコマンドが導入または変更されました：**netconf-yang ssh access-list**および**restconf access-list** • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ（RSP2） • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco Catalyst IE 3200、3300、3400 高耐久性シリーズ • Cisco エンベデッド サービス 3300 シリーズ スイッチ • Cisco IR1101 耐環境性能 サービス統合型ルータ • Cisco Network Convergence System 4200 シリーズ • Cisco Network Convergence System 520 シリーズ

第 11 章

gNMI プロトコル

この機能では、gRPC ネットワーク管理インターフェイス（gNMI）の機能を使用したモデル駆動型の設定と運用データの取得、およびリモートプロシージャ コール（RPC）の取得、設定、登録について説明します。gNMI バージョン 0.4.0 がサポートされています。

gNMI プロトコルの制約事項

機能には、次のような制約事項が適用されます。

- JSON、BYTES、PROTO、および ASCI エンコーディング オプションはサポートされていません。

  JSON IETF キーには、次の要素の名前空間が親とは異なる YANG プレフィックスが含まれている必要があります。たとえば、openconfig-vlan.yang の拡張から派生した routed-vlan は、親ノードの名前空間とは異なるため（親ノードはプレフィックス oc-if を持ちます）、oc-vlan:routed-vlan と入力する必要があります。

- GetRequest：
  - 運用データのフィルタリングはサポートされていません。
  - モデルの使用はサポートされていません。これらは、Get RPC コールへの応答として返す必要があるデータ要素を定義するスキーマ定義モジュールを示す一連のモデル データ メッセージです。

- GetResponse：
  - エイリアスはサポートされていません。これは、通知メッセージの中で指定されたプレフィックスのエイリアスを提供する文字列です。

- 削除はサポートされていません。これは、データ ツリーから削除する一連のパスです。

gNMI プロトコルの概要

gNMI について

gNMI は Google によって開発された gRPC ネットワーク管理インターフェイスです。gNMI はネットワーク デバイスの設定をインストール、操作、および削除し、また、運用データの表示も実行するメカニズムです。gNMI を通じて提供されるコンテンツは YANG を使用してモデル化できます。

gRPC は、クラウド サーバと通信するモバイル クライアントを使用して低遅延で拡張可能な配布を実現するために Google によって開発されたリモート プロシージャ コールです。gRPC は gNMI を伝送し、データと動作要求を公式化して送信する手段を提供します。

gNMI サービスの障害が発生した場合、gNMI ブローカ（GNMIB）によって、up から down への動作状態の変化が示され、データベースが起動して実行されるまではすべての RPC がサービス利用不可のメッセージを返します。リカバリ時には、GNMIB によって down から up への動作状態の変化が示され、RPC の通常の処理が再開されます。

gNMI は<subscribe> RPC サービスをサポートします。詳細については、「モデル駆動型テレメトリ」の章を参照してください。

YANG データ ツリーの JSON IETF エンコーディング

RFC 7951 では、YANG データ ツリーとそのサブツリーの JavaScript オブジェクト表記（JSON）エンコーディングが規定されています。gNMI は、コンテンツ層でのデータのエンコードに JSON を使用します。

JSON タイプは、値が JSON 文字列としてエンコードされていることを示します。JSON_IETF でエンコードされたデータは、RFC 7951 で規定されている JSON シリアル化のルールに準拠している必要があります。クライアントとターゲットの両方が JSON エンコーディングをサポートしている必要があります。

YANG データ ノード（リーフ、コンテナ、リーフリスト、リスト、anydata ノード、および anyxml ノード）のインスタンスは、JSON オブジェクトまたは名前と値のペアのメンバーとしてエンコードされます。エンコーディング ルールは、設定データ、状態データ、RPC 操作のパラメータ、アクション、通知など、すべてのタイプのデータ ツリーで同じです。

データ ノード インスタンスはすべて名前と値のペアとしてエンコードされ、その名前はデータ ノード識別子から形成されます。値は、データ ノードのカテゴリによって異なります。

リーフデータノード

リーフ ノードは、データ ツリー内に値がありますが子はありません。リーフ インスタンスは、名前と値のペアとしてエンコードされます。この値には、リーフのタイプに応じて、文字列、数値、リテラル true または false、または特殊な配列 [null] を使用できます。指定されたパスのデータ項目がリーフ ノードの場合（子が存在せず、関連付けられた値を持つ）、そのリーフの値が直接エンコードされます（そのままの JSON 値が含まれています。JSON オブジェクトは必要ありません）。

次に、リーフ ノード定義の例を示します。

```
leaf foo {
 type uint8;
}
```

次に、JSON でエンコードされた有効なインスタンスを示します。

```
"foo": 123
```

gNMI GET Request

gNMI Get RPC は、データ ツリーから、1 つ以上の設定属性、状態属性、派生状態属性、またはサポートされているモードに関連付けられたすべての属性を取得する方法を指定します。データ ツリーから値を取得するために、GetRequest がクライアントからターゲットに送信されます。GetRequest への応答として GetResponse が送信されます。

GetRequest の JSON 構造

次に、GetRequest JSON の構造の例を示します。GetRequest と GetResponse の両方が表示されます。

GetRequest

```
The following is a path for the
openconfig-interfaces model
++++++++ Sending get request: ++++++++
path {
  elem {
    name: "interfaces"
  }
  elem {
    name: "interface"
    key {
      key: "name"
      value: "Loopback111"
    }
  }
}
```

GetResponse

```
encoding: JSON_IETF
++++++++ Received get response: ++++++++
notification {
  timestamp: 1521699434792345469
  update {
    path {
      elem {
```

```
    name: "interfaces"
  }
  elem {
    name: "interface"
    key {
      key: "name"
      value: "\"Loopback111\""
    }
  }
}


val {
  json_ietf_val: "{\n\t\"openconfig-interfaces:name\":\t\
  "Loopback111\",\n\t\
   "openconfig-interfaces:config\":\t{\n\t\t\
   "openconfig-interfaces:type\":\t\"ianaift:
    softwareLoopback\",\n\t\t\
   "openconfig-interfaces:name\":\t\"Loopback111\",\n\t\t\
   "openconfig-interfaces:enabled\":\t\"true\"\n\t},\n\t\
   "openconfig-interfaces:state\":\t{\n\t\t\
   "openconfig-interfaces:type\":\t\"ianaift:
    softwareLoopback\",\n\t\t\
   "openconfig-interfaces:name\":\t\"Loopback111\",\n\t\t\
   "openconfig-interfaces:enabled\":\t\"true\",\n\t\t\
   "openconfig-interfaces:ifindex\":\t52,\n\t\t\


   "openconfig-interfaces:admin-status\":\t\"UP\",\n\t\t\
   "openconfig-interfaces:oper-status\":\t\"UP\",\n\t\t\
   "openconfig-interfaces:last-change\":\t2018,\n\t\t\
   "openconfig-interfaces:counters\":\t{\n\t\t\t\
   "openconfig-interfaces:in-octets\":\t0,\n\t\t\t\
   "openconfig-interfaces:in-unicast-pkts\":\t0,\n\t\t\t\
   "openconfig-interfaces:in-broadcast-pkts\":\t0,\n\t\t\t\
   "openconfig-interfaces:in-multicast-pkts\":\t0,\n\t\t\t\
   "openconfig-interfaces:in-discards\":\t0,\n\t\t\t\
   "openconfig-interfaces:in-errors\":\t0,\n\t\t\t\
   "openconfig-interfaces:in-unknown-protos\":\t0,\n\t\t\t\
   "openconfig-interfaces:out-octets\":\t0,\n\t\t\t\
   "openconfig-interfaces:out-unicast-pkts\":\t0,\n\t\t\t\
   "openconfig-interfaces:out-broadcast-pkts\":\t0,\n\t\t\t\
   "openconfig-interfaces:out-multicast-pkts\":\t0,\n\t\t\t\
   "openconfig-interfaces:out-discards\":\t0,\n\t\t\t\
   "openconfig-interfaces:out-errors\":\t0,\n\t\t\t\
   "openconfig-interfaces:last-clear\":\t2018\n\t\t},\n\t\t\

   "openconfig-platform:hardware-port\":\t\
   "Loopback111\"\n\t},\n\t\
   "openconfig-interfaces:subinterfaces\":\t{\n\t\t\
   "openconfig-interfaces:index\":\t0,\n\t\t\
   "openconfig-interfaces:config\":\t{\n\t\t\t\
   "openconfig-interfaces:index\":\t0,\n\t\t\t\
   "openconfig-interfaces:name\":\t\"Loopback111\",\n\t\t\t\
   "openconfig-interfaces:enabled\":\t\"true\"\n\t\t},\n\t\t\
   "openconfig-interfaces:state\":\t{\n\t\t\t\
   "openconfig-interfaces:index\":\t0,\n\t\t\t\
   "openconfig-interfaces:name\":\t\"Loopback111.0\",\n\t\t\t\
   "openconfig-interfaces:enabled\":\t\"true\",\n\t\t\t\
   "openconfig-interfaces:admin-status\":\t\"UP\",\n\t\t\t\
   "openconfig-interfaces:oper-status\":\t\"UP\",\n\t\t\t\
   "openconfig-interfaces:last-change\":\t2018,\n\t\t\t\
   "openconfig-interfaces:counters\":\t{\n\t\t\t\t\
   "openconfig-interfaces:in-octets\":\t0,\n\t\t\t\t\
```

```
       "openconfig-interfaces:in-unicast-pkts\":\t0,\n\t\t\t\t\
       "openconfig-interfaces:in-broadcast-pkts\":\t0,\n\t\t\t\t\
       "openconfig-interfaces:in-multicast-pkts\":\t0,\n\t\t\t\t\
       "openconfig-interfaces:in-discards\":\t0,\n\t\t\t\t\
       "openconfig-interfaces:in-errors\":\t0,\n\t\t\t\t\

       "openconfig-interfaces:out-octets\":\t0,\n\t\t\t\t\
       "openconfig-interfaces:out-unicast-pkts\":\t0,\n\t\t\t\t\
       "openconfig-interfaces:out-broadcast-pkts\":\t0,\n\t\t\t\t\
       "openconfig-interfaces:out-multicast-pkts\":\t0,\n\t\t\t\t\
       "openconfig-interfaces:out-discards\":\t0,\n\t\t\t\t\
       "openconfig-interfaces:out-errors\":\t0,\n\t\t\t\t\
       "openconfig-interfaces:last-clear\":\
        t2018\n\t\t\t}\n\t\t},\n\t\t\
       "openconfig-if-ip:ipv6\":\t{\n\t\t\t\
       "openconfig-if-ip:config\":\t\"false\",\n\t\t\t\
       "openconfig-if-ip:state\":\t\"false\"\n\t\t}\n\t}\n}"
    }
  }
}
```

リーフ値に対する GetRequest

次に、リーフ値に対するGetRequestの例を示します。GetRequest と、対応する GetResponse の両方が表示されます。

GetRequest

```
++++++++ Sending get request: ++++++++
path {
  elem {
    name: "interfaces"
  }
  elem {
    name: "interface"
    key {
      key: "name"
      value: "Loopback111"
    }
  }
  elem {
    name: "state"
  }
  elem {
    name: "oper-status"
  }
}
```

GetResponse

```
encoding: JSON_IETF
++++++++ Received get response: ++++++++
notification {
  timestamp: 1521699326012374332
  update {
    path {
      elem {
        name: "interfaces"
      }
      elem {
```

```
        name: "interface"
        key {
          key: "name"
          value: "\"Loopback111\""
        }
      }
      elem {
        name: "state"
      }
      elem {
        name: "oper-status"
      }
    }
    val {
      json_ietf_val: "\"UP\""
    }
  }
}
```

gNMI SetRequest

Set RPC は、サポートされているモデルに関連付けられた 1 つ以上の設定可能な属性を設定する方法を指定します。データツリー内の値を更新するために、SetRequest がクライアントからターゲットに送信されます。

SetRequest は JSON キーもサポートしており、キーには YANG プレフィックスが含まれている必要があります。このプレフィックスでは要素の名前空間が親とは異なります。

たとえば、openconfig-vlan.yang の拡張から派生した routed-vlan は、親ノードの名前空間とは異なるため（親ノードのプレフィックスは oc-if）、oc-vlan:routed-vlan と入力する必要があります。

1 つの SetRequest に含まれる削除、置換、および更新は、全体で 1 つのトランザクション セットとして扱われます。トランザクションのいずれかの下位要素で障害が発生した場合は、トランザクション全体が拒否されてロールバックされます。SetRequest に対して SetResponse が返信されます。

表 *16 : SetRequest* の *JSON* 構造の例

SetRequest	SetResponse

++++++++ Sending set request: ++++++++
update {
  path {
  elem {
  name: "interfaces"
  }
  elem {
  name: "interface"
  key {
  key: "name"
  value: "Loopback111"
  }
  }
  elem {
  name: "config"
  }
  }
  val {
  json_ietf_val:
"{\"openconfig-interfaces:enabled\":\"false\"}"
 	
  }
}
	++++++++ Received set response: ++++++++
	response {
	  path {
	  elem {
	  name: "interfaces"
	  }
	  elem {
	  name: "interface"
	  key {
	  key: "name"
	  value: "Loopback111"
	  }
	  }
	  elem {
	  name: "config"
	  }
	  }
	  op: UPDATE
	}
	timestamp: 1521699342123890045

表 *17 :* リーフ値での *SetRequest* の例

SetRequest	SetResponse

++++++++ Sending set request: ++++++++
update {
  path {
  elem {
  name: "interfaces"
  }
  elem {
  name: "interface"
  key {
  key: "name"
  value: "Loopback111"
  }
  }
  elem {
  name: "config"
  }
  elem {
  name: "description"
  }
  }
  val {
  json_ietf_val: "\"UPDATE DESCRIPTION\""
 	
  }
}
	++++++++ Received set response: ++++++++
	response {
	  path {
	  elem {
	  name: "interfaces"
	  }
	  elem {
	  name: "interface"
	  key {
	  key: "name"
	  value: "Loopback111"
	  }
	  }
	  elem {
	  name: "config"
	  }
	  elem {
	  name: "description"
	  }
	  }
	  op: UPDATE
	}
	timestamp: 1521699342123890045

gNMI の名前空間

名前空間は、メッセージの origin フィールドで使用されるパスプレフィックスを指定します。

ここでは、Cisco IOS XE Gibraltar 16.10.1 以降のリリースで使用される名前空間について説明します。

- RFC 7951 で指定された名前空間：パス プレフィックスは、RFC 7951 で定義されている YANG モジュール名を使用します。

  RFC 7951 で指定された値のプレフィックスは、YANG モジュール名を使用します。

  値のプレフィックスは、選択されたパスプレフィックスの名前空間の影響を受けません。次に、RFC 7951 で指定された値のプレフィックスの例を示します。

```
val {
   json_ietf_val:"{
     "openconfig-interfaces:config": {
       "openconfig-interfaces:description":
         "DESCRIPTION"
     }
   }"
 }
```

  RFC 7951 で指定された名前空間プレフィックスは、YANG モジュール名も使用します。たとえば、ループバック インターフェイスへの openconfig パスは次のようになります。

```
/openconfig-interfaces:interfaces/interface[name=Loopback111]/
```

  次の例は、RFC 7951 の名前空間指定を使用した gNMI パスを示しています。

```
path {
  origin: "rfc7951"
  elem {
    name: "openconfig-interface:interfaces"
  }
  elem {
    name: "interface"
    key {
      key: "name"
      value: "Loopback111"
    }
  }
}
```

- openconfig：パスプレフィックスを使用しません。これらは openconfig モデルへのパスでのみ使用できます。

  openconfig 名前空間プレフィックスの動作は、発信元または名前空間が指定されていない場合と同じです。たとえば、ループバック インターフェイスへの openconfig パスは次のようになります。

```
/interfaces/interface[name=Loopback111]/
```

  次の例は、openconfig 名前空間指定を使用した gNMI パスを示しています。

```
path {
  origin: "openconfig"
  elem {
    name: "interfaces"
```

```
  }
  elem {
    name: "interface"
    key {
      key: "name"
      value: "Loopback111"
    }
  }
}
```

- 空：openconfig プレフィックスと同じです。これがデフォルトです。

  次の例は、空の openconfig 名前空間指定を使用した gNMI パスを示しています。

```
path {
elem {
    name: "interfaces"
  }
  elem {
    name: "interface"
    key {
      key: "name"
      value: "Loopback111"
    }
  }
}
```

ここでは、Cisco IOS XE Gibraltar 16.10.1 より前のリリースで使用されるパス プレフィックスについて説明します。

ここでは、パス プレフィックスは、YANG モジュール定義で定義されている YANG モジュール プレフィックスを使用します。たとえば、ループバック インターフェイスへの openconfig パスは次のようになります。

```
/oc-if:interfaces/interface[name=Loopback111]/
```

次の例は、従来の名前空間指定を使用した gNMI パスを示しています。

```
path {
  origin: "legacy"
  elem {
    name: "oc-if:interfaces"
  }
  elem {
    name: "interface"
    key {
      key: "name"
      value: "Loopback111"
    }
  }
}
```

gNMI のワイルドカード

gNMI プロトコルは、Get パスのワイルドカードをサポートしています。これは、複数の要素を一致させるためにパス内でワイルドカードを使用する機能です。これらのワイルドカードは、スキーマ内の指定されたサブツリーにあるすべての要素を示します。

elem は要素であり、xPath 内の / 文字の間の値です。elem は gNMI パスでも使用できます。たとえば、 elem 名を基準とするワイルドカードの位置は、ワイルドカードがインターフェイスを表し、すべてのインターフェイスとして解釈されることを暗に意味します。

ワイルドカードには暗黙的と明示的の2つのタイプがあり、どちらもサポートされています。Get パスは、パス ワイルドカードのすべてのタイプと組み合わせをサポートします。

- 暗黙的なワイルドカード：これらは、要素ツリー内の要素のリストを展開します。暗黙的なワイルドカードは、リストの要素にキー値が指定されていない場合に出現します。

  次に、パスの暗黙的なワイルドカードの例を示します。このワイルドカードは、デバイスにあるすべてのインターフェイスの説明を返します。

```
path {
  elem {
    name: "interfaces"
  }
  elem {
    name: "interface"
  }
  elem {
    name: "config"
  }
  elem {
    name: "description"
  }
}
```

- 明示的なワイルドカード：下記の指定によって同じ機能を提供します。
  - パス要素名またはキー名のいずれかにアスタリスク（*）を指定します。

    次に、パスのアスタリスクワイルドカードをキー名として使用する例を示します。このワイルドカードは、デバイスにあるすべてのインターフェイスの説明を返します。

```
path {
  elem {
    name: "interfaces"
  }
  elem {
    name: "interface"
    key {
     key: "name"
      value: "*"
    }
  }
  elem {
    name: "config"
  }
  elem {
    name: "description"
  }
}
```

    次に、パスのアスタリスク ワイルドカードをパス名として使用する例を示します。このワイルドカードは、Loopback111 インターフェイスで使用可能なすべての要素の説明を返します。

```
path {
  elem {
    name: "interfaces"
```

```
  }
  elem {
    name: "interface"
    key {
      key: "name"
      value: "Loopback111"
    }
  }
   elem {
    name: "*"
  }
  elem {
    name: "description"
  }
}
```

- 要素名として省略記号（...）または空のエントリを指定します。これらのワイルドカードは、パス内の複数の要素に展開できます。

  次に、パスの省略記号ワイルドカードの例を示します。このワイルドカードは、/interfaces 配下で使用可能なすべての説明フィールドを返します。

```
path {
  elem {
    name: "interfaces"
  }
  elem {
    name: "..."
  }
  elem {
    name: "description"
  }
}
```

次に、暗黙的なワイルドカードを使用した GetRequest の例を示します。この GetRequest は、デバイスにあるすべてのインターフェイスの oper-status を返します。

```
path {
  elem {
    name: "interfaces"
    }
  elem {
    name: "interface"
  }
  elem {
    name: "state"
  }
  elem {
    name: "oper-status"
  }
},
type: 0,
encoding: 4
```

次に、暗黙的なワイルドカードを使用した GetResponse の例を示します。

```
notification {
  timestamp: 1520627877608777450
  update {
```

```
    path {
      elem {
        name: "interfaces"
      }
      elem {
        name: "interface"
        key {
          key: "name"
          value: "\"FortyGigabitEthernet1/1/1\""
        }
      }
      elem {
        name: "state"
}
      elem {
        name: "oper-status"
      }
    }
    val {
      json_ietf_val: "\"LOWER_LAYER_DOWN\""
    }
  },

<snip>
…
</snip>

update {
    path {
      elem {
        name: "interfaces"
      }
      elem {
        name: "interface"
        key {
          key: "name"
          value: "\"Vlan1\""
        }
      }
      elem {
        name: "state"
      }
      elem {
        name: "oper-status"
      }
    }
    val {
      json_ietf_val: "\"DOWN\""
    }
  }
}
```

gNMI 設定の永続化

gNMI 設定の永続化機能により、gNMI SetRequest RPC によって行われたすべての正常な設定変更が、デバイスの再起動後も設定に保持されるようになります。この機能が導入される前は、gNMI 設定はデバイスの実行コンフィギュレーションに保存されていました。また、変更は **write memory** コマンドまたは SaveConfig NETCONF RPC の発行によって保存されていました。

実行コンフィギュレーションのすべての変更は、gNMI 以外の処理によって変更されたデータであっても、SetRequest RPC が発行されるとスタートアップ コンフィギュレーションにデータが保存されます。

この機能はデフォルトで有効であり、無効にすることはできません。

gNMI ユーザ名とパスワードによる認証

ユーザログイン情報、ユーザ名、およびパスワードは、各 gNMI RPC でメタデータとして承認を提供します。次に、ユーザ名とパスワードを使用するサンプル gNMI 機能 RPC を示します。

```
metadata = [('username','admin'), ('password','lab')]
cap_request = gnmi_pb2.CapabilityRequest()
# pass metadata to the gnmi_pb2_grpc.gNMIStub object
secure_stub.Capabilities(cap_request, metadata=metadata)
```

gNMI のエラー メッセージ

エラーが発生すると、gNMI は説明的なエラー メッセージを返します。次のセクションでは gNMI エラー メッセージをいくつか示します。

次に、パスが無効な場合に表示されるエラー メッセージの例を示します。

```
gNMI Error Response:
  <_Rendezvous of RPC that terminated with (StatusCode.TERMINATED,
   An error occurred while parsing provided xpath: unknown tag:
   "someinvalidxpath" Additional information: badly formatted or nonexistent path)>
```

次に、非実装エラーが発生した場合に表示されるエラー メッセージの例を示します。

```
gNMI Error Response:
  <_Rendezvous of RPC that terminated with (StatusCode.UNIMPLEMENTED,
    Requested encoding "ASCII" not supported)>
```

次に、データ要素が空の場合に表示されるエラー メッセージの例を示します。

```
gNMI Error Response:
  <_Rendezvous of RPC that terminated with (StatusCode.NOT_FOUND,
   Empty set returned for path "/oc-if:interfaces/noinfohere")>
```

gNMI プロトコルを有効にする方法

gNMI プロトコル を有効にするには、次の手順を実行します。

1. gNMI クライアントと、認証局（CA）によって署名されたデバイス用に一連の証明書を作成します。

1. Linux で OpenSSL を使用して証明書を作成します。
2. デバイスに証明書をインストールします。
3. デバイスで gNMI を設定します。
4. gNMI が有効になっていて実行されているかどうかを確認します。

2. 前の手順で設定したクライアント証明書とルート証明書を使用して gNMI クライアントを接続します。

Linux での OpenSSL を使用した証明書の作成

証明書とトラストポイントは、セキュア gNMI サーバにのみ必要です。

次に、Linux マシン上で OpenSSL を使用して証明書を作成する例を示します。

```
# Setting up a CA
openssl genrsa -out rootCA.key 2048
openssl req -subj /C=/ST=/L=/O=/CN=rootCA -x509 -new -nodes -key rootCA.key -sha256 -out
 rootCA.pem

# Setting up device cert and key
openssl genrsa -out device.key 2048
openssl req -subj /C=/ST=/L=/O=/CN=<hostnameFQDN> -new -key device.key -out device.csr
openssl x509 -req -in device.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out
device.crt -sha256
# Encrpyt device key - needed for input to IOS
openssl rsa -des3 -in device.key -out device.des3.key -passout pass:<password - remember
 this for later>

# Setting up client cert and key
openssl genrsa -out client.key 2048
openssl req -subj /C=/ST=/L=/O=/CN=gnmi_client -new -key client.key -out client.csr
openssl x509 -req -in client.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out
client.crt -sha256
```

CLI によるデバイスへの証明書のインストール

次の例は、デバイスに証明書をインストールする方法を示しています。

```
# Send:
Device# configure terminal
Device(config)# crypto pki import trustpoint1 pem terminal password password1

# Receive:
% Enter PEM-formatted CA certificate.
% End with a blank line or "quit" on a line by itself.

# Send:
# Contents of rootCA.pem, followed by newline + 'quit' + newline:
-----BEGIN CERTIFICATE-----
<snip>
-----END CERTIFICATE-----
quit
```

```
# Receive:
% Enter PEM-formatted encrypted private General Purpose key.
% End with "quit" on a line by itself.

# Send:
# Contents of device.des3.key, followed by newline + 'quit' + newline:
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,D954FF9E43F1BA20
<snip>
-----END RSA PRIVATE KEY-----
quit

# Receive:
% Enter PEM-formatted General Purpose certificate.
% End with a blank line or "quit" on a line by itself.

# Send:
# Contents of device.crt, followed by newline + 'quit' + newline:
-----BEGIN CERTIFICATE-----
<snip>
-----END CERTIFICATE-----
quit

# Receive:
% PEM files import succeeded.
Device(config)#

# Send:
Device(config)# crypto pki trustpoint trustpoint1
Device(ca-trustpoint)# revocation-check none
Device(ca-trustpoint)# end
Device#
```

非セキュア モードでの gNMI の有効化

（注） このタスクは、Cisco IOS XE Amsterdam 17.3.1 以降のリリースに適用されます。

[Day Zero setup] で、最初にデバイスを非セキュア モードで有効にしてから、デバイスを無効にし、セキュア モードを有効にします。インセキュアモードで gNxI を停止するには、**no gnxi server** コマンドを使用します。

（注） gNxI 非セキュアサーバとセキュアサーバはデバイス上で同時に実行できます。

（注） **gnxi** コマンドは、gNMI および gRPC ネットワーク操作インターフェイス（gNOI）サービスの両方に適用されます。gNxI ツールは、gNMI および gNOI プロトコルを使用するネットワーク管理用ツールのコレクションです。

手順の概要

1. **enable**
2. **configure terminal**
3. **gnxi**
4. **gnxi server**
5. **gnxi port** *port-number*
6. **end**
7. **show gnxi state**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**gnxi** 例： `Device(config)# gnxi`	gNxl プロセスを起動します。
ステップ 4	**gnxi server** 例： `Device(config)# gnxi server`	gNxI サーバを非セキュアモードで有効にします。
ステップ 5	**gnxi port** *port-number* 例： `(Optional) Device(config)# gnxi port 50000`	リッスンする gNxI ポートを設定します。 • デフォルトの非セキュア gNxI ポートは 50052 です。
ステップ 6	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 7	**show gnxi state** 例： `Device# show gnxi state`	gNxI インターフェイスのステータスが表示されます。

セキュア モードでの gNMI の有効化

（注） このタスクは、Cisco IOS XE Amsterdam 17.3.1 以降のリリースに適用されます。

セキュアモードで gNxI を停止するには、**no gnxi secure-server** コマンドを使用します。

（注） gNxI 非セキュアサーバとセキュアサーバはデバイス上で同時に実行できます。

（注） **gnxi** コマンドは、gNMI および gRPC ネットワーク操作インターフェイス（gNOI）サービスの両方に適用されます。gNxI ツールは、gNMI および gNOI プロトコルを使用するネットワーク管理用ツールのコレクションです。

手順の概要

1. **enable**
2. **configure terminal**
3. **gnxi**
4. **gnxi secure-server**
5. **gnxi secure-trustpoint** *trustpoint-name*
6. **gnxi secure-client-auth**
7. **gnxi secure-port**
8. **end**
9. **show gnxi state**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**gnxi** 例： `Device(config)# gnxi`	gNxl プロセスを起動します。

	コマンドまたはアクション	目的
ステップ 4	**gnxi secure-server** 例： `Device(config)# gnxi secure-server`	gNxI サーバをセキュアモードで有効にします。
ステップ 5	**gnxi secure-trustpoint** *trustpoint-name* 例： `Device(config)# gnxi secure-trustpoint trustpoint1`	gNxI が認証に使用するトラストポイントと証明書セットを指定します。
ステップ 6	**gnxi secure-client-auth** 例： `Device(config)# gnxi secure-client-auth`	（任意）gNxIプロセスは、ルート証明書と照合してクライアント証明書を認証します。
ステップ 7	**gnxi secure-port** 例： `Device(config)# gnxi secure-port`	（任意）リッスンする gNxI ポートを設定します。 • デフォルトのセキュア gNxI ポートは9339です。
ステップ 8	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 9	**show gnxi state** 例： `Device# show gnxi state`	gNxI サーバのステータスを表示します。

例

次に、**show gnxi state** コマンドの出力例を示します。

```
Device# show gnxi state

State          Status
--------------------------------
Enabled        Up
```

gNMI クライアントの接続

以前に設定したクライアント証明書とルート証明書を使用して gNMI クライアントが接続されます。

次に、Python を使用して gNMI クライアントを接続する例を示します。

```
# gRPC Must be compiled in local dir under path below:
>>> import sys
```

```
>>> sys.path.insert(0, "reference/rpc/gnmi/")
>>> import grpc
>>> import gnmi_pb2
>>> import gnmi_pb2_grpc
>>> gnmi_dir = '/path/to/where/openssl/creds/were/generated/'

# Certs must be read in as bytes
>>> with open(gnmi_dir + 'rootCA.pem', 'rb') as f:
>>>     ca_cert = f.read()
>>> with open(gnmi_dir + 'client.crt', 'rb') as f:
>>>     client_cert = f.read()
>>> with open(gnmi_dir + 'client.key', 'rb') as f:
>>>     client_key = f.read()

# Create credentials object
>>> credentials = grpc.ssl_channel_credentials(root_certificates=ca_cert,
private_key=client_key, certificate_chain=client_cert)

# Create a secure channel:
# Default port is 9339, can be changed on ios device with 'gnxi secure-port ####'
>>> port = 9339
>>> host = <HOSTNAME FQDN>
>>> secure_channel = grpc.secure_channel("%s:%d" % (host, port), credentials)

# Create secure stub:
>>> secure_stub = gnmi_pb2_grpc.gNMIStub(secure_channel)

# Done! Let's test to make sure it works:
>>> secure_stub.Capabilities(gnmi_pb2.CapabilityRequest())
supported_models {
<snip>
}
supported_encodings: <snip>
gNMI_version: "0.4.0"
```

gNMI プロトコルの設定例

例：非セキュア モードでの gNMI の有効化

（注） この例は Cisco IOS XE Amsterdam 17.3.1 以降のリリースに適用されます。

次に、gNxI サーバを非セキュアモードで有効にする例を示します。

```
Device> enable
Device# configure terminal
Device(config)# gnxi
Device(config)# gnxi server
Device(config)# gnxi port 50000 <The default port is 50052.>
Device(config)# end
Device#
```

例：セキュア モードでの gNMI の有効化

（注） この例は Cisco IOS XE Amsterdam 17.3.1 以降のリリースに適用されます。

次に、gNxI サーバをセキュアモードで有効にする例を示します。

```
Device> enable
Device# configure terminal
Device(config)# gnxi
Device(config)# gnxi server
Device(config)# gnxi secure-server
Device(config)# gnxi secure-trustpoint trustpoint1
Device(config)# gnxi secure-client-auth
Device(config)# gnxi secure-port 50001 <The default port is 9339.>
Device(config)# end
Device#
```

gNMI プロトコルの関連資料

関連資料

関連項目	マニュアル タイトル
DevNet	https://developer.cisco.com/site/ios-xe/
gNMI	https://github.com/openconfig/reference/blob/master/rpc/gnmi/gnmi-specification.md
gNMI パス エンコーディング	https://github.com/openconfig/reference/blob/master/rpc/gnmi/gnmi-path-conventions.md

標準および RFC

標準/RFC	タイトル
RFC 7951	YANG でモデル化されたデータの JSON エンコーディング

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

gNMI プロトコルの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

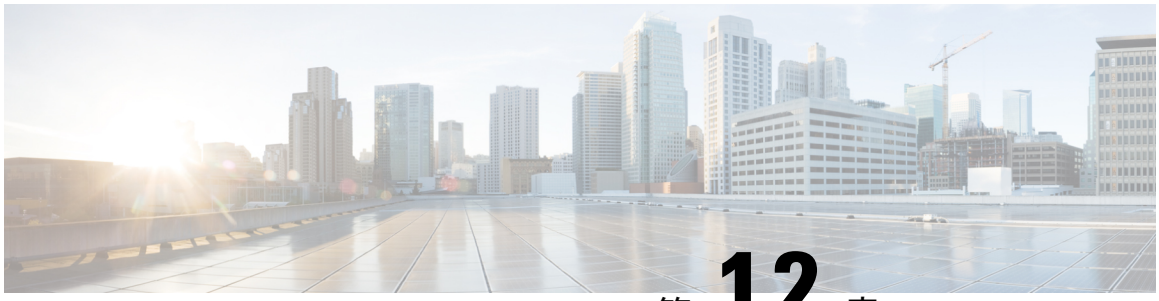
表 *18 : gNMI* プロトコルの機能情報

機能名	リリース	機能情報
gNMI プロトコル	Cisco IOS XE Fuji 16.8.1a	この機能では、gNMI の機能と GET および SET RPC を使用したモデル駆動型の設定と運用データの取得について説明します。 この機能は、次のプラットフォームに実装されていました。 • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ
	Cisco IOS XE Gibraltar 16.10.1	Cisco IOS XE Gibraltar 16.10.1 では、この機能は Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチに実装されていました。
	Cisco IOS XE Gibraltar 16.11.1	Cisco IOS XE Gibraltar 16.11.1 では、この機能は Cisco Catalyst 9600 シリーズ スイッチに実装されていました。
	Cisco IOS XE Gibraltar 16.12.1	この機能は、Cisco IOS XE Gibraltar 16.12.1 で次のプラットフォームに実装されました。 • Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300L SKU • Cisco cBR-8 コンバージドブロードバンド ルータ
	Cisco IOS XE Amsterdam 17.1.1	

機能名	リリース	機能情報
		この機能は、Cisco IOS XE Amsterdam 17.1.1で次のプラットフォームに実装されました。 • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco Network Convergence System 520 シリーズ • Cisco Network Convergence System 4200 シリーズ
	Cisco IOS XE Amsterdam 17.2.1r	この機能は、Cisco IOS XE Amsterdam 17.2.1r で、Cisco ASR 1000 シリーズ アグリゲーション サービス ルータに実装されました。
gNMIユーザ名とパスワードによる認証	Cisco IOS XE Gibraltar 16.12.1	ユーザ名とパスワードによる認証機能がgNMIプロトコルに追加されました。この機能は、gNMIをサポートするすべてのIOS XEプラットフォームでサポートされます。

機能名	リリース	機能情報
gNMI 設定の永続化	Cisco IOS XE Amsterdam 17.3.1	gNMI SetRequest RPCを介して行われたすべての正常な設定変更は、デバイスの再起動後も保持されます。この機能は、次のプラットフォームに実装されていました。 • Cisco 1000 シリーズ サービス統合型ルータ • Cisco 4000 シリーズ サービス統合型ルータ • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ
gNOI 証明書の管理	Cisco IOS XE Amsterdam 17.3.1	gNOI証明書の管理サービスは、RPCを提供して、インストール、ローテーション、証明書の取得、証明書の失効、および証明書署名要求の生成を行います。この機能は、次のプラットフォームに実装されていました。 • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ

機能名	リリース	機能情報
証明書サービスによる gNOI ブートストラップ	Cisco IOS XE Amsterdam 17.3.1	gNOI 証明書をインストールした後、ブートストラップを使用してターゲットデバイスを設定または操作します。gNMI ブートストラップは、**gnxi-secure-int** コマンドで有効、**secure-allow-self-signed-trustpoint** コマンドで無効になります。 この機能は、次のプラットフォームに実装されていました。 • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ



第 12 章

gRPC ネットワーク操作インターフェイス

Google リモートプロシージャコール（gRPC） ネットワーク操作インターフェイス（gNOI）は一連のマイクロサービスであり、それぞれが一連の操作に対応しています。このモジュールでは、サポートされている gNOI サービスについて説明します。

gRPC ネットワーク操作インターフェイスに関する情報

gNOI プロトコル

gNOI は、ネットワークデバイス上で操作コマンドを実行するための gRPC ベースのマイクロサービスセットを定義します。gNMIサービスは、設定管理、動作状態の取得、およびストリーミングテレメトリによるバルクデータ収集の動作を定義します。 gNOI では、デバイスがサポートするサービスのみを採用できます。gNOIは、OSインストールサービスをサポートします。

gNOI は、ユーザ認証の有無にかかわらず使用できます。ユーザ認証はデフォルトでディセーブルになっています。**gnxi secure-password-auth** コマンドを使用してユーザ認証を有効にします。OpenConfig モデルによるユーザー認証の有効化については、https://github.com/YangModels/yang/blob/master/vendor/cisco/xe/1751/openconfig-system-management.yangを参照してください。

gNOI プロトコルは、次の操作をサポートします。

- 証明書の管理
- ブートストラップ
- OS インストールサービス

証明書管理サービス

証明書管理サービスでは、初めに 2 つの主要な RPC、Install と Rotate がエクスポートされます。これらの RPC はそれぞれ新しい証明書のインストールとデバイス上の既存の証明書のローテーションに使用されます。

証明書管理サービスでは、次の RPC がサポートされています。

- Install：証明書をインストールします。すべての証明書は、証明書 ID によって一意に識別されます。証明書 ID は文字列です。
- Rotate：既存の証明書をローテーションします。
- RevokeCertificates：1 つ以上の証明書を取り消します。
- GetCertificates：すべての証明書を照会します。
- CanGenerateCSR：デバイスが証明書署名要求（CSR）を生成できるかどうかを照会します。

前述の RPC によって作成されたトラストポイントと証明書は、スイッチオーバーおよびデバイスのリブート後も保持されます。

次に、証明書管理サービスの定義の例を示します。

```
service CertificateManagement {
 rpc Install(stream InstallCertificateRequest)
  returns (stream InstallCertificateResponse);

 rpc Rotate(stream RotateCertificateRequest)
  returns (stream RotateCertificateResponse);

 rpc RevokeCertificates(RevokeCertificateRequest)
  returns (RevokeCertificateResponse);

 rpc GetCertificates(GetCertificateRequest)
  returns (GetCertificateResponse);

 rpc CanGenerateCSR(CanGenerateCSRRequest)
  returns (CanGenerateCSRResponse);
}
```

Install RPC

Install RPC は、新しい CSR 要求を作成して、新しい証明書をデバイスに追加します。新しい証明書は、デバイスの新しい証明書 ID に関連付けられます。デバイスに指定された証明書 ID を持つ既存の証明書がある場合、操作は失敗します。

Install RPC は、双方向ストリーミング RPC です。入力（InstallCertificateRequest）と出力（IntsallCertificateResponse）があり、どちらもストリーミングです。ストリームが中断されるか、プロセスのいずれかのステップが失敗すると、デバイスは変更をロールバックします。

次に、Install RPC の定義とメッセージの例を示します。

```
rpc Install(stream InstallCertificateRequest)
returns (stream InstallCertificateResponse);

// Request messages to install new certificates on the target.
message InstallCertificateRequest {
  // Request Messages.
  oneof install_request {
    GenerateCSRRequest generate_csr = 1;
    LoadCertificateRequest load_certificate = 2;
  }
}
// Request to generate the CSR.
message GenerateCSRRequest {
  // Parameters for creating a CSR.
  CSRParams csr_params = 1;
  // The certificate id with which this CSR will be associated. The target
  // configuration should bind an entity which wants to use a certificate to
  // the certificate_id it should use.
  string certificate_id = 2;
}
// Parameters to be used when generating a Certificate Signing Request.
message CSRParams {
  // The type of certificate which will be associated for this CSR.
  CertificateType type = 1;

  // Minimum size of the key to be used by the target when generating a
  // public/private key pair.
  uint32 min_key_size = 2;

  // If provided, the target must use the provided key type. If the target
  // cannot use the algorithm specified in the key_type, it should cancel the
  // stream with an Unimplemented error.
  KeyType key_type = 3;

  // --- common set of parameters applicable for any type of certificate --- //
  string common_name = 4;           // e.g "device.corp.google.com"
  string country = 5;               // e.g "US"
  string state = 6;                 // e.g "CA"
  string city = 7;                  // e.g "Mountain View"
  string organization = 8;          // e.g "Google"
  string organizational_unit = 9;   // e.g "Security"
  string ip_address = 10;
  string email_id = 11;
}
// A certificate.
message Certificate {
  // Type of certificate.
  CertificateType type = 1;

  // Actual certificate.
  // The exact encoding depends upon the type of certificate.
  // for X509, this should be a PEM encoded Certificate.
  bytes certificate = 2;
}

message LoadCertificateRequest {
  // The certificate to be Loaded on the target.
  Certificate certificate = 1;

  // The key pair to be used with the certificate. This is provided in the event
  // that the target cannot generate a CSR (and the corresponding public/private
  // keys).
  KeyPair key_pair = 2;
```

```
  // Certificate Id of the above certificate. This is to be provided only when
  // there is an externally generated key pair.
  string certificate_id = 3;

  // Optional pool of CA certificates to be used for authenticating the client.
  repeated Certificate ca_certificate = 4;
}

// A message representing a pair of public/private keys.
message KeyPair {
  bytes private_key = 1;
  bytes public_key = 2;
}

// Response Messages from the target for the InstallCertificateRequest.
message InstallCertificateResponse {
  // Response messages.
  oneof install_response {
    GenerateCSRResponse generated_csr = 1;
    LoadCertificateResponse load_certificate = 2;
  }
}

// GenerateCSRResponse contains the CSR associated with the Certificate ID
// supplied in the GenerateCSRRequest. When a Certificate is subsequently
// installed on the target in the same streaming RPC session, it must be
// associated to that Certificate ID.
//
// An Unimplemented error will be returned if the target cannot generate a CSR
// as per the request. In this case, the caller must generate its own key pair.
message GenerateCSRResponse {
  CSR csr = 1;
}

// A Certificate Signing Request.
message CSR {
  // Type of certificate.
  CertificateType type = 1;

  // Bytes representing the CSR.
  // The exact encoding depends upon the type of certificate requested.
  // for X509: This should be the PEM encoded CSR.
  bytes csr = 2;
}
```

ターゲットデバイスが起動し、gNOI がデフォルト状態になると、コントローラ（サードパーティの実装）は Install RPC を使用して、認証局（CA）によって署名された証明書をインストールします。証明書は、証明書 ID によって一意に識別されます。この ID は、公開キーインフラストラクチャ（PKI）設定でトラストポイント名として使用されます。既存の証明書 ID を持つ証明書をインストールしようとすると、インストールは失敗します。

次のセクションでは、デバイスによって CSR が生成される方法について説明します。

1. デバイスは、Install RPC を使用して自己署名証明書を生成します。暗号化モード（または gNMI のデフォルト状態）では、コントローラはターゲットデバイスによって提示された証明書を検証しないため、コントローラはこの証明書のコピーを必要としません。これは、デフォルトの状態です。
2. コントローラはデバイスに CSR の生成を要求し、CSR を CA に送信し、CA から署名証明書を取得します。

3. 署名証明書は、証明書の署名に使用される CA 証明書とともにデバイスにインストールされます。CA 証明書は ca_certificates バンドルに存在し、デバイス証明書をインストールするために PKI が要求します。
4. gNMI または gNOI サービスは、プロビジョニングされた状態になった、新しくインストールされた証明書を使用して再起動します。

Rotate RPC

Rotate RPC により既存の証明書が更新されます。これはすでにインストールされている証明書です。証明書がまだインストールされていない場合、Rotate RPC は失敗します。使用されていない証明書はローテーションできますが、クライアントはそれをテストできません。

次に、Rotate RPC の定義の例を示します。

```
rpc Rotate(stream RotateCertificateRequest)
returns (stream RotateCertificateResponse);

// Request messages to rotate existing certificates on the target.
message RotateCertificateRequest {
  // Request Messages.
  oneof rotate_request {
    GenerateCSRRequest generate_csr = 1;
    LoadCertificateRequest load_certificate = 2;
    FinalizeRequest finalize_rotation = 3;
  }
}

// A Finalize message is sent to the target to confirm the Rotation of
// the certificate and that the certificate should not be rolled back when
// the RPC concludes. The certificate must be rolled back if the target returns
// an error after receiving a Finalize message.
message FinalizeRequest {
}

message RotateCertificateResponse {
  // Response messages.
  oneof rotate_response {
    GenerateCSRResponse generated_csr = 1;
    LoadCertificateResponse load_certificate = 2;
  }
}
```

Rotate RPC は、次の点で Install RPC と異なります。

- PKI は（ロールバックの目的で）新しい証明書をインストールするときに、古い証明書と CA 証明書を保存またはキャッシュする必要があります。
- コントローラは新しい接続を作成し、更新された証明書が機能するかどうかをテストし、成功した場合は証明書のローテーションを完了します。

Revoke RPC

この RPC は、証明書 ID によって一意に識別される 1 つ以上の証明書を失効させるために使用されます。証明書を失効させると、対応するトラストポイントが Cisco IOS XE の設定から削除されます。対応するトラストポイントが現在使用されている場合、またはトラストポイントが存在しない場合は、証明書の失効が失敗する可能性があります。

RevokeCertificate RPC では、証明書の失効が成功する場合も失敗する場合もあります。ターゲットデバイスでは、失効は単純な削除操作です。 CA による実際の失効はクライアントによって行われます。クライアントが使用中の証明書を失効させた場合、新しい接続は失敗しますが、既存の接続は影響を受けません。

次に、RevokeCertificate RPC の例を示します。

```
// An RPC to revoke specific certificates.
// If a certificate is not present on the target, the request should silently
// succeed. Revoking a certificate should render the existing certificate
// unusable by any endpoints.
rpc RevokeCertificates(RevokeCertificatesRequest)
returns (RevokeCertificatesResponse);

message RevokeCertificatesRequest {
  // Certificates to revoke.
  repeated string certificate_id = 1;
}

message RevokeCertificatesResponse {
  // List of certificates successfully revoked.
  repeated string revoked_certificate_id = 1;

  // List of errors why certain certificates could not be revoked.
  repeated CertificateRevocationError certificate_revocation_error = 2;
}

// An error message indicating why a certificate id could not be revoked.
message CertificateRevocationError {
  string certificate_id = 1;
  string error_message = 2;
}
```

GetCertificate RPC

この RPC はすべての証明書 ID を照会します。

クエリに対する応答には、次の情報が含まれます。

- 証明書 ID で識別されるすべての証明書の証明書情報。
- この証明書を使用するエンドポイント（トンネル、デーモンなど）のリスト。

（注） サポートされないエンドポイント。

（注） 応答には ca_certificate バンドルは含まれません。

次に、GetCertificate RPC の例を示します。

```
// An RPC to get the certificates on the target.
rpc GetCertificates(GetCertificatesRequest) returns (GetCertificatesResponse);

// The request to query all the certificates on the target.
message GetCertificatesRequest {
}

// Response from the target about the certificates that exist on the target what
// what is using them.
message GetCertificatesResponse {
  repeated CertificateInfo certificate_info = 1;
}

message CertificateInfo {
  string certificate_id = 1;
  Certificate certificate = 2;

  // List of endpoints using this certificate.
  repeated Endpoint endpoints = 3;

  // System modification time when the certificate was installed/rotated in
  // nanoseconds since epoch.
  int64 modification_time = 4;
}

// An endpoint represents an entity on the target which can use a certificate.
message Endpoint {
  // Type of endpoint that can use a cert. This list is to be extended based on
  // conversation with vendors.
  enum Type {
    EP_UNSPECIFIED = 0;
    EP_IPSEC_TUNNEL = 1;
    EP_DAEMON = 2;
  }
  Type type = 1;

  // Human readable identifier for an endpoint.
  string endpoint = 2;
}
```

CanGenerateCSR RPC

この RPC は、デバイスが特定のキータイプ、証明書タイプ、およびキーサイズの CSR を生成できるかどうかを照会します。サポートされるキータイプは、Rivest、Shamir、およびAdelman（RSA）で、サポートされる証明書タイプは X.509 です。

この RPC 要求が Install RPC の一部として完全に新しい証明書をインストールするために作成されている場合、証明書 ID が新しいものであり、デバイス上のエンティティがこの証明書 ID にバインドされていないことをデバイスで確認する必要があります。既存の証明書が証明書 ID と一致する場合、この要求は失敗します。

この RPC 要求が、Rotate RPC の一部として既存の証明書をローテーションするように作成された場合、証明書 ID がすでに使用可能であることをデバイスで確認する必要があります。証明書のローテーションで証明書のロードを続行する場合は、新しい証明書を以前に作成した証明書 ID に関連付ける必要があります。

次に、CanGenerateCSR RPC の例を示します。

```
// An RPC to ask a target if it can generate a Certificate.
rpc CanGenerateCSR(CanGenerateCSRRequest) returns (CanGenerateCSRResponse);

// A request to ask the target if it can generate key pairs.
message CanGenerateCSRRequest {
  KeyType key_type = 1;
  CertificateType certificate_type = 2;
  uint32 key_size = 3;
}

// Algorithm to be used for generation the key pair.
enum KeyType {
  // 1 - 500, for known types.
  // 501 and onwards for private use.
  KT_UNKNOWN = 0;
  KT_RSA = 1;
}

// Types of certificates.
enum CertificateType {
  // 1 - 500 for public use.
  // 501 onwards for private use.
  CT_UNKNOWN = 0;
  CT_X509 = 1;
}

// Response from the target about whether it can generate a CSR with the given
// parameters.
message CanGenerateCSRResponse {
  bool can_generate = 4;
}
```

相互認証

相互認証は双方向認証です。2つのパーティが同時に相互に認証します。相互認証を有効にするには、**gnmi-yang secure-peer-verify-trustpoint** コマンドを使用します。このコマンドが有効になっていない場合、認証サービスが gNMI クライアントをすべての既存のトラスポイントおよびトラストプールの内容に対して検証します。

相互認証のために CA 証明書をローテーションするには、クライアントがターゲットデバイスに新しいバンドルを提示し、古いバンドルを削除する必要があります。ただし、CA 証明書はトラストプールに存在しており、トラストプールから選択的に削除することはできません。

証明書サービスによるブートストラップ

gNOI 証明書をインストールした後、ブートストラップを使用してターゲットデバイスを設定または操作します。ターゲットデバイスに既存の証明書がない場合、gNOI 証明書管理サービスを使用してブートストラップにより証明書をインストールできます。証明書のインストール後、デバイスはセキュアな gNOI 接続または gNMI 接続を確立できます。このプロセスは、既存のセキュアな環境を前提としています。

gNMI ブートストラップを有効にするには、**gnxi secure-int** コマンドを使用します。

（注） gNOI 証明書管理サービスは、ブートストラップの前にインストールする必要があります。

gNOI 証明書管理サービスには 2 種類の状態があります。これらの状態は、gNOI サービスと gNMI サービスの両方でサポートされます。

- Default/Encrypted：デバイス上の gNOI と gNMI は、クライアントが検証しない自己署名（デフォルト）証明書を使用します。証明書は認証を必要としません。この状態では、gNOI 証明書サービスのみがターゲットデバイスで有効になります。
- Provisioned：デバイス上の gNOI および gNMI は、クライアントによって検証されたインストール済み証明書を使用します。クライアントはその証明書を提示し、デバイスは証明書ストアと照合して証明書を検証します。デバイスは、相互認証が有効になっている場合にのみクライアント証明書を検証します。

OS インストールサービス

OS インストールサービスは、インストールに使用される gNOI API を定義します。OS インストールサービスは、gNOI プロトコルでサポートされています。

このサービスは、OS をデバイスにインストールするためのインターフェイスを提供します。次の 3 つの RPC をサポートしています。

- Install：この RPC はイメージをデバイスに転送します。これらのイメージは、バージョン文字列によって一意に識別されます。この RPC は **install add** コマンドに似ています。主な違いは、イメージが RPC の一部として転送されることです。
- Activate：この RPC は、RPC への入力の一部である要求された OS バージョンを、次回の再起動時に使用されるバージョンとして設定し、デバイスを再起動します。この RPC は、**install activate** および **install commit** コマンドと同じです。
- Verify：この RPC は現在の OS バージョンを確認します。

Cisco IOS XE デバイスは、ソフトウェアイメージの起動で、インストールモードとバンドルモードの両方をサポートします。

インストールモードでは、flash: ファイルシステム内に存在するソフトウェアパッケージのプロビジョニングファイルを起動して、デバイスを起動できます。インストールされている各パッケージの ISO ファイルシステムは、フラッシュからルートファイルシステム（rootfs）に直接マウントされます。

バンドルモードでは、バンドル（.bin）ファイルを使用してデバイスを起動できます。パッケージはバンドルから取得され、RAM にコピーされます。各パッケージの ISO ファイルシステムは、rootfs にマウントされます。インストールモードでの起動とは異なり、バンドルモードでの起動では、バンドルのサイズに対応するサイズの追加メモリが使用されます。

次のシナリオでは、デバイスがバンドルモードで起動するとエラーメッセージが生成されます。

- デバイスが、バンドルモードで実行している現在のイメージで起動する。
- 新しいイメージをインストールするために、デバイスで Install RPC が開始される。

エラー メッセージの例を次に示します。

```
May 11 09:24:15.385 PST: %INSTALL-3-OPERATION_ERROR_MESSAGE:
Switch 1 R0/0: install_engine: Failed to install_add package
flash:gNOI_iosxe_17.05.01.0.144.1617180620.bin, Error: [2|install_add(ERR, )]:
Booted in bundle mode. For Bundle-to-Install mode conversion,
please use one-shot CLI - install add file <> activate commit
```

エラーメッセージが生成されても、Install RPC はクライアントに成功を返します。エラーメッセージは無視しても問題ありません。後続の Activate RPC は影響を受けません。新しいイメージで再起動すると、デバイスはインストールモードになります。

（注） このエラーメッセージは、デバイスが最初にインストールモードで実行していた場合は表示されません。これは、デバイスがバンドルモードで起動する場合にのみ該当します。

すべてのエラーメッセージを表示するには、https://github.com/openconfig/gnoi/blob/master/os/os.proto#L218 を参照してください。

インストールモードの詳細については、システム管理コンフィギュレーション ガイドの「デバイスのセットアップ設定の実行」の章を参照してください。

デュアルルートプロセッサのサポート

シスコのデバイスは、インサービス ソフトウェア アップデート（ISSU）（インストールモードのみサポート）と非 ISSU モードの両方をサポートします。ISSU がサポートされていない場合、または Install RPC を介して使用できない場合、gNOI OS インストールサービスは非 ISSU インストールを要求します。

デュアルルートプロセッサ（RP）の場合にデバイスが ISSU アップグレードをサポートする場合、gNOI OS インストール サービス インターフェイスは install activate ISSU ワークフローを呼び出します。ISSU がサポートされていない、またはデバイスが単一の RP をサポートしている、他のすべてのシナリオでは、gNOI OS インストールサービスは通常の非 ISSU イメージ インストール ワークフローを使用して gRPC アクティベート要求を処理します。

バンドルモードでは、**install add file** *filename* **activate commit** コマンドを使用してアップグレードが実行されます。このアップグレードは、単一の RP を持つデバイスの場合も同じです。ISSU がサポートされていないということは、両方の RP が同時にリロードされ、1 つの RP が起動するまでデバイスがダウンすることを意味します。

ISSU を使用しないインストールモードでは、両方の RP が同時にリロードされ、1 つの RP が起動するまでデバイスがダウンします。ISSU を使用したインストールモードでは、RP のリロードが同時に行われ、デバイスのダウンタイムが短くなります。

OS Install RPC

Install RPC は、イメージをデバイスに転送します。この RPC は、入力の InstallRequest RPC と出力の InstallResponse RPC で構成されます。どちらも双方向ストリーミング RPC です。

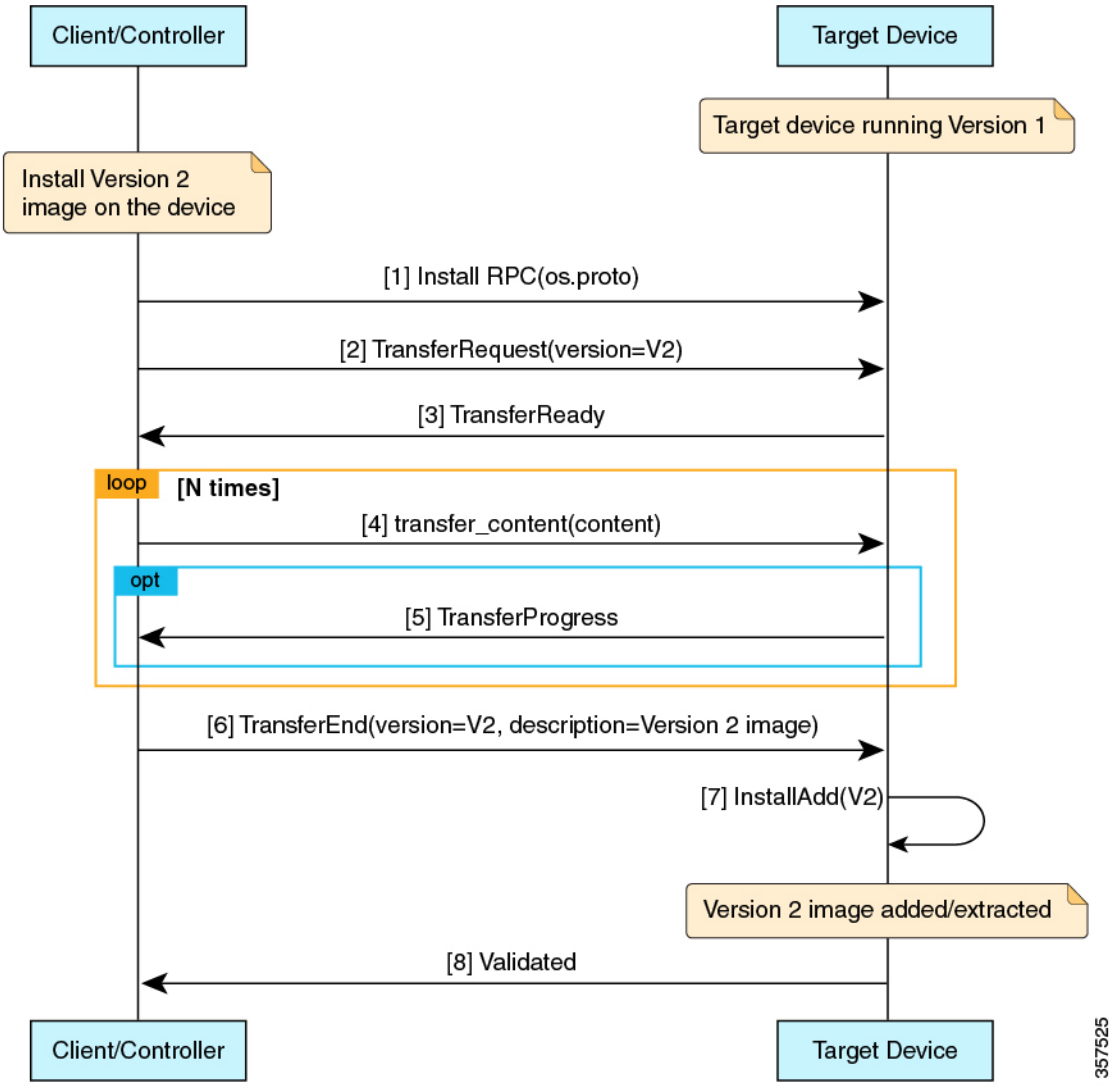
この RPC はソフトウェア メンテナンス アップデート（SMU）をサポートしていません。

次に、単一の RP がオペレーティング システム バージョン 1 を実行しているデバイスでの Install RPC のメッセージシーケンスの概要を示します。

1. クライアントがデバイスへの Install RPC を開始します。
2. クライアントは、バージョンをバージョン 2 に設定した TransferRequest メッセージをデバイスに送信します。
3. デバイスは、TransferReady メッセージでクライアントに応答します。これは、クライアントがイメージの転送を開始するために必要です。
4. クライアントは、複数の transfer_content メッセージをデバイスに送信して、イメージを転送します。
5. オプションで、デバイスはクライアントに TransferProgress メッセージを送信します。
6. クライアントは、イメージ転送が完了したことを示すTransferEnd メッセージをデバイスに送信します。
7. インストールモードでは、デバイスは **install add** コマンドと同等の操作をプログラムで実行します。パッケージの内容が抽出されます。
8. デバイスは、イメージから抽出したバージョンを含む Validated メッセージをクライアントに送信し、イメージ転送が有効であることを示します。

（注） クライアントによって Install RPC が途中で停止した場合、または操作の一部が失敗した場合は、ローカルイメージファイルが削除され、**install remove inactive** コマンドが自動的に呼び出されます。適切なステータスコードがクライアントに返されます。

図 *5*: 単一 *RP* のイメージ インストール ワークフロー



OS Activate RPC

Activate RPC は、要求されたオペレーティングシステムのバージョンを次回の再起動時に使用するバージョンとして設定し、ターゲットデバイスを再起動します。このRPCは、インストールされたオペレーティングシステムのバージョンをアクティブ化します。指定されたバージョンがまだインストールされていない場合、Activate RPC は失敗します。

クライアントは、Install RPC の Validated メッセージで受信したバージョンを提供する必要があります。

次に、単一の RP がオペレーティング システム バージョン 1 を実行しているデバイスでの Activate RPC のメッセージシーケンスを示します。

1. クライアントは、デバイスに対して Activate RPC を開始します。
2. クライアントは、デバイスにバージョン 2 の ActivateRequest メッセージを送信します。

   このメッセージシーケンスでは、バージョン 2 が Install RPC によってすでにインストールされているものとします。
3. デバイスは、インストールモードの場合は **install activate commit** コマンドに相当するプログラム操作を、バンドルモードの場合は **install add file activate commit** コマンドに相当するプログラム操作を実行します。
4. アクティブ化プロセスでエラーが検出されないため、デバイスはクライアントに ActivateResponse(OK) メッセージで応答します。
5. デバイスにバージョン 2 がリロードされます。
6. リロード後にデバイスが起動すると、**install commit** コマンドと同等のプログラム操作が実行されます。

（注） 1 つの非アクティブ イメージ バージョンのみがサポートされます。このため、クライアントがバージョン 2 をインストールしてからバージョン 3 をインストールすると、バージョン 2 のファイルが削除されます。

次の図は、イメージのアクティブ化ワークフローを示しています。

図 *6 :* シングル *RP* イメージのアクティブ化ワークフロー

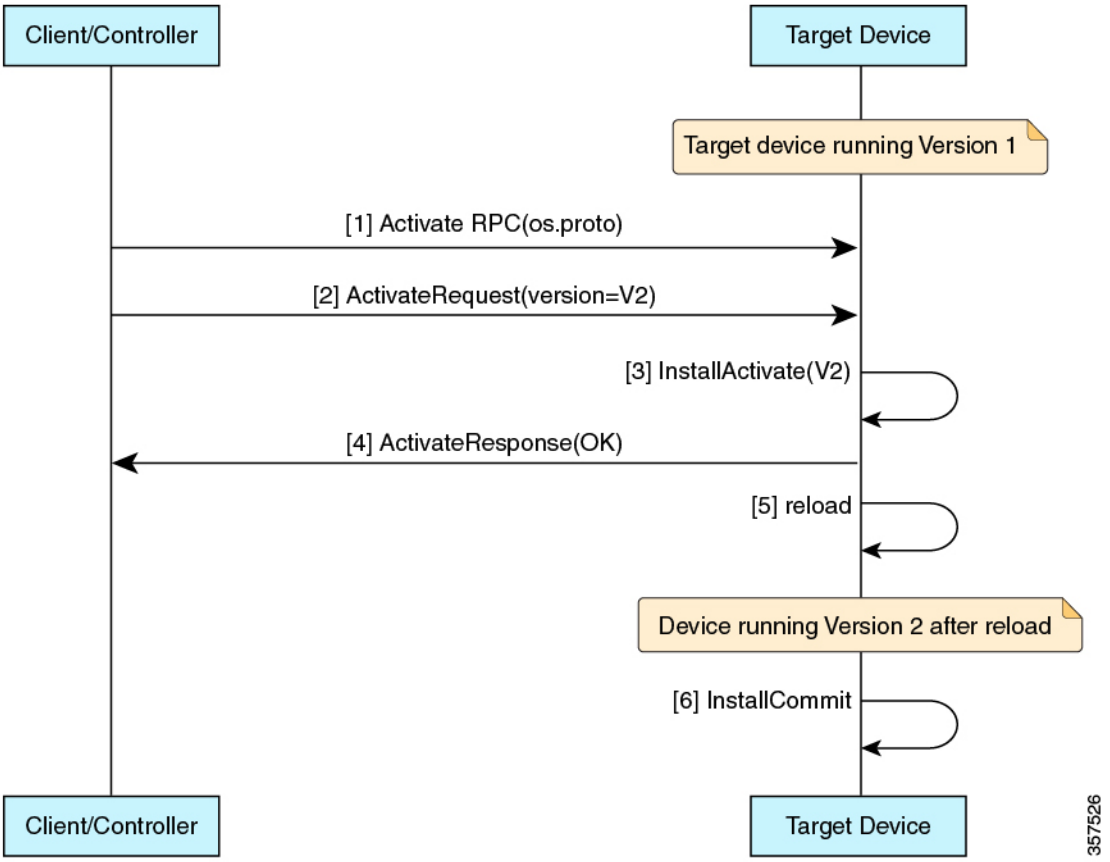


図 *7*: バンドルモードでのデュアル *RP* イメージインストール + 非 *ISSU* アクティブ化のワークフロー

図 *8 :* デュアル *RP* イメージインストール *+* 非 *ISSU* アクティブ化のンワークフロー

OS Verify RPC

Verify RPC は、実行中の OS バージョンを検証します。RPC への応答には、スタンバイ RP のサポートとプレゼンスに関する情報が含まれています。

最後の Activate RPCでエラーが発生した場合は、そのエラーが文字列として応答で返されます。gNOIOSインストールサービスは、インストール運用モデルとプラットフォームモデルを使用してこの情報を入力します。現在、インストール運用モデルは、2 つの RP で実行される異なるバージョンをサポートしていません。

gRPCネットワーク操作インターフェイスに関する追加情報

関連資料

関連項目	マニュアル タイトル
DevNet	https://developer.cisco.com/site/ios-xe/
gNOI	https://github.com/openconfig/gnoi
OS サービス	https://github.com/openconfig/gnoi/blob/master/os/os.proto
デバイスのセットアップ設定の実行	• システム管理コンフィギュレーション ガイド（Catalyst 9200 スイッチ） • システム管理コンフィギュレーション ガイド（Catalyst 9300 スイッチ） • システム管理コンフィギュレーション ガイド（Catalyst 9400 スイッチ） • システム管理コンフィギュレーション ガイド（Catalyst 9500 スイッチ） • システム管理コンフィギュレーション ガイド（Catalyst 9600 スイッチ）

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

gRPC ネットワーク操作インターフェイスの機能情報

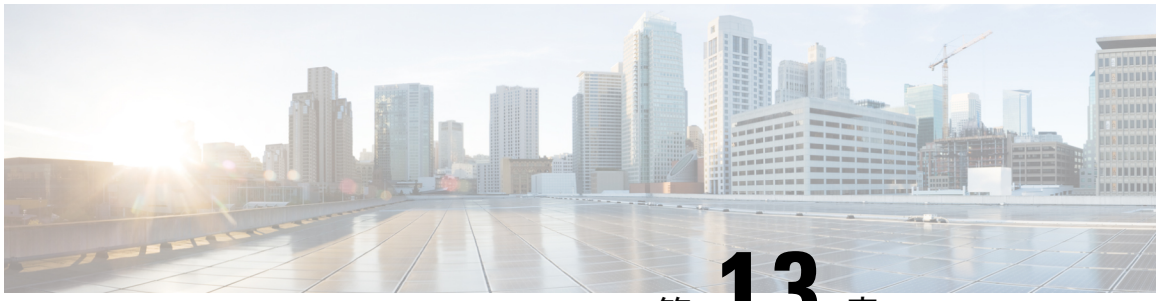
次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *19 : gRPC* ネットワーク操作インターフェイスの機能情報

機能名	リリース	機能情報
gNOI 証明書の管理	Cisco IOS XE Amsterdam 17.3.1	gNOI 証明書の管理サービスは、RPC を提供して、インストール、ローテーション、証明書の取得、証明書の失効、および証明書署名要求の生成を行います。 この機能は、Cisco IOS XE Amsterdam 17.3.1 で次のプラットフォームに実装されました。 • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ

機能名	リリース	機能情報
証明書サービスによる gNOI ブートストラップ	Cisco IOS XE Amsterdam 17.3.1	gNOI 証明書をインストールした後、ブートストラップを使用してターゲットデバイスを設定または操作します。gNMI ブートストラップは、 **gnxi secure-int** コマンドを使用して有効化できます。 この機能は、Cisco IOS XE Amsterdam 17.3.1 で次のプラットフォームに実装されました。 • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ
gNOI OS インストール サービス	Cisco IOS XE Bengaluru 17.5.1	gNOI OSインストールサービスは、インストールに使用される gNOI API を定義します。 この機能は、Cisco IOS XE Bengaluru 17.5.1で次のプラットフォームに実装されました。 • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 および 9500 ハイパフォーマンス シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ



第 13 章

モデルベースの AAA

NETCONF インターフェイスと RESTCONFイ ンターフェイスは、NETCONF アクセス制御モデル（NACM）を実装しています。NACM は、RFC 6536 で規定されたロールベース アクセス コントロール（RBAC）の形式の 1 つです。

モデルベースの AAA

モデルベースの AAA の前提条件

モデルベースの AAA 機能を使用するには、次の内容について事前に理解しておく必要があります。

- NETCONF-YANG
- NETCONF-YANG kill セッション
- RFC 6536：ネットワーク設定プロトコル（NETCONF）アクセス制御モデル

初期操作

NETCONF サービスや RESTCONF サービスが有効になると、/nacm サブツリーが事前に設定されていないデバイスは、特権レベル 15 のユーザ以外のすべての操作とデータへの読み取り/書き込み/実行アクセスを拒否します。これについては、/nacm サブツリーの次の設定に記述されています。

```
<nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
      <enable-nacm>true</enable-nacm>
      <read-default>deny</read-default>
      <write-default>deny</write-default>
      <exec-default>deny</exec-default>
      <enable-external-groups>true</enable-external-groups>
      <rule-list>
```

```
    <name>admin</name>
    <group>PRIV15</group>
    <rule>
      <name>permit-all</name>
      <module-name>*</module-name>
      <access-operations>*</access-operations>
      <action>permit</action>
    </rule>
  </rule-list>
</nacm>
```

グループ メンバーシップ

ユーザのグループ メンバーシップは2つのソースから取得できます。1つ目は、認証に使用するAAAサーバで設定されているユーザの権限レベルです。2つ目は、/nacm/groupsサブツリーで設定されている権限レベルです。各権限レベルに対応するグループの名前は次のとおりです。

権限レベル	NACM グループ名
0	PRIV00
1	PRIV01
2	PRIV02
3	PRIV03
4	PRIV04
5	PRIV05
6	PRIV06
7	PRIV07
8	PRIV08
9	PRIV09
10	PRIV10
11	PRIV11
12	PRIV12
13	PRIV13
14	PRIV14
15	PRIV15

（注） 従来のIOSコマンド許可（権限レベルに基づくものなど）は、NETCONFまたはRESTCONFには適用されません。

（注） 権限レベルに基づいてNACMグループに付与されたアクセスは、権限レベルが高いNACMグループには本来適用されません。たとえば、PRIV10に適用されるルールは、PRIV11、PRIV12、PRIV13、PRIV14、およびPRIV15にも自動的に適用されるわけではありません。

NACM 権限レベルの依存関係

AAA 設定が **no aaa new-model** で設定されている場合は、ユーザに対してローカルに設定された権限レベルが使用されます。AAA 設定が **aaa new-model** で設定されている場合、権限レベルは、メソッドリスト **aaa authorization exec default** に関連付けられている AAA サーバによって決まります。

NACM の設定の管理と保守

NACM設定は、NETCONFまたはRESTCONFを使用して変更できます。ユーザがNCAM設定にアクセスできるようにするには、そのための明示的な権限を持たせる必要があります。つまり、NACMルールを使用します。/nacmサブツリーの下の設定は、**copy running-config startup-config** EXECコマンドが発行されるとき、または **cisco-ia:save-config** RPCが発行されるときは持続します。

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save-config xmlns="http://cisco.com/yang/cisco-ia"/>
</rpc>
```

（注） NETCONFセッションに適用されるNACMルールは、セッションの確立時に/nacmサブツリーで設定されているものです。/nacmサブツリーに変更を加えても、NETCONFセッションはすでに確立されているため影響を受けません。<kill-session> RPCまたは **clear netconf-yang session** EXECコマンドを使用して、不要なNETCONFセッションを強制的に終了することができます。NETCONF Killセッション（162ページ）を参照してください。

（注） 特定のデータへのアクセスを拒否するルールを作成する場合は、同じデータが複数のYANGモジュールとデータ ノードのパスを介して公開される可能性があるため、注意が必要です。たとえば、インターフェイス コンフィギュレーションは **Cisco-IOS-XE-native** と **ietf-interface** の両方を介して公開されます。同じ元データの1つの表現に適用される可能性があるルールは、そのデータの他の表現には適用されない場合があります。

NACM 設定のリセット

/nacm サブツリーの設定を初期設定にリセットするには、次のコマンドを使用します（「初期操作」を参照）。

```
Router#request platform software yang-management nacm reset-config
```

NACM の設定例

（注） ここで挙げている例は説明のみを目的とするものです。

次に、グループ設定の例を示します。

```
<nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
    <groups>
      <group>
        <name>administrators</name>
        <user-name>admin</user-name>
        <user-name>root</user-name>
      </group>

      <group>
        <name>limited-permission</name>
        <user-name>alice</user-name>
        <user-name>bob</user-name>
      </group>
    </groups>
  </nacm>
```

表 *20* : グループ設定の設定パラメータの説明

パラメータ	説明
<name>administrators</name>	グループ名
<user-name>admin</user-name>	ユーザ名
<user-name>root</user-name>	ユーザ名

次に、モジュール ルールを作成する例を示します。

```
<nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
  <rule-list>
    <name>only-ietf-interfaces</name>
    <group>limited-permission</group>
    <rule>
      <name>deny-native</name>
      <module-name>Cisco-IOS-XE-native</module-name>
      <access-operations>*</access-operations>
      <action>deny</action>
    </rule>
    <rule>
      <name>allow-ietf-interfaces</name>
      <module-name>ietf-interfaces</module-name>
```

```
      <access-operations>*</access-operations>
      <action>permit</action>
    </rule>
  </rule-list>
</nacm>
```

表 *21* : モジュール ルールを作成するための設定パラメータの説明

パラメータ	説明
<name>only-ietf-interfaces</name>	固有のルールリスト名
< group > permission </group >	ルールリストが適用されるグループ
<name>deny-native</name>	固有のルール名
<module-name>Cisco-IOS-XE-native</module-name>	YANG モジュールの名前
<access-operations>*</access-operations>	CRUDx の動作タイプ
<action>deny</action>	許可/拒否

次に、プロトコル操作ルールを作成する例を示します。

```
<nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
  <rule-list>
    <name>only-get</name>
    <group>limited-permission</group>

    <rule>
      <name>deny-edit-config</name>
      <module-name>ietf-netconf</module-name>
      <rpc-name>edit-config</rpc-name>
      <access-operations>exec</access-operations>
      <action>deny</action>
    </rule>
    <rule>
      <name>allow-get</name>
      <module-name>ietf-netconf</module-name>
      <rpc-name>get</rpc-name>
      <access-operations>exec</access-operations>
      <action>permit</action>
    </rule>
  </rule-list>
</nacm>
```

表 *22* : プロトコル操作ルールを作成するための設定パラメータの説明

パラメータ	説明
<name>only-get</name>	固有のルールリスト名
< group > permission </group >	ルールリストが適用されるグループ
<name>deny-edit-config</name>	固有のルール名

パラメータ	説明
<module-name>ietf-netconf</module-name>	RPC を含むモジュールの名前
<rpc-name>edit-config</rpc-name>	RPC の名前
<access-operations>exec</access-operations>	RPC の実行権限
<action>deny</action>	許可/拒否

次に、データ ノード ルールを作成する例を示します。

```
<nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
  <rule-list>
    <name>hide-enable-passwords</name>
    <group>limited-permission</group>

    <rule>
      <name>deny-enable-passwords</name>
     <path xmlns:ios="http://cisco.com/ns/yang/Cisco-IOS-XE-native">/ios:native/enable

      </path>
      <access-operations>*</access-operations>
      <action>deny</action>
    </rule>
  </rule-list>
</nacm>
```

表 *23*：データ ノード ルールを作成するための設定パラメータの説明

パラメータ	説明
<name>hide-enable-passwords</name>	固有のルールリスト名
< group > permission </group >	ルールリストが適用されるグループ
<name>deny-enable-passwords</name>	固有のルール名
<path xmlns:ios="http://cisco.com/ns/yang/Cisco-IOS-XE-native">/ios:native/enable</path>	許可または拒否されるデータ ノードへのパス
<access-operations>*</access-operations>	CRUDx の動作タイプ
<action>deny</action>	許可/拒否

次に、すべてのグループに対して、標準の NETCONF RPC <get> および <get-config> の使用、スキーマ ダウンロード RPC <get-schema>、およびモジュール **ietf-interfaces** にあるデータへの読み取り専用アクセスを許可する NACM 設定の例を示します。

```
<nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
      <rule-list>
        <name>readonly-protocol</name>
        <group>*</group>
        <rule>
          <name>get-permit</name>
          <module-name>ietf-netconf</module-name>
```

```
      <rpc-name>get</rpc-name>
      <access-operations>exec</access-operations>
      <action>permit</action>
    </rule>
    <rule>
      <name>get-config-permit</name>
      <module-name>ietf-netconf</module-name>
      <rpc-name>get-config</rpc-name>
      <access-operations>exec</access-operations>
      <action>permit</action>
    </rule>
    <rule>
      <name>get-schema-permit</name>
      <module-name>ietf-netconf-monitoring</module-name>
      <rpc-name>get-schema</rpc-name>
      <access-operations>exec</access-operations>
      <action>permit</action>
    </rule>
  </rule-list>
  <rule-list>
    <name>readonly-data</name>
    <group>*</group>
    <rule>
      <name>ietf-interfaces-permit</name>
      <module-name>ietf-interfaces</module-name>
      <access-operations>read</access-operations>
      <action>permit</action>
    </rule>
  </rule-list>
</nacm>
```

モデルベースの AAA に関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
IOS-XE、IOS-XR、および NX-OS プラットフォームのさまざまなリリースの YANG データ モデル	開発者に分かりやすい方法で Cisco YANG モデルにアクセスするには、*GitHub* リポジトリを複製し、*vendor/cisco* サブディレクトリに移動します。ここでは、IOS XE、IOS-XR、および NX-OS プラットフォームのさまざまなリリースのモデルを使用できます。

標準および RFC

標準/RFC	タイトル
RFC 6020	*YANG*：*Network Configuration Protocol*（*NETCONF*）向けデータ モデリング言語
RFC 6241	ネットワーク設定プロトコル（*NETCONF*）
RFC 6536	ネットワーク設定プロトコル（*NETCONF*）アクセス制御モデル

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

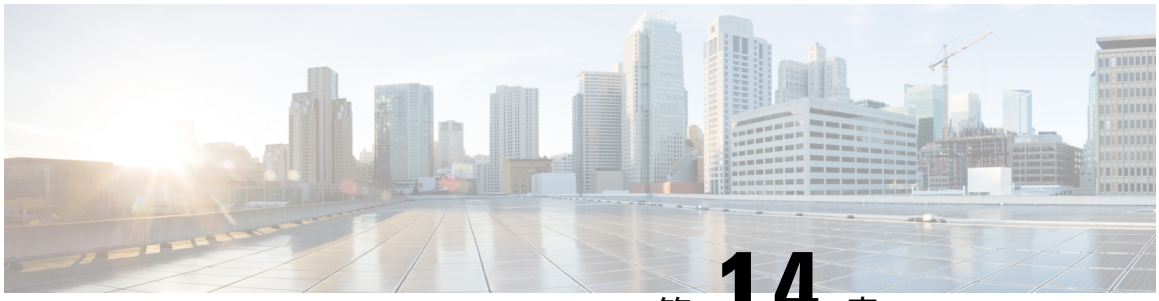
モデルベースの AAA に関する機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *24 :* プログラマビリティの機能情報：データ モデル

機能名	リリース	機能情報
モデルベースの AAA	Cisco IOS XE Fuji 16.8.1	この機能は、次のプラットフォームに実装されていました。 • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco CSR 1000v スイッチ • Cisco ISR 1100 シリーズ サービス統合型ルータ • Cisco ISR 4000 シリーズ サービス統合型ルータ • Cisco NCS 4200 シリーズ
	Cisco IOS XE Fuji 16.8.1a	この機能は、次のプラットフォームに実装されていました。 • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ



第 14 章

モデル駆動型テレメトリ

モデル駆動型テレメトリ

モデル駆動型テレメトリは、YANG モデル化されたデータをデータ コレクタにストリーミングするためのメカニズムを提供します。このモジュールでは、モデル駆動型テレメトリについて説明し、テレメトリ リモート プロシージャ コール（RPC）の例を示します。

モデル駆動型テレメトリの前提条件

- テレメトリを使用する際に必要なデータを理解して定義するには、YANGに関する知識が必要です。
- XML、XML 名前空間、および XML XPath の知識。
- IETF テレメトリ仕様で定義されている標準および原則の知識。
- urn:ietf:params:netconf:capability:notification:1.1 機能は、hello メッセージでリストする必要があります。この機能は、IETF テレメトリをサポートするデバイスでのみアドバタイズされます。
- NETCONF-YANG がデバイス上で設定済みであり稼働している必要があります。

（注） NETCONF を使用しない場合でも、テレメトリが機能するように NETCONF-YANG を設定する必要があります。NETCONF-YANG の設定の詳細については、「NETCONF プロトコル」モジュールを参照してください。

show platform software yang-management process コマンドを使用して、次のプロセスが実行中であることを確認します。

```
Device# show platform software yang-management process
```

```
confd : Running
nesd : Running
syncfd : Running
ncsshd : Running
dmiauthd : Running
nginx : Running
ndbmand : Running
pubd : Running
gnmib : Running
```

（注） プロセス pubd はモデル駆動型テレメトリ プロセスであり、これが実行していない場合にはモデル駆動型テレメトリは機能しません。

次の表に、各デバイス管理インターフェイス（DMI）プロセスの詳細を示します。

表 *25 :* フィールドの説明

デバイス管理インターフェイスプロセス名	主要な役割
confd	コンフィギュレーション デーモン
nesd	ネットワーク要素シンクロナイザデーモン
syncfd	同期デーモン（実行状態と対応するモデル間の同期を維持）
ncsshd	NETCONF セキュアシェル（SSH）デーモン
dmiauthd	DMI 認証デーモン。
nginx	NGINX Web サーバ。RESTCONF の Web サーバとして機能します。
ndbmand	NETCONF データベースマネージャ
pubd	モデル駆動型テレメトリに使用されるパブリケーションマネージャおよびパブリッシャ
gnmib	GNMI プロトコルサーバ。

NETCONF 固有の前提条件

- NETCONF とその使用方法に関する次の知識。
  - NETCONF セッションの確立。
  - hello および機能メッセージの送受信。

- 確立された NETCONF セッションによる YANG XML RPC の送受信詳細については、『*Configure NETCONF / YANG and Validate Example for Cisco IOS XE 16.x Platforms*』を参照してください。

NETCONF の有効化と検証

NETCONF の機能を確認するには、有効なユーザ名とパスワードを使用してデバイスへの SSH 接続を作成し、デバイスの機能を含む hello メッセージを受信します。

```
Device:~ USER1$ ssh -s cisco1@172.16.167.175 -p 830 netconf
cisco1@172.16.167.175's password: cisco1

<?xml version="1.0" encoding="UTF-8"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<capabilities>
<capability>urn:ietf:params:netconf:base:1.0</capability>
<capability>urn:ietf:params:netconf:base:1.1</capability>
<capability>urn:ietf:params:netconf:capability:writable-running:1.0</capability>
<capability>urn:ietf:params:netconf:capability:xpath:1.0</capability>
<capability>urn:ietf:params:netconf:capability:validate:1.0</capability>
<capability>urn:ietf:params:netconf:capability:validate:1.1</capability>
<capability>urn:ietf:params:netconf:capability:rollback-on-error:1.0</capability
.
.
.
</capabilities>
<session-id>2870</session-id></hello>]]>]]>

Use < ^C > to exit
```

hello メッセージに対して正常な応答を受信すると、NETCONF を使用する準備が整います。

RESTCONF 固有の前提条件

- RESTCONF とその使用方法に関する次の知識（RESTCONF を使用してサブスクリプションを作成する場合）。
- RESTCONF がデバイスで設定されている必要があります。
- RESTCONF は、RESTCONF *RFC 8040* に準拠した、正しい形式の Uniform Resource Identifier（URI）を送信する必要があります。

RESTCONF の有効化と検証

適切なクレデンシャルと次の URI を使用して、RESTCONF を検証します。

```
Operation: GET
Headers:
" Accept: application/yang-data.collection+json, application/yang-data+json,
application/yang-data.errors+json
" Content-Type: application/yang-data+json
Returned Output (omitted for breverity):
{
    "ietf-restconf:data": {
        "ietf-yang-library:modules-state": {
            "module": [
                {
```

```
                    "name": "ATM-FORUM-TC-MIB",
                    "revision": "",
                    "schema":
"https://10.85.116.28:443/restconf/tailf/modules/ATM-FORUM-TC-MIB",
                    "namespace": "urn:ietf:params:xml:ns:yang:smiv2:ATM-FORUM-TC-MIB"
                },
                {
                    "name": "ATM-MIB",
                    "revision": "1998-10-19",
                    "schema":
"https://10.85.116.28:443/restconf/tailf/modules/ATM-MIB/1998-10-19",
                    "namespace": "urn:ietf:params:xml:ns:yang:smiv2:ATM-MIB"
                },
                {
                    "name": "ATM-TC-MIB",
                    "revision": "1998-10-19",
                    "schema": "https://10.85.116.28:443/restconf/tailf/
..
<snip>
..
}
```

すべてのデバイス機能で前述の応答を受信すると、RESTCONF が正常に検証されます。

gRPC固有の前提条件

- キー値 Google Protocol Buffers（GPB）エンコーディングを理解する gRPC コレクタを設定します。

モデル駆動型テレメトリの制約事項

- yang-push ストリームを使用している場合、選択における自動階層は、変更時サブスクリプション向けにサポートされません。つまり、リストを選択するときに、リストの子リストが自動的には含まれません。たとえば、サブスクライバでは、子リストごとにサブスクリプションを手動で作成する必要があります。
- データアクセス許可のチェックはサポートされていません。サブスクライバによって要求されたすべてのデータが送信されます。
- サブツリーフィルタはサポートされていません。サブツリーフィルタが指定された場合、サブスクリプションは無効としてマークされます。
- サブスクリプションパラメータの中で複数の受信者を定義することはサポートされていません。最初の受信者の宛先だけが試行されます。他の定義済みの受信者は無視されます。

gRPC 固有の制限事項

- デバイスとレシーバ間の Transport Layer Security ベース（TLS ベース）の認証はサポートされていません。

  TLS ベースの認証は、Cisco IOS XE Amsterdam 17.1.1 以降のリリースでサポートされています。

yang-push 固有の制限

- サブスクリプションの Quality of Service（QoS）はサポートされていません。

モデル駆動型テレメトリについて

次のセクションでは、モデル駆動型テレメトリのさまざまな側面について説明します。

モデル駆動型テレメトリの概要

テレメトリは、自動の通信プロセスです。これにより、測定およびその他のデータがリモートポイントまたはアクセス不能なポイントで収集され、モニタ用の受信装置に送信されます。モデル駆動型テレメトリは、YANG モデル化されたデータをデータ コレクタにストリーミングするためのメカニズムを提供します。

アプリケーションでは、NETCONF、RESTCONF、または gRPC ネットワーク管理インターフェイス（gNMI）の各プロトコルを介した標準ベースの YANG データ モデルを使用して、必要とする特定のデータ項目をサブスクライブできます。サブスクリプションは CLI を使用して作成することもできます（設定済みサブスクリプションの場合）。

構造化データは、サブスクリプション基準とデータ タイプに基づき、定義されたパターンでまたは変更時にパブリッシュされます。

テレメトリ ロール

テレメトリを使用するシステムでは、さまざまなロールが関与します。このドキュメントでは、次のテレメトリ ロールを使用します。

- パブリッシャ：テレメトリ データを送信するネットワーク要素。
- 受信者：テレメトリデータを受信します。コレクタとも呼ばれます。
- コントローラ：サブスクリプションを作成するがテレメトリ データを受信しないネットワーク要素。作成したサブスクリプションに関連付けられたテレメトリデータが受信者に送信されます。管理エージェントまたは管理エンティティとも呼ばれます。
- サブスクライバ：サブスクリプションを作成するネットワーク要素。技術的には、受信者でもある必要はありませんが、このドキュメントではどちらも同じです。

サブスクリプションの概要

サブスクリプションは、テレメトリ ロール間の関連付けを作成する項目であり、ロール間で送信されるデータを定義します。

具体的には、サブスクリプションは、テレメトリデータの一部として要求される一連のデータを定義するために使用されます。たとえば、データがいつ必要か、データの書式設定の方法、また暗黙的でない場合は誰（どの受信者）がデータを受信するかを定義します。

サポートされているサブスクリプションの最大数はプラットフォームによって異なりますが、現在は 100 個のサブスクリプションがサポートされています。サブスクリプションは、設定済

みか動的のいずれかにすることができ、トランスポートプロトコルの任意の組み合わせを使用できます。有効なすべての設定済みサブスクリプションをアクティブにするために同時に多数のサブスクリプションが動作している場合、サブスクリプションの数が多すぎると、アクティブなサブスクリプションを削除したときに、非アクティブであるが有効な設定済みサブスクリプションの1つが試行されます。定期的にトリガーされるサブスクリプション（デフォルトの最小値は100センチ秒）と、変更時にトリガーされるサブスクリプションがサポートされています。

サブスクリプションの設定では、NETCONF やその他のノースバウンド プログラマブル インターフェイス（RESTCONF、gNMI など）がサポートされています。

Cisco IOS XE システムのテレメトリでは、ダイナミック サブスクリプションと設定済みサブスクリプションの 2 種類のサブスクリプションが使用されます。

動的サブスクリプションは、パブリッシャに接続するクライアント（サブスクライバ）によって作成されるため、ダイヤルインと見なされます。設定済みサブスクリプションでは、パブリッシャは受信者への接続を開始し、その結果ダイヤルアウトと見なされます。

ダイヤルインおよびダイヤルアウトのモデル駆動型テレメトリ

モデル駆動型テレメトリには、ダイヤルインとダイヤルアウトの 2 種類があります。

表 *26 :* ダイヤルインおよびダイヤルアウトのモデル駆動型テレメトリ

ダイヤルイン（動的）	ダイヤルアウト（静的または設定済み）
テレメトリの更新は、イニシエータまたはサブスクライバに送信されます。	テレメトリの更新は、指定された受信者またはコレクタに送信されます。
サブスクリプションの存続期間は、そのサブスクリプションを作成した接続（セッション）に結び付けられ、その存続期間中テレメトリの更新が送信されます。実行コンフィギュレーションでは変更は観察されません。	サブスクリプションは実行コンフィギュレーションの一部として作成されます。これは、コンフィギュレーションが削除されるまでデバイス設定として残ります。
ダイヤルイン サブスクリプションはリロード後に再起動する必要があります。これは、確立された接続またはセッションがステートフル スイッチオーバー時に kill されるためです。	ダイヤルアウト サブスクリプションはデバイス設定の一部として作成され、ステートフル スイッチオーバー後に自動的に受信者に再接続します。
サブスクリプション ID は、サブスクリプションの確立が成功したときに動的に生成されます。	サブスクリプション ID は固定であり、設定の一部としてデバイス上で設定されます。

データ ソースの仕様

サブスクリプション内のテレメトリ データのソースは、ストリームとフィルタを使用して指定されます。ここでのストリームとは、関連する一連のイベントを指します。RFC 5277 ではイ

ベントストリームを、いくつかの転送基準に一致する一連のイベント通知として定義しています。

通常は、ストリームからの一連のイベントはフィルタ処理されます。異なるストリームタイプごとに異なるフィルタ タイプが使用されます。

Cisco IOS XE は、yang-push と yang-notif-native の 2 つのストリームをサポートしています。

更新の通知

サブスクリプションの一部として、データが必要になるタイミングを指定できます。ただし、これはストリームによって異なります。ストリーム内で変更が行われたとき、またはイベントが発生した後にのみデータを使用できるようにするストリームもあれば、変更発生時に、または定義済みの時間間隔でデータを使用できるようにするストリームもあります。

この「タイミング」指定の結果は、対象のテレメトリ データを送る一連の更新通知となります。データの送信方法は、パブリッシャと受信者間の接続に使用されるプロトコルによって異なります。

サブスクリプション識別子

サブスクリプションは 32 ビットの正の整数値で識別されます。設定済みサブスクリプションの ID はコントローラによって設定され、動的サブスクリプションの場合はパブリッシャによって設定されます。

コントローラは、パブリッシャで作成された動的サブスクリプションとの競合を避けるために、設定済みサブスクリプションに使用する値を 0 ～ 2147483647 の範囲に制限する必要があります。動的サブスクリプションの ID 空間はグローバルです。つまり、独立して作成された動的サブスクリプションのサブスクリプション ID は重複しません。

サブスクリプション管理

管理操作の任意の形式を使用して、設定済みサブスクリプションの作成、削除、および変更を行うことができます。これには、CLI とネットワーク プロトコルの両方の管理操作が含まれます。

すべてのサブスクリプション（設定済みと動的）は、**show** コマンド、およびネットワーク プロトコル管理操作を使用して表示できます。

次の表で、サポートされているストリームとエンコーディング、およびサポートされている組み合わせについて説明します。入力としてのストリームは出力としてのプロトコルから独立していることを意図していますが、すべての組み合わせがサポートされているわけではありません。

表 *27:* サポートされるプロトコルの組み合わせ

トランスポートプロトコル	NETCONF		gRPC		gNMI	
	ダイヤルイン	ダイヤルアウト	ダイヤルイン	ダイヤルアウト	ダイヤルイン	ダイヤルアウト
Stream						
yang-push	対応	非対応	非対応	対応	対応	非対応
yang-notif-native	対応	非対応	非対応	対応	非対応	非対応
Encodings	XML	非対応	非対応	キー値 Google Protocol Buffers（kvGPB）	JSON_IETF	非対応

テレメトリの *RPC* サポート

確立された NETCONF セッションで、YANG XML リモート プロシージャ コール（RPC）の送受信が行えます。

テレメトリには <establish-subscription> RPC と <delete-subscription> RPC がサポートされています。

<establish-subscription> RPC が送信されると、パブリッシャからの RPC 応答には <rpc-reply> メッセージと、結果ストリングを含む <subscription-result> 要素が含まれます。

次の表は、<rpc-reply> メッセージでの応答と、応答の理由を示しています。

結果文字列	RPC	原因
ok	<establish-subscription> <delete-subscription>	成功
error-no-such-subscription	<delete-subscription>	指定されたテンプレートは存在しません。
error-no-such-option	<establish-subscription>	要求されたサブスクリプションはサポートされていません。
error-insufficient-resources	<establish-subscription>	サブスクリプションは次の理由により作成できません。 • サブスクリプションが多すぎる。

結果文字列	RPC	原因
		• 要求されたデータの量が大きすぎる。 • 定期的なサブスクリプションの間隔が短すぎる。
error-other	<establish-subscription>	その他の何らかのエラーです。

サービス *gNMI*

gNMI 仕様は、ハイレベル RPC を含む gNMI という名前の単一のトップレベルサービスを識別します。次に、サブスクライブサービス RPC を含むサービス定義を示します。

```
service gNMI{
  .
  .
  .
   rpc Subscribe(stream SubscribeRequest)
       returns (stream SubscribeResponse);
```

<subscribe RPC> は、動的サブスクリプションを要求するために管理エージェントによって使用されます。このRPCには一連のメッセージが含まれています。次のセクションでは、<subscribe RPC> でサポートされているメッセージについて説明します。

SubscribeRequest メッセージ

このメッセージは、指定されたパスのセットに対するターゲットからの更新を要求するためにクライアントによって送信されます。次に、メッセージの定義を示します。

```
message SubscribeRequest {
  oneof request {
    SubscriptionList subscribe = 1;
    PollRequest poll = 3;
    AliasList aliases = 4;
  }
  Repeated gNMI_ext.Extensions = 5;
}
```

（注） request.subscribe のみがサポートされます。

SubscribeResponse メッセージ

このメッセージは、確立された <subscribe RPC> を介してターゲットからクライアントに送信されます。次に、メッセージの定義を示します。

```
message SubscribeResponse {
  oneof response {
    Notification update = 1;
    Bool sync_response = 3;
    Error error = 4 [deprecated=true];
  }
}
```

（注） 通知の更新のみがサポートされます。

SubscriptionList メッセージ

このメッセージは、共通のサブスクリプション動作が必要なパスのセットを示すために使用されます。SubscriptionList メッセージの仕様内で、クライアントはモデル内の特定のプレフィックスに対する 1 つ以上のサブスクリプションを識別できます。次に、SubscriptionList メッセージの定義を示します。

```
message SubscriptionList {
  Path prefix = 1;
   repeated Subscription subscription = 2;
  bool use_aliases = 3;
  QOSMarking qos = 4;
  enum Mode {
      STREAM = 0;
     ONCE = 1;
     POLL = 2;
  }
  Mode mode = 5;
  bool allow_aggregation = 6;
  repeated ModelData use_models = 7;
  Encoding encoding = 8; // only JSON_IETF supported in R16.12
  Bool updates_only = 9;

}
```

（注） Path prefix（明示的な要素名のみ）、Subscription subscription、Mode mode STREAM、および Encoding encoding IETF_JSON がサポートされています。

プレフィックスメッセージ

有効なサブスクリプションリストには、xPath の（要求されたすべてのサブスクリプション間で）共有部分で構成された入力済みのプレフィックスが含まれている場合と、含まれていない場合があります。

```
message Path {
  repeated string element  = 1; [ deprecated ]
  string origin = 2;
   repeated PathElem elem = 3;
```

```
  optional string target = 4;

}
```

（注） origin（サポートされる値は「」と「openconfig」）、elem（サポートされる要素名はプレフィックスなし）、および target がサポートされます。

サブスクリプションメッセージ

このメッセージは、クライアントによってサブスクライブされるデータのセットを一般的に説明しています。通知動作を制御するために使用されるパスと属性が含まれます。次に、サブスクリプションメッセージの定義を示します。

```
message Subscription {
   Path path = 1;
    SubscriptionMode mode = 2;
    uint64 sample_interval = 3;
   bool suppress_redundant = 4;
   uint64 heartbeat_interval = 5;
}
```

（注） Path path、SubscriptionMode mode、Uint64 sample_interval、および Uint64 heartbeat_interval（値が 0 に設定されている場合のみ）がサポートされます。

パスメッセージ

有効なサブスクリプションには、パスが入力されています。これは、サブスクリプションリストに関連付けられたプレフィックスに追加されると、完全修飾パスを構成します。次に、パスメッセージの定義を示します。

```
message Path {
  repeated string element  = 1; [ deprecated ]
  string origin = 2;
   repeated PathElem elem = 3;
   optional string target = 4;

}
```

（注） origin（サポートされる値は「」と「openconfig」）、elem（サポートされる要素名はプレフィックスなし）、および target がサポートされます。

SubscriptionMode メッセージ

このメッセージは、通知の更新をトリガーする方法をターゲットに通知します。次に、SubscriptionMode メッセージの定義を示します。

```
enum SubscriptionMode {
   TARGET_DEFINED = 0;
   ON_CHANGE      = 1;
   SAMPLE           = 2;
}
```

（注） SAMPLE のみがサポートされます。

通知メッセージ

このメッセージは、サブスクリプションターゲットからコレクタにテレメトリデータを配信します。次に、通知メッセージの定義を示します。

```
message Notification {
  int64 timestamp = 1;
   Path prefix = 2;
  string alias = 3;
  repeated Update update = 4;
  repeated Path delete = 5;
  bool atomic = 6;
}
```

（注） タイムスタンプ、プレフィックス、および更新がサポートされます。

ダイナミックサブスクリプション管理

ここでは、動的サブスクリプションを作成および削除する方法について説明します。

NETCONF ダイヤルインの動的サブスクリプションの作成

動的サブスクリプションは、パブリッシャに接続し、その接続内部のメカニズム（通常はRPC）を使用してサブスクリプション作成のための呼び出しを行うサブスクライバによって作成されます。サブスクリプションの存続期間は、サブスクライバとパブリッシャ間の接続の存続期間に制限され、テレメトリデータはそのサブスクライバにのみ送信されます。これらのサブスクリプションは、パブリッシャまたはサブスクライバのいずれかが再起動された場合は存続しません。動的サブスクリプションの作成にはインバンドの <establish-subscription> RPC を使用できます。<establish-subscription> RPC は、IETF テレメトリのサブスクライバからネットワークデバイスに送信されます。RPC では、stream、xpath-filter、および period の各フィールドが必須です。

NETCONF による動的サブスクリプションの作成および削除に使用する RPC は、イベント通知のカスタムサブスクリプション *draft-ietf-netconf-subscribed-notifications-03* および *YANG* データストア プッシュ更新のサブスクライブ *draft-ietf-netconf-yang-push-07* で定義されています。

定期的な動的サブスクリプション

次に、ダイヤルインの定期的なサブスクリプションの例を示します。

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <establish-subscription
          xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications"
          xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
        <stream>yp:yang-push</stream>
        <yp:xpath-filter>/mdt-oper:mdt-oper-data/mdt-subscriptions</yp:xpath-filter>
        <yp:period>1000</yp:period>
      </establish-subscription>
    </rpc>
```

変更時動的サブスクリプション

次に、NETCONF を介した変更時動的サブスクリプションの例を示します。

```
<establish-subscription xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications"
xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
    <stream>yp:yang-push</stream>

<yp:xpath-filter>/cdp-ios-xe-oper:cdp-neighbor-details/cdp-neighbor-detail</yp:xpath-filter>

    <yp:dampening-period>0</yp:dampening-period>
</establish-subscription>
```

動的サブスクリプションの削除

動的サブスクリプションを削除するには、インバンドの <delete subscription> RPC、 **clear telemetry ietf subscription** コマンド、<kill-subscription> RPC を使用し、トランスポートセッションを切断します。

gNMI では、SubscribeRequest.subscribe.subscription の各サブスクリプションが個別のダイナミック サブスクリプション ID として生成されます。<kill-subscription> RPC または CLI のクリアを使用してこれらのサブスクリプション ID のいずれかを強制終了すると、サブスクライブ要求で指定されたすべてのサブスクリプションが強制終了されます。

Cisco IOS XE Gibraltar 16.10.1 で導入された <delete-subscription> RPC は、サブスクライバのみが発行でき、そのサブスクライバが所有するサブスクリプションのみを削除します。

Cisco IOS XE Gibraltar 16.11.1 以降のリリースでは、**clear telemetry ietf subscription** コマンドを使用してダイナミックサブスクリプションを削除できます。Cisco IOS XE Gibraltar 16.11.1 で導入された <kill-subscription> RPC は、**clear telemetry ietf subscription** コマンドと同じ方法でダイナミック サブスクリプションを削除します。

親のNETCONFセッションが切断されると、サブスクリプションも削除されます。ネットワーク接続が中断された場合は、 SSH またはNETCONF セッションがタイムアウトしてその後にサブスクリプションが削除されるまで、多少の時間がかかることがあります。

<kill-subscription> RPCは<delete-subscription> RPCと類似しています。ただし、<kill-subscription> RPCは、*subscription-id* 要素の代わりに、削除するサブスクリプションのIDを含む *identifier* 要素を使用します。ターゲットサブスクリプションで使用されるトランスポートセッションも、<delete-subscription> RPC で使用されているものと異なります。

CLI を使用したサブスクリプションの削除

次の出力例は、使用可能なすべてのサブスクリプションを示しています。

```
Device# show telemetry ietf subscription all

 Telemetry subscription brief

  ID               Type        State       Filter type
  ---------------------------------------------------------
  2147483648       Dynamic     Valid       xpath
  2147483649       Dynamic     Valid       xpath
```

次に、ダイナミック サブスクリプションを削除する例を示します。

```
Device# clear telemetry ietf subscription 2147483648
```

NETCONF <delete-Subscription> RPC を使用したサブスクリプションの削除

次に、NETCONF を使用してサブスクリプションを削除する例を示します。

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <delete-subscription xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications"
    xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
        <subscription-id>2147483650</subscription-id>
    </delete-subscription>
</rpc>
```

NETCONF <kill-Subscription> RPC を使用したサブスクリプションの削除

次に、<kill-subscription> RPC を使用してサブスクリプションを削除する例を示します。

```
<get>
<filter>
<mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
<mdt-subscriptions/>
</mdt-oper-data>
</filter>
</get>

* Enter a NETCONF operation, end with an empty line

<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="2">
  <data>
    <mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
```

```
      <mdt-subscriptions>
        <subscription-id>2147483652</subscription-id>
        <base>
…
        </base>
        <type>sub-type-dynamic</type>
        <state>sub-state-valid</state>
        <comments/>
        <mdt-receivers>
…
        </mdt-receivers>
       <last-state-change-time>2018-12-13T21:16:48.848241+00:00</last-state-change-time>

      </mdt-subscriptions>
      <mdt-subscriptions>
        <subscription-id>2147483653</subscription-id>
        <base>
…
        </base>
        <type>sub-type-dynamic</type>
        <state>sub-state-valid</state>
        <comments/>
        <mdt-receivers>
…
        </mdt-receivers>
       <last-state-change-time>2018-12-13T21:16:51.319279+00:00</last-state-change-time>

      </mdt-subscriptions>
      <mdt-subscriptions>
        <subscription-id>2147483654</subscription-id>
        <base>
…
        </base>
        <type>sub-type-dynamic</type>
        <state>sub-state-valid</state>
        <comments/>
        <mdt-receivers>
…
        </mdt-receivers>
       <last-state-change-time>2018-12-13T21:16:55.302809+00:00</last-state-change-time>

      </mdt-subscriptions>
      <mdt-subscriptions>
        <subscription-id>2147483655</subscription-id>
        <base>
…
        </base>
        <type>sub-type-dynamic</type>
        <state>sub-state-valid</state>
        <comments/>
        <mdt-receivers>
…
        </mdt-receivers>
       <last-state-change-time>2018-12-13T21:16:57.440936+00:00</last-state-change-time>

      </mdt-subscriptions>
    </mdt-oper-data>
  </data>
</rpc-reply>
<kill-subscription xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications"
xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
    <identifier>2147483653</identifier>
</kill-subscription>
```

```
* Enter a NETCONF operation, end with an empty line

<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="2">
  <subscription-result xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications"

xmlns:notif-bis="urn:ietf:params:xml:ns:yang:ietf-event-notifications">notif-bis:ok</subscription-result>
</rpc-reply>
<get>
<filter>
<mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
<mdt-subscriptions/>
</mdt-oper-data>
</filter>
</get>

* Enter a NETCONF operation, end with an empty line

<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="2">
  <data>
    <mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
      <mdt-subscriptions>
        <subscription-id>2147483652</subscription-id>
        <base>
…
        </base>
        <type>sub-type-dynamic</type>
        <state>sub-state-valid</state>
        <comments/>
        <mdt-receivers>
…
        </mdt-receivers>
       <last-state-change-time>2018-12-13T21:16:48.848241+00:00</last-state-change-time>

      </mdt-subscriptions>
      <mdt-subscriptions>
        <subscription-id>2147483654</subscription-id>
        <base>
…
        </base>
        <type>sub-type-dynamic</type>
        <state>sub-state-valid</state>
        <comments/>
        <mdt-receivers>
…
        </mdt-receivers>
       <last-state-change-time>2018-12-13T21:16:55.302809+00:00</last-state-change-time>

      </mdt-subscriptions>
      <mdt-subscriptions>
        <subscription-id>2147483655</subscription-id>
        <base>
…
        </base>
        <type>sub-type-dynamic</type>
        <state>sub-state-valid</state>
        <comments/>
        <mdt-receivers>
…
        </mdt-receivers>
       <last-state-change-time>2018-12-13T21:16:57.440936+00:00</last-state-change-time>

      </mdt-subscriptions>
```

```
   </mdt-oper-data>
  </data>
</rpc-reply>
```

設定済みサブスクリプションの管理

ここでは、設定済みサブスクリプションを作成、変更、および削除する方法について説明します。

設定済みサブスクリプションの作成

設定済みサブスクリプションは、コントローラによるパブリッシャでの管理操作によって作成され、サブスクリプションによって定義されたテレメトリデータの受信者の指定が明示的に含まれています。これらのサブスクリプションは、パブリッシャの再起動後も持続します。

設定済みサブスクリプションは複数の受信者を使用して設定できますが、最初の有効な受信者のみが使用されます。受信者がすでに接続されている場合、または接続中の場合は、他の受信者への接続は試行されません。その受信者が削除されると、別の受信者が接続されます。

設定済みダイヤルアウト サブスクリプションは、次の方法でデバイスに設定されます。

- 設定 CLI を使用し、コンソール/VTY を介してデバイス設定に変更を加えます。
- NETCONF/RESTCONF を使用し、目的のサブスクリプションを設定します。

ここでは、設定済みサブスクリプションを作成するための RPC の例を示します。

定期的なサブスクリプション

次の例は、CLIを使用して、設定済みサブスクリプションのトランスポートプロトコルとしてgRPC を設定する方法を示しています。

```
telemetry ietf subscription 101
  encoding encode-kvgpb
  filter xpath /memory-ios-xe-oper:memory-statistics/memory-statistic
  stream yang-push
  update-policy periodic 6000
  source-vrf Mgmt-intf
  receiver ip address 10.28.35.45 57555 protocol grpc-tcp
```

次の RPC の例は、NETCONF を使用して定期的なサブスクリプションを作成し、60 秒ごとにテレメトリの更新を受信者に送信する方法を示します。

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"><edit-config>
 <target>
  <running/>
 </target>
 <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <mdt-config-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-cfg">
   <mdt-subscription>
    <subscription-id>200</subscription-id>
    <base>
     <stream>yang-push</stream>
     <encoding>encode-kvgpb</encoding>
     <period>6000</period>
     <xpath>/memory-ios-xe-oper:memory-statistics/memory-statistic</xpath>
```

```
    </base>
    <mdt-receivers>
     <address>10.22.23.48</address>
     <port>57555</port>
     <protocol>grpc-tcp</protocol>
    </mdt-receivers>
   </mdt-subscription>
  </mdt-config-data>
 </config>
</edit-config>
</rpc>
```

次に、RESTCONF を使用して定期的なサブスクリプションを作成する RPC の例を示します。

```
URI:https://10.85.116.28:443/restconf/data/Cisco-IOS-XE-mdt-cfg:mdt-config-data
Headers:
application/yang-data.collection+json, application/yang-data+json,
application/yang-data.errors+json
Content-Type:
application/yang-data+json
BODY:
{
"mdt-config-data": {
 "mdt-subscription":[
 {
  "subscription-id": "102",
  "base": {
   "stream": "yang-push",
   "encoding": "encode-kvgpb",
                    "period": "6000",
   "xpath": "/memory-ios-xe-oper:memory-statistics/memory-statistic"
  }
        "mdt-receivers": {
            "address": "10.22.23.48"
            "port": "57555"
        }
 }
 ]
}
}
```

変更時サブスクリプション

次の RPC の例は、NETCONF を使用して変更時サブスクリプションを作成し、ターゲットデータベースに変更が生じた場合にのみ更新を送信する方法を示します。

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"><edit-config>
 <target>
  <running/>
 </target>
 <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <mdt-config-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-cfg">
   <mdt-subscription>
    <subscription-id>200</subscription-id>
    <base>
     <stream>yang-push</stream>
     <encoding>encode-kvgpb</encoding>
     <no-synch-on-start>false</no-synch-on-start>
     <xpath>/cdp-ios-xe-oper:cdp-neighbor-details/cdp-neighbor-detail</xpath>
    </base>
    <mdt-receivers>
     <address>10.22.23.48</address>
     <port>57555</port>
     <protocol>grpc-tcp</protocol>
```

```
    </mdt-receivers>
   </mdt-subscription>
  </mdt-config-data>
 </config>
</edit-config>
</rpc>
```

次の RPC の例は、RESTCONF を使用して変更時サブスクリプションを作成する方法を示します。

```
URI:
https://10.85.116.28:443/restconf/data/Cisco-IOS-XE-mdt-cfg:mdt-config-data
Headers:
application/yang-data.collection+json, application/yang-data+json,
application/yang-data.errors+json
Content-Type:
application/yang-data+json
BODY:
{
"mdt-config-data": {
 "mdt-subscription":[
 {
  "subscription-id": "102",
  "base": {
   "stream": "yang-push",
   "encoding": "encode-kvgpb",
                     "dampening period": "0",
   "xpath": "/cdp-ios-xe-oper:cdp-neighbor-details/cdp
                           -neighbor-detail "
  }
        "mdt-receivers": {
            "address": "10.22.23.48"
            "port": "57555"
        }
 }
 ]
}
}
```

gNMI ダイヤルイン サブスクリプション

次に、gNMI ダイヤルイン サブスクリプションの例を示します。

```
subscribe: <
  prefix: <>
  subscription: <
    path: <
      origin: "openconfig"
      elem: <name: "routing-policy">
    >
    mode: SAMPLE
    sample_interval: 10000000000
  >
  mode: STREAM
  encoding: JSON_IETF
>'

subscribe: <
  prefix: <>
  subscription: <
    path: <
      origin: "legacy"
      elem: <name: "oc-platform:components">
```

```
      elem: <
        name: "component"
        key: <
          key: "name"
          value: "PowerSupply8/A"
        >
      >
      elem: <name: "power-supply">
      elem: <name: "state">
    >
    mode: SAMPLE
    sample_interval: 10000000000
  >
  mode: STREAM
  encoding: JSON_IETF
>'
```

設定済みサブスクリプションの変更

設定済みサブスクリプションを変更するには、次の 2 つの方法があります。

- NETCONF <edit-config> RPC などの管理プロトコル設定操作
- CLI（サブスクリプションの作成と同じ手順）

サブスクリプションの受信者はアドレスとポート番号によって識別されます。受信者を変更することはできません。受信者の特性（プロトコル、プロファイルなど）を変更するには、先に受診者を削除してから新しい受信者を作成する必要があります。

有効なサブスクリプションの有効な受信者設定が切断状態にあり、管理側で受信者への接続のセットアップ時に新しい試行を強制する場合は、同一の特性を持つ受信者を書き換える必要があります。

設定済みサブスクリプションの削除

CLI または管理操作を使用して、設定済みサブスクリプションを削除できます。**no telemetry ietf subscription** コマンドは、設定済みサブスクリプションを削除します。設定されたサブスクリプションは、設定インターフェイスからのみ削除できます。

CLI を使用したサブスクリプションの削除

```
Device# configure terminal
Device(config)# no telemetry ietf subscription 101
Device(config)# end
```

NETCONF を使用したサブスクリプションの削除

次の RPC の例は、設定済みサブスクリプションを削除する方法を示しています。

```
<edit-config>
   <target>
     <running/>
   </target>
   <config>
     <mdt-config-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-cfg">
```

```
      <mdt-subscription  operation="delete">
      <subscription-id>102</subscription-id>
     </mdt-subscription>
    </mdt-config-data>
  </config>
</edit-config>
```

gRPC サブスクリプションの *FQDN* サポート

gRPC テレメトリ サブスクリプションは設定ベースです。つまり、ユーザーはデバイス設定の一部として受信ホストとその他のサブスクリプションパラメータを指定する必要があります。このレシーバ設定は、テレメトリの更新を送信するための接続の詳細を決定するために使用されます。gRPC サブスクリプション機能の FQDN サポートの導入により、IP アドレスに加え、完全修飾ドメイン名（FQDN）も gRPC サブスクリプションに使用できます。

テレメトリ サブスクリプションでは、レシーバの詳細をサブスクリプションの一部として指定することも、独立して設定することも可能になりました。ここで、レシーバには名前があり、サブスクリプションを設定するときにこの名前を使用してレシーバを指定します。どちらの場合も、複数のサブスクリプションに同じレシーバ名を指定できます。

この機能をディセーブルにできません。

名前付きレシーバ

FQDN のサポートにより、名前付きレシーバ設定と呼ばれる新しいレシーバ設定方法が導入されました。名前付きレシーバは、サブスクリプションとは無関係に存在できるトップレベルの設定エンティティです。名前付きレシーバは名前で識別されます。この名前は任意の文字列で、システムで名前付きレシーバレコードのインデックスまたはキーとなります。名前付きレシーバ設定には、レシーバに関連付けられているすべての設定が含まれます。これらの設定はサブスクリプションに依存しません。

名前付きレシーバを使用する利点は次のとおりです。

- さまざまなタイプのレシーバをサポートできます。
- より正確な状態および診断の情報を得られます。
- IP アドレスの代わりにホスト名を使用して、プロトコルレシーバのホストを指定できます。
- 複数のサブスクリプションで使用されるレシーバのパラメータを、1 つの場所で変更できます。

次のプロトコルタイプ名前付きレシーバだけがサポートされます。

- cloud-native：クラウド ネイティブ プロトコル
- cntp-tcp：Civil Network Time Protocol （CNTP）TCP プロトコル
- cntp-tls：CNTP TLS プロトコル
- grpc-tcp：gRPC TCP プロトコル

- grpc-tls：gRPC TLS プロトコル
- native：ネイティブプロトコル
- tls-native：ネイティブ TLS プロトコル

名前付きプロトコルレシーバ

名前付きプロトコルレシーバは、プロトコルを使用するテレメトリの転送を指定するために使用されます。名前付きプロトコルレシーバは、レシーバを識別する名前に加えて、ホスト指定も使用します。ホスト指定には、ホスト名または IP アドレス、および宛先ポート番号を指定します。セキュアプロトコル転送もプロファイル文字列を使用します。

（注） 有効な名前付きプロトコルレシーバが作成されると、レシーバに自動的に接続されません。レシーバへの接続を作成するには、指定されたプロトコルレシーバが、少なくとも 1 つのサブスクリプションによって要求される必要があります。

CLI または YANG モデルを使用して、名前付きプロトコルレシーバを設定できます。

名前付きプロトコルレシーバの YANG モデルを使用した設定

YANG モデル Cisco-IOS-XE-mdt-cfg には、名前付きプロトコルレシーバが含まれています。最上位の mdt-config-data コンテナ内のコンテナ mdt-named-protocol-rcvrs には、mdt-named-protocol-rcvr 構造体のリストがあります。このグループには、次の 5 つのメンバーがあります。

- Name：リスト内のインデックス
- プロトコル
- Profile
- ホストネーム
- ポート番号

次に、名前付きプロトコルレシーバの作成方法を示す、NETCONF RPC の例を示します。

```
<edit-config>
 <target>
  <running/>
 </target>
 <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <mdt-config-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-cfg">
   <mdt-named-protocol-rcvrs>
    <mdt-named-protocol-rcvr>
     <name>receiver1</name>
     <protocol>tls-native</protocol>
     <profile>tls-trustpoint</profile>
     <host>
      <hostname>rcvr.test.com</hostname>
     </host>
     <port>45000</port>
```

```
    </mdt-named-protocol-rcvr>
   </mdt-named-protocol-rcvrs>
  </mdt-config-data>
 </config>
</edit-config>
```

名前付きレシーバを使用したサブスクリプションの設定

サブスクリプションで名前付きレシーバを使用するには、レシーバタイプとレシーバ名の両方を指定する必要があります。すべてのレシーバ固有の設定は名前付きレシーバ設定の一部であるため、追加のレシーバ設定は必要ありません。ただし、名前付きプロトコルレシーバは、接続解決プロセスの一部として、サブスクリプションの送信元 VRF および送信元アドレスを引き続き使用します。

サポートされる名前レシーバタイプは protocol のみです。

サブスクリプションは名前付きレシーバまたはレガシーレシーバのいずれかを使用できますが、両方を使用することはできません。レガシーレシーバが設定されている場合、サブスクリプション レシーバ タイプと名前付きレシーバ名の設定はブロックされます。同様に、サブスクリプション レシーバ タイプまたは名前付きレシーバを指定した場合は、レガシーレシーバを設定できません。

複数のレシーバが設定されている場合でも、サブスクリプションは1つのレシーバのみを使用することに注意してください。

レガシーレシーバを使用するサブスクリプションと名前付きレシーバを使用するサブスクリプションは、同じ接続を使用できます。ただし、これは推奨されません。

名前付きレシーバサブスクリプション設定の YANG モデルを使用した設定

名前付きレシーバを使用する場合、rcvr-type でサポートされる唯一の値は rcvr-type-protocol です。レガシーレシーバを使用する場合、この値はデフォルトの rcvr-type-unspecified です。

次に、NETCONF RPC の例を示します。名前付きプロトコルレシーバを使用してサブスクリプションを作成する方法を示しています。

```
<edit-config>
 <target>
  <running/>
 </target>
 <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <mdt-config-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-cfg">
   <mdt-subscription>
    <subscription-id>1</subscription-id>
    <base>
     <rcvr-type>rcvr-type-protocol</rcvr-type>
    </base>
    <mdt-receiver-names>
     <mdt-receiver-name>
      <name>receiver1</name>
     </mdt-receiver-name>
    </mdt-receiver-names>
   </mdt-subscription>
  </mdt-config-data>
 </config>
```

```
</edit-config>
```

名前付きレシーバの動作と動作状態

名前付きレシーバオブジェクトとサブスクリプション レシーバオブジェクト（名前付きレシーバを参照する）には、2つの異なる動作状態があります。その動作状態は有効または無効です。名前付きレシーバが無効になる最も一般的な理由は、設定が不完全であることですが、他の理由も考えられます。名前付きレシーバの動作状態ビューには、レシーバが無効である理由をテキストで説明するフィールドがあります。レシーバの状態が有効な場合、このフィールドは空です。

CLI を使用した名前付きレシーバの状態の表示

すべてのタイプの名前付きレシーバの状態を表示するには、**show telemetry receiver** コマンドを使用します。**all** キーワードは、すべての名前付きレシーバに関する情報を簡単な形式で表示し、**name** キーワードは、指定された名前付きレシーバに関する詳細情報を表示します。

次に、**show telemetry receiver all** コマンドの出力例を示します。

```
Device# show telemetry receiver all

Telemetry receivers

  Name       <…>     Type        Profile               State       Explanation
  -----------<…>-------------------------------------------------------------------
  receiver1  <…>     protocol    tls-trustpoint        Valid
```

次に、**show telemetry receiver name** コマンドの出力例を示します。

```
Device# show telemetry receiver name receiver1

  Name: receiver1
  Profile: tls-trustpoint
  State: Valid
  Last State Change: 08/12/20 19:55:54
  Explanation:
  Type: protocol
  Protocol: tls-native
  Host: rcvr.test.com
  Port: 45000
```

名前付きレシーバの YANG モデルを使用した状態

名前付きレシーバの状態は、Cisco-IOS-XE-mdt-oper-v2 YANG モデルを使用して取得できます。mdt-oper-v2-data コンテナには、すべての名前付きレシーバの動作状態を含む mdt-named-receivers リストが含まれています。

次に、名前付きレシーバの状態を取得する NETCONF 応答の例を示します。

```
<get>
 <filter>
  <mdt-oper-v2-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper-v2">
```

```
   <mdt-named-receivers/>
  </mdt-oper-v2-data>
 </filter>
</get>

<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="2">
 <data>
  <mdt-oper-v2-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper-v2">
   <mdt-named-receivers>
    <name>receiver1</name>
    <profile>tls-trustpoint</profile>
    <params>
     <protocol>tls-native</protocol>
<host>
      <hostname>rcvr.test.com </hostname>
     </host>
     <port>45000</port>
    </params>
    <state>named-rcvr-state-valid</state>
    <last-state-change-time>2020-…:00</last-state-change-time>
   </mdt-named-receivers>
  </mdt-oper-v2-data>
 </data>
</rpc-reply>
```

サブスクリプションレシーバの動作と動作状態

サブスクリプションレシーバは、実際のサブスクリプションレシーバまたはコレクタに接続するサブスクリプション関連のオブジェクトです。コレクタに到達するために必要なメカニズムはレシーバタイプに固有ですが、接続は、サブスクリプションがレシーバまたはコレクタに到達できるようにするために使用されるエンティティです。

サブスクリプションレシーバの状態は、レシーバへの接続を要求および使用する機能に基づいており、複数の状態があります。それぞれの状態は、サブスクリプションがレシーバまたはコレクタに更新を送信できるようにするために必要な他のリソースの制御に関連付けられています。

サブスクリプションレシーバの状態

サブスクリプションレシーバの動作状態は、設定された名前（接続のインデックス）、レシーバの状態、状態に関する説明またはメモ、および最後の状態変更の時刻で構成されます。説明文字列は常に使用されるわけではありません。

サブスクリプションレシーバの考えられる状態を次の表に示します。

表 *28 :* サブスクリプションレシーバの状態

サブスクリプションレシーバの状態		説明
CLI 値	**YANG 値**	
Disconnected（切断）	rcvr-state-disconnected	レシーバは切断され、再接続は試行されません。

サブスクリプションレシーバの状態		説明
Resolving	rcvr-state-resolving	レシーバに到達するために必要な接続パラメータを解決しています。
Transport requested	rcvr-state-transport-requested	レシーバに到達するための接続要求が、Resolving 状態から決定された接続パラメータを使用していました。
接続中（Connecting）	rcvr-state-connecting	サブスクリプションをレシーバに接続するために必要なリソースが割り当てられています。
接続されている状態	rcvr-state-connecting	サブスクリプションがレシーバに接続され、更新をレシーバに送信できるようになりました。
Disconnecting	rcvr-state-disconnected	接続で使用されているリソースが再割り当てされています。

YANG 値 rcvr-state-invalid は、レガシーレシーバでのみ使用されます。無効なサブスクリプションレシーバは接続できないため、サブスクリプションレシーバの状態が無効な場合は disconnected に設定されます。説明文字列で、invalid のサブスクリプションレシーバと disconnected のサブスクリプションレシーバを区別できます。

サブスクリプションレシーバは、次の理由で切断されることがあります。

- サブスクリプションの別のレシーバが切断されていない。
- 接続のセットアップが永続的に失敗した。
- 名前付きレシーバが存在しない。
- 名前付きレシーバがサブスクリプションで指定されたタイプではない。
- 名前付きレシーバが無効である。
- サブスクリプションが無効である。
- 要求された接続が別のレシーバによって使用されている。

サブスクリプションレシーバの接続

このセクションでは、サブスクリプションレシーバが接続を使用する方法について説明します。

テレメトリ接続

テレメトリ接続は、サブスクリプションがレシーバに到達するために使用するトランスポートインスタンスを表し、単に動作状態を表します。テレメトリ接続は、整数のインデックス値で識別されます。接続に関するその他の情報は、サブスクリプションが使用するように設定されているレシーバのタイプに基づく接続のタイプに固有です。

セキュアなシスコ独自のトランスポートでは、設定された名前付きレシーバのホスト部分は、接続のセットアップ時にレシーバが提供する証明書の識別名（DN）と一致する必要があります。このため、同じ接続を使用する複数のレシーバを持つことはできません。

この項で説明するすべての状態は、すべてのタイプの接続で該当しますが、すべてが使用されるわけではありません。

表 *29*：テレメトリの接続状態

接続状態		説明
CLI 値	**YANG 値**	
Pending	con-state-pending	接続が作成されましたが、まだ開始されていません。
Connecting	con-state-connecting	接続をセットアップする要求が進行中です。
Active	con-state-active	接続が確立されており、サブスクリプションレシーバで使用できます。
Disconnecting	con-state -disconnecting	接続が切断され、サブスクリプションレシーバによる解放を待機しています。

接続に関連付けられているその他の動作状態には、リモートレシーバ（使用可能な場合はピア）の ID、および最後の状態変更の時刻などがあります。

テレメトリプロトコル接続

ここでは、プロトコルタイプ接続と、これらが名前付きプロトコルレシーバに割り当てられたサブスクリプションレシーバによりどのように使用されるかについて説明します。

表 *30*：プロトコルタイプ接続のパラメータ

パラメータ	発信元	注
Destination IP address	名前付きレシーバホスト	ホストはドメイン名を使用するため、ドメイン名の解決が必要になる場合があります。

パラメータ	発信元	注
Destination port number	名前付きレシーバポート	明示的に設定する必要があります。
Source VRF	サブスクリプション（指定されている場合）	VRF が指定されていない場合、デフォルトの VRF が使用されます。それ以外の場合、VRF 名は内部識別子に解決されます。
Source IP address	サブスクリプション（指定されている場合）	指定しない場合、送信元 IP アドレスは VRF および宛先 IP アドレスに基づいて決定されます。

これらのパラメータの一部は、サブスクリプションレシーバの親サブスクリプションの設定に基づいています。

設定から接続パラメータを解決するときに、順序が指定されていない場合は、最初に VRF が決定され、次に宛先 IP アドレス、最後に送信元 IP アドレスが決定されます。解決の特定のステップが非永続的に失敗する場合、5 秒間隔で無限に再試行されます。

接続は、要求されるとすぐにインスタンス化されます。つまり、最初のサブスクリプションレシーバが resolving 状態から transport requested 状態に移行するとすぐに、サブスクリプションレシーバによって解決されたパラメータを持つ接続インスタンスが作成されます。

要求された接続が正常に設定され、テレメトリで使用されると、接続状態は connected に変わります。これは、Cisco IOS XE デバイスとレシーバデバイスの間に接続が存在することを意味します。レシーバが使用するリソースを再割り当てするために、リソースを使用するサブスクリプションレシーバに、接続が設定されたことが通知されます。これらのサブスクリプションレシーバの状態は、connecting に移行して、サブスクリプションをレシーバに接続するために必要なリソースを設定します。これらのリソースが確保されると、サブスクリプションレシーバの状態が connected に変更され、レシーバが更新通知を受信します。

テレメトリ接続がアクティブになることができない理由の一部を次に示します。

- 接続先に到達できない。
- リモートホストポートにリスナーが存在しない。
- リモートホストポートのリスナーのタイプが正しくない。
- Authentication failures（認証エラー）：

（注） 接続セットアップが進行中のときは、接続セットアップを開始するために必要なパラメータが正常に解決されているため、この接続を使用するサブスクリプションレシーバはすべて connecting 状態になります。

接続セットアップが失敗したときに実行されるアクションは、プロトコルによって異なります。次の表に、単一のセットアップ要求内の接続の再試行動作と、接続セットアップ要求が失敗した場合の再解決要求の再試行動作を示します。この動作は、レガシーレシーバによって要求された接続でも同じです。

表 *31* : プロトコル固有の再試行間隔

Protocol	接続の再試行回数	再解決要求
• grpc-tcp • grpc-tls	1、3、4、7 秒の間隔で 5 回再試行	制限なし。接続の再試行が失敗すると、継続的に再解決を要求します（試行ごとに 14 秒）。
• cloud-native • cntp-tcp • cntp-tls • native • tls-native		5、10、15、20、25、および30秒。

名前付きレシーバ接続のトラブルシューティング

サブスクリプションが設定されている場合、一般的な問題の1つは、テレメトリ更新メッセージが受信されないことです。原因として、送信するイベントがないか、サブスクリプションが無効であることが考えられます。ここでは、名前付きレシーバ接続で発生する一般的な問題のトラブルシューティング方法について説明します。

テレメトリプロセスからのログ、および一部の **show** コマンドの出力には、名前付きレシーバ設定のトラブルシューティングに使用できる情報が含まれています。

表 *32* : 名前付きレシーバ接続のトラブルシューティング

問題	確認方法/症状	対処法
サブスクリプション構成が無効。	**show telemetry ietf subscription** *id* **details**	サブスクリプションの設定を修正します。
サブスクリプションレシーバが無効。	**show telemetry ietf subscription** *id* **receiver**	名前付きレシーバの設定を修正します。
サブスクリプションレシーバの接続パラメータを解決できない。	**show telemetry ietf subscription** *id* **receiver** サブスクリプションレシーバの状態が、解決中の状態から変わらない。	レシーバ、ネットワーク設定、またはインターフェイスの状態を確認します。

問題	確認方法/症状	対処法
サブスクリプションレシーバ接続がアップ状態にならない。	**show telemetry ietf subscription** *id***receiver** サブスクリプションレシーバの状態が、解決中から接続中に何度も変化する。	解決された接続が有効であり、レシーバまたはコレクタが到達可能で、指定されたトランスポートを使用してインバウンド接続を受け入れることができることを確認します。
サブスクリプションレシーバの接続が拒否される。	**show telemetry ietf subscription** *id***receiver** サブスクリプションレシーバの状態が、切断を除くすべての状態で変化し続ける。	コレクタのタイプが正しいこと、および設定されている認証と許可が有効であることを確認します。
サブスクリプションレシーバが接続されているが、更新が受信されない。	**show telemetry internal subscription** *id***stats** メッセージドロップカウントが増加しているが、送信されたレコードが増加しない。	コレクタが更新通知のフローに追随できていることを確認します。
サブスクリプションレシーバが接続されているが、更新が受信されない。	**show telemetry internal subscription** カウントが変化しない。	変更時のサブスクリプションの場合は、実際にイベントが発生していないことを確認します。 定期的なサブスクリプションの場合は、更新期間が短く、時間は 100 分の 1 秒単位で指定されていることを確認します。

show telemetry internal connection：このコマンドは、オプションの接続インデックス値を取ります。インデックスが指定されていない場合は、使用されているすべての接続の基本的な接続パラメータ情報が表示されます。コマンドで接続インデックスを指定すると、接続に関する詳細が表示されます。コマンド出力はトランスポート固有であり、すべてのトランスポートで使用できるとは限りません。このコマンドの出力は変更される場合があります。

show telemetry internal diagnostics：このコマンドは、すべてのテレメトリログと動作状態をダンプしようとします。問題を報告するときは、可能な限り問題の発生時のすぐ後にこのコマンドを使用し、**show running-config | section telemetry** コマンドの出力も提供すると、解決に役立ちます。

変更時サブスクリプション YANG モデルの表示

変更時サブスクリプションをサポートしている YANG モデルのリストを表示するには、**show platform software ndbman switch** {*switch-number* | **active** | **standby**} **models** コマンドを使用します。

（注） Cisco Catalyst 9800-80 ワイヤレスコントローラでは、**show platform software ndbman chassis** {*number* | **active** | **standby**} **models** コマンドを使用します。

Cisco-IOS-XE-mdt-capabilities-oper.YANG モデルには、変更時サブスクリプションでサポートされるモデルも表示されます。

サブスクリプションのモニタリング

CLI および管理プロトコル操作を使用して、すべてのタイプのサブスクリプションを監視できます。

CLI

テレメトリのサブスクリプションに関する情報を表示するには、**show telemetry ietf subscription** コマンドを使用します。コマンドからの出力例を、次に示します。

```
Device#  show telemetry ietf subscription 2147483667 detail

Telemetry subscription detail:

  Subscription ID: 2147483667
  State: Valid
  Stream: yang-push
  Encoding: encode-xml
  Filter:
    Filter type: xpath
    XPath: /mdt-oper:mdt-oper-data/mdt-subscriptions
  Update policy:
    Update Trigger: periodic
    Period: 1000
  Notes:
```

NETCONF

次に、テレメトリのサブスクリプションに関する情報を表示する NETCONF メッセージの例を示します。

```
<get>
<filter>
<mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
<mdt-subscriptions/>
</mdt-oper-data>
</filter>
</get>


* Enter a NETCONF operation, end with an empty line
<?xml version="1.0" encoding="UTF-8"?>
```

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="2">
  <data>
    <mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
      <mdt-subscriptions>
        <subscription-id>101</subscription-id>
        <base>
          <stream>yang-push</stream>
          <encoding>encode-kvgpb</encoding>
          <source-vrf>RED</source-vrf>
          <period>10000</period>
          <xpath>/ios:native/interface/Loopback[name="1"]</xpath>
        </base>
        <type>sub-type-static</type>
        <state>sub-state-valid</state>
        <comments/>
        <mdt-receivers>
          <address>5.22.22.45</address>
          <port>57500</port>
          <protocol>grpc-tcp</protocol>
          <state>rcvr-state-connecting</state>
          <comments/>
          <profile/>
          <last-state-change-time>1970-01-01T00:00:00+00:00</last-state-change-time>
        </mdt-receivers>
        <last-state-change-time>1970-01-01T00:00:00+00:00</last-state-change-time>
      </mdt-subscriptions>
      <mdt-subscriptions>
        <subscription-id>2147483648</subscription-id>
        <base>
          <stream>yang-push</stream>
          <encoding>encode-xml</encoding>
          <source-vrf/>
          <period>1000</period>

<xpath>/if:interfaces-state/interface[name="GigabitEthernet0/0"]/oper-status</xpath>
        </base>
        <type>sub-type-dynamic</type>
        <state>sub-state-valid</state>
        <comments/>
        <mdt-receivers>
          <address>5.22.22.45</address>
          <port>51259</port>
          <protocol>netconf</protocol>
          <state>rcvr-state-connected</state>
          <comments/>
          <profile/>
          <last-state-change-time>1970-01-01T00:00:00+00:00</last-state-change-time>
        </mdt-receivers>
        <last-state-change-time>1970-01-01T00:00:00+00:00</last-state-change-time>
      </mdt-subscriptions>
    </mdt-oper-data>
  </data>
</rpc-reply>
```

ストリーム

ストリームは、サブスクライブ可能な一連のイベントを定義します。ほぼすべてのイベントがこの一連のイベントとして有効です。ただし、各ストリームの定義に従い、すべてのイベントの候補は何らかの形で関連しています。ここでは、サポートされているストリームについて説明します。

サポートされているストリームのセットを表示するには、管理プロトコル操作を使用して、*mdt-streams* コンテナにある Cisco-IOS-XE-mdt-oper モジュール（YANG モデル Cisco-IOS-XE-mdt-oper.yang からのもの）から *streams* テーブルを取得します。

次に、NETCONF を使用して、サポートされているストリームを取得する例を示します。

```
<get>
<filter>
<mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
<mdt-streams/>
</mdt-oper-data>
</filter>
</get>

* Enter a NETCONF operation, end with an empty line

<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="2">
  <data>
    <mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
      <mdt-streams>
        <stream>native</stream>
          <stream>yang-notif-native</stream>
          <stream>yang-push</stream>
      </mdt-streams>
    </mdt-oper-data>
  </data>
</rpc-reply>
```

この例は、native、yang-notif-native、yang-push の 3 つのストリームがサポートされていることを示しています。ストリーム native は汎用としては使用できず、無視できます。

（注） 現在のところ、サポートされているストリームのリストを返す CLI はありません。

YANG-push ストリーム

yang-push ストリームは、サポートされている YANG モデルにより記述される、構成データベース内と運用データベース内のデータです。このストリームは、ストリームの中で対象とするデータを指定するための XPath フィルタをサポートしており、XPath 式は対象のデータを定義する YANG モデルに基づきます。

このストリームの更新通知は、対象のサブスクリプションについて、データの変更時または固定間隔で送信される場合がありますが、両方に対応して送信されることはありません。現在存在しないデータのサブスクリプションは許可され、通常のサブスクリプションとして実行されます。

サポートされている唯一のターゲットデータベースは「実行中」です。

変更時機能の決定

現在のところ、変更時サブスクリプションを使用し、サブスクライブ可能なデータのタイプについて YANG モデルの中で指定する手段はありません。変更時サブスクリプションを使用し

て、サブスクライブができないデータにサブスクライブしようとすると、失敗（動的）となるか、無効なサブスクリプション（設定済み）となります。On-Change パブリケーションの詳細については、「*On-Change Publication for yang-push*」の項を参照してください。

IETF ドラフトへの準拠

yang-push ストリームを使用するテレメトリは、テレメトリの IETF NETCONF ワーキング グループの初期ドラフトに基づいています。これらを次に示します。

- イベント通知のカスタム サブスクリプション、バージョン *03*
- *YANG* データストア プッシュ更新のサブスクライブ、バージョン *07*

（注） 対応するドラフトに記載されている次の機能はサポートされていません。

- サブツリー フィルタ
- アウトオブバンドの通知
- サポート対象として明示的に記載されていないすべてのサブスクリプション パラメータ

YANG-push の *XPath* フィルタ

サブスクライブ先の *yang-push* ストリーム内のデータセットは、XPath フィルタを使用して指定する必要があります。XPath 式には次のガイドラインが適用されます。

- XPath 式では、リストまたはコンテナに 1 つのエントリを指定するためのキーを持たせることができます。サポートされているキー指定の構文は次のとおりです。

  ```
  [{key name}={key value}]
  ```

  XPath 式の例を次に示します。

  ```
  filter xpath
  /rt:routing-state/routing-instance[name="default"]/ribs/rib[name="ipv4-default"]/routes/route

  # VALID!
  ```

  複合キーを使用するには、複数のキー指定を使用します。キーの名前と値は正確である必要があります。範囲やワイルドカードによる値はサポートされていません。

- XPath 式で、キーの間に [] を使用して複数のキーを選択し、”. で文字列をカプセル化します。XPath 式の例を次に示します。

  ```
  filter xpath
  /environment-ios-xe-oper:environment-sensors/environment-sensor[location=\"Switch\
  1\"]
  [name=\"Inlet\ Temp\ Sens\"]/current-reading
  ```

Cisco Catalyst 9800 ワイヤレスコントローラでサポートされる *XPath* 式

Cisco IOS XE Bengaluru、17.4.1 では、次の OpenConfig XPath 式のセットが Cisco Catalyst 9800 シリーズ ワイヤレスコントローラでサポートされています。

テレメトリ サブスクリプションを有効にするには、NETCONF、RESTCONF、gNMI プロトコルなどのプログラマビリティ インターフェイスを使用して、次の RPC を実行します。

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <provision-aps xmlns="http://openconfig.net/yang/wifi/ap-manager">
        <provision-ap>
          <mac>eth_mac_of_the_AP</mac>
          <config>
            <mac>eth_mac_of_the_AP</mac>
            <hostname>AP_NAME</hostname>
          </config>
        </provision-ap>
      </provision-aps>
    </config>
  </edit-config>
</rpc>
```

次に示す XPath 式はすべて、openconfig-access-points YANG モデルの一部です。ただし、最後の式だけは openconfig-ap-manager YANG モデルの一部です。テレメトリ操作が正しく機能するように、OpenConfig モデルに基づいて設定が行われていることを確認します。

- /access-points/access-point/radios/radio/state
- /access-points/access-point/radios/radio/neighbors/neighbor
- /access-points/access-point/radios/radio/neighbors/neighbor/state
- /access-points/access-point/ssids/ssid/bssids/bssid/state/counters
- /access-points/access-point/ssids/ssid/clients/client/state/counters
- /access-points/access-point/ssids/ssid/clients/client/client-rf/state
- /access-points/access-point/ssids/ssid/clients/client/client-connection/state
- /access-points/access-point/system/aaa/server-groups/server-group/servers/server/radius/state
- /joined-aps/joined-ap/state/opstate

XPath をサブスクライブすると、サブスクライブされた XPath とその階層内のすべての XPath のデータを受信します。たとえば、/access-points/access-point/radios/radio/state へサブスクライブすると、関連付けられているすべてのリーフとその下のサブコンテナのデータが配信されます。

情報のサブセットのみが必要な場合は、XPath 式でフィルタを設定して更新を制限します。特定のアクセスポイント（AP）のデータをフィルタリングするには、ノードの後にキーを使用します。たとえば、ホスト名が ‘my_hostname’ である AP のデータを受信するには、サブスクリプション XPath: access-point[hostname=’my_hostname’] を使用します。データ更新には、定義済みの限定されたサブセットだけでなく、すべてのリーフからのデータオブジェクトが含まれることに注意してください。

拡張性に関する情報

次の表に、3 つの異なるスケールシナリオにおけるそれぞれの収集ポイントの最小推奨間隔を示します。

シナリオ 1： 4 つの SSID によるフルスケール

表 *33 :* 設定

AP	2,000
クライアント	30,000
AP ごとの SSID	4
AP ごとの BSSID	8
AP ごとの物理ネイバー	12
AP ごとのネイバー	96

表 *34 :* 推奨間隔

収集ポイント	レコード	推奨間隔（秒） コレクタ X1	推奨間隔（秒） コレクタ X2
参加	2000	30	60
AAA	2000	30	60
無線	4000	30	60
クライアント RF	30,000	30	60
クライアント CNTR	30,000	30	60
クライアント CONN	30,000	60	120
BSSID	16,000	90	180
Neighbor	192,000	180	360

シナリオ 2： 6 つの SSID によるフルスケール

表 *35 :* 設定

AP	2,000
クライアント	30,000
AP ごとの SSID	6

AP ごとの BSSID	12
AP ごとの物理ネイバー	12
AP ごとのネイバー	144

表 *36 :* 推奨間隔

収集ポイント	レコード	推奨間隔（秒） コレクタ X1	推奨間隔（秒） コレクタ X2
参加	2000	30	60
AAA	2000	30	60
無線	4000	30	60
クライアント RF	30,000	30	60
クライアント CNTR	30,000	30	60
クライアント CONN	30,000	60	120
BSSID	24000	120	240
Neighbor	288,000	240	420

シナリオ 3： 6 つの SSID による減少スケール

表 *37 :* 設定

AP	1,000
クライアント	15,000
AP ごとの SSID	6
AP ごとの BSSID	12
AP ごとの物理ネイバー	12
AP ごとのネイバー	144

表 *38 :* 推奨間隔

収集ポイント	レコード	推奨間隔（秒） コレクタ X1	推奨間隔（秒） コレクタ X2
参加	1000	NA	30
AAA	1000	NA	30
無線	2000	NA	30
クライアント RF	15,000	NA	30
クライアント CNTR	15,000	NA	30
クライアント CONN	15,000	NA	30
BSSID	12,000	NA	120
Neighbor	144,000	該当なし	180

YANG-push の定期パブリケーション

定期的なサブスクリプションでは、サブスクライブ対象情報による最初のプッシュ更新は即時に送信されます。ただしデバイスがビジー状態であったりネットワークが混雑していたりすると遅延することがあります。次に更新は、設定された定期タイマーの満了時に送信されます。たとえば、期間を 10 分と設定すると、サブスクリプションの作成直後に最初の更新が送信され、その後は 10 分おきに送信されます。

期間は、定期的なプッシュ更新間のセンチ秒（1/100 秒）単位の時間です。期間が 1000 であれば、サブスクライブ対象情報の更新は 10 秒ごとになります。設定できる最小の期間間隔は 100（つまり 1 秒）です。デフォルト値はありません。この値は、動的サブスクリプションの場合は <establish-subscription> RPC で明示的に設定する必要があり、設定済みサブスクリプションの場合は設定で明示的に設定する必要があります。

定期的な更新には、サポートされているすべてのトランスポートプロトコルに関連するサブスクライブ対象のデータ要素またはテーブルのフルコピーが含まれています。

定期的なサブスクリプションを使用して空のデータをサブスクライブすると、要求された期間で空の更新通知が送信されます。データが存在するようになると、次の期間の値が通常の更新通知として送信されます。

YANG-push の変更時パブリケーション

変更時サブスクリプションを作成する場合は、ダンプニング期間がないことを示すためにダンプニング期間を 0 に設定する必要があります。その他の値はサポートされていません。

変更時サブスクリプションでは、最初のプッシュ更新は、サブスクライブされたデータのセット全体です（IETF の文書で定義されている初期同期）。これは制御できません。以降の更新は、データが変更され、変更後のデータのみで構成されている場合に送信されます。ただし、変更とみなされる最小のデータ分解能は行です。したがって、変更時サブスクリプションが行内のリーフに対するものである場合、その行のいずれかの項目が変更されると、更新通知が送信されます。更新通知の正確な内容はトランスポート プロトコルによって異なります。

また、変更時サブスクリプションは階層状ではありません。つまり、子コンテナを持つコンテナにサブスクライブしても、子コンテナ内の変更はサブスクリプションには認識されません。

現在存在しないデータのサブスクリプションは許可され、通常のサブスクリプションとして実行されます。初期同期更新通知は空であり、データが利用可能になるまでそれ以上更新されません。

yang-notif-native ストリーム

yang-notif-native ストリームは、パブリッシャ内の任意の YANG 通知であり、通知の元のイベントソースで Cisco IOS XE のネイティブのテクノロジーが使用されています。このストリームは、対象となる通知を指定する XPath フィルタもサポートしています。このストリームの更新通知は、通知の対象になるイベントが発生した場合にのみ送信されます。

このストリームは変更時サブスクリプションのみをサポートしているため、ダンプニング間隔として値 0 を指定する必要があります。

（注） 現在のところ、このストリームは Google リモート プロシージャ コール（gRPC）経由ではサポートされていません。

yang-notif-native の *XPath* フィルタ

サブスクライブ先の yang-notif-native ストリーム内のデータセットは、XPath フィルタを使用して指定します。XPath 式には次のガイドラインが適用されます。

- XPath 式は単一のオブジェクトを指定する必要があります。このオブジェクトには、コンテナ、リーフ、リーフリスト、またはリストを使用できます。
- XPath 式は YANG 通知全体を指定する必要があります。属性のフィルタ処理はサポートされていません。
- XPath式では、単一のサブスクリプションで複数のオブジェクトをサポートできるように、結合演算子（|）を使用できます。

Cisco Catalyst 9800 ワイヤレスコントローラの *XPath* 値と対応するレート

Cisco-IOS-XE-wireless-mesh-rpc の XPath `/exec-linktest-ap/data-rate-idx` で許容されている値と対応するレートを次に示します。

```
ewlc-mesh-linktest-rate-idx-1 1 Mbps
ewlc-mesh-linktest-rate-idx-2 2 Mbps
ewlc-mesh-linktest-rate-idx-3 5 Mbps
ewlc-mesh-linktest-rate-idx-4 6 Mbps
```

```
ewlc-mesh-linktest-rate-idx-5 9 Mbps
ewlc-mesh-linktest-rate-idx-6 11 Mbps
ewlc-mesh-linktest-rate-idx-7 12 Mbps
ewlc-mesh-linktest-rate-idx-8 18 Mbps
ewlc-mesh-linktest-rate-idx-9 24 Mbps
ewlc-mesh-linktest-rate-idx-10 36 Mbps
ewlc-mesh-linktest-rate-idx-11 48 Mbps
ewlc-mesh-linktest-rate-idx-12 54 Mbps
ewlc-mesh-linktest-rate-idx-13 108 Mbps
ewlc-mesh-linktest-rate-idx-14 m0
ewlc-mesh-linktest-rate-idx-15 m1
ewlc-mesh-linktest-rate-idx-16 m2
ewlc-mesh-linktest-rate-idx-17 m3
ewlc-mesh-linktest-rate-idx-18 m4
ewlc-mesh-linktest-rate-idx-19 m5
ewlc-mesh-linktest-rate-idx-20 m6
ewlc-mesh-linktest-rate-idx-21 m7
ewlc-mesh-linktest-rate-idx-22 m8
ewlc-mesh-linktest-rate-idx-23 m9
ewlc-mesh-linktest-rate-idx-24 m10
ewlc-mesh-linktest-rate-idx-25 m11
ewlc-mesh-linktest-rate-idx-26 m12
ewlc-mesh-linktest-rate-idx-27 m13
ewlc-mesh-linktest-rate-idx-28 m14
ewlc-mesh-linktest-rate-idx-295 m15
```

TLDP 変更時の通知

Targeted Label Discovery Protocol（T-LDP）は、直接接続されていないラベルスイッチドルータ（LSR）間の LDP セッションです。TLDP 変更時の通知機能は、TLDP セッションが起動または停止したとき、および TLDP が設定またはディセーブルになったときにユーザに通知します。通知を機能させるには、TLDP を有効にする必要があります。

イベントベースの通知は、次の 2 つのシナリオで生成されます。

- 設定されたイベントは、TLDPが設定され、デバイスから削除されたときに生成されます。通知は、TLDP セッションがアップまたはダウンしたときにも生成されます。
- 通知は、TLDP セッションがアップまたはダウンしたときにも生成されます。

トランスポート プロトコル

データの送信方法は、パブリッシャと受信者間の接続に使用されるプロトコルによって決まります。このプロトコルはトランスポートプロトコルと呼ばれ、設定済みサブスクリプションの管理プロトコルからは独立しています。トランスポートプロトコルは、データのエンコーディング（XML、Google Protocol Buffers（GPB）など）と更新通知自体の形式に影響を与えます。

（注） また、選択したストリームも更新通知の形式に影響を与える場合があります。

サポートされているトランスポートプロトコルは、gNMI、gRPC、NETCONF です。

NETCONF プロトコル

NETCONFプロトコルは、動的サブスクリプションのトランスポートにのみ使用でき、*yang-push* ストリームと *yang-notif-native* ストリームで使用できます。

NETCONFをトランスポートプロトコルとして使用する場合は、次の3つの更新通知形式が使用されます。

- サブスクリプションでyang-pushストリームが使用されていて、定期的な場合、または、初期同期更新通知が変更時サブスクリプションで送信される場合。
- サブスクリプションでyang-pushストリームが使用されていて、初期同期更新通知以外の変更時サブスクリプションの場合。
- サブスクリプションでyang-notif-nativeストリームが使用されている場合。

yang-push 形式

yangpush ソースストリームがNETCONFを介してXMLエンコーディングのトランスポートとして送信される場合、2つの更新通知形式が定義されます。これらの更新通知形式は、*draft-ietf-netconf-yang-push-07* に基づいています。詳細については、IETFドラフトの3.7項を参照してください。

yang-notif-native 形式

ソース ストリームがyang-notif-nativeの場合、NETCONFを介してXMLでエンコードされるときの更新通知の形式はRFC 7950によって定義されています。詳細については、RFCの7.16.2項を参照してください。

yang-pushストリームの形式とは異なり、サブスクリプションIDは更新通知にはありません。

gRPC プロトコル

gRPC プロトコルは、設定済みサブスクリプションのトランスポートに対してのみ使用でき、yang-pushストリームでのみ使用できます。gRPC トランスポートプロトコルではkvGPBエンコーディングのみがサポートされています

gRPCプロトコルに基づく受信者の接続の再試行（指数バックオフ）がサポートされています。

protoファイルで定義されたテレメトリメッセージについては、 mdt_grpc_dialout.proto および https://github.com/cisco-ie/cisco-proto/blob/9cc3967cb1cabbb3e9f92f2c46ed96edf8a0a78b/proto/xe/telemetry.prototelemetry.proto を参照してください。

テレメトリにおけるハイ アベイラビリティ

テレメトリの動的な接続は、アクティブなスイッチかスイッチスタック内のメンバーへのSSH、またはハイアベイラビリティ対応デバイスでのアクティブなルートプロセッサへのSSHを介してNETCONFセッションで確立されます。切り替え後は、テレメトリのサブスクリプションを伝送するNETCONFセッションを含め、暗号を使用するすべてのセッションを破棄し、再確立する必要があります。また、スイッチオーバー後にすべてのダイナミック サブスクリプショ

ンを再作成する必要があります。 gNMI ダイヤルイン サブスクリプションも、SSH を介した NETCONF セッションと同様に機能します。

gRPC ダイヤルアウト サブスクリプションは、アクティブなスイッチまたはスタック メンバの実行コンフィギュレーションの一部としてデバイスに設定されます。スイッチオーバーが発生すると、テレメトリ受信者への既存の接続が切断され、再接続されます（受信者へのルートが残っている限り）。サブスクリプションを再設定する必要はありません。

（注） デバイスのリロード時には、サブスクリプションの設定をデバイスのスタートアップ コンフィギュレーションに同期させる必要があります。これにより、デバイスの再起動後もサブスクリプション設定がデバイス上にそのまま残ります。必要なプロセスが起動して実行されると、デバイスはテレメトリ受信者への接続を試行し、通常の動作を再開します。

サンプルのモデル駆動型テレメトリ RPC

次のセクションでは、RPC の例のリストを示し、サブスクリプションの設定方法について説明します。

設定済みサブスクリプションの管理

（注） 現在のところ、設定済みサブスクリプションの管理に使用できるのは gRPC プロトコルのみです。

手順の概要

1. **enable**
2. **configure terminal**
3. **telemetry ietf subscription** *id*
4. **stream yang-push**
5. **filter xpath** *path*
6. **update-policy** {**on-change** | **periodic**} *period*
7. **encoding encode-kvgpb**
8. **source-vrf** *vrf-id*
9. **source-address** *source-address*
10. **receiver ip address** *ip-address receiver-port* **protocol** *protocol* **profile** *name*
11. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**telemetry ietf subscription** *id* 例： `Device(config)# telemetry ietf subscription 101`	テレメトリのサブスクリプションを作成し、テレメトリサブスクリプション モードを開始します。
ステップ 4	**stream yang-push** 例： `Device(config-mdt-subs)# stream yang-push`	サブスクリプションのストリームを設定します。
ステップ 5	**filter xpath** *path* 例： `Device(config-mdt-subs)# filter xpath /memory-ios-xe-oper:memory-statistics/memory-statistic`	サブスクリプションの XPath フィルタを指定します。
ステップ 6	**update-policy** {**on-change** | **periodic**} *period* 例： `Device(config-mdt-subs)# update-policy periodic 6000`	サブスクリプションの定期的な更新ポリシーを設定します。
ステップ 7	**encoding encode-kvgpb** 例： `Device(config-mdt-subs)# encoding encode-kvgpb`	kvGPB エンコードを指定します。
ステップ 8	**source-vrf** *vrf-id* 例： `Device(config-mdt-subs)# source-address Mgmt-intf`	ソースの VRF インスタンスを設定します。
ステップ 9	**source-address** *source-address* 例： `Device(config-mdt-subs)# source-vrf 192.0.2.1`	送信元アドレスを設定します。
ステップ 10	**receiver ip address** *ip-address receiver-port* **protocol** *protocol* **profile** *name* 例： `Device(config-mdt-subs)# receiver ip address 10.28.35.45 57555 protocol grpc-tcp`	通知の受信者の IP アドレス、プロトコル、およびプロファイルを設定します。

	コマンドまたはアクション	目的
ステップ 11	**end** 例： `Device(config-mdt-subs)# end`	テレメトリサブスクリプションのコンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

gRPC の変更時サブスクリプションの設定

手順の概要

1. **enable**
2. **configure terminal**
3. **telemetry ietf subscription** *id*
4. **stream yang-push**
5. **filter xpath** *path*
6. **update-policy** {**on-change** | **periodic** *period*}
7. **encoding encode-kvgpb**
8. **receiver ip address** *ip-address receiver-port* **protocol** *protocol* **profile** *name*
9. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**telemetry ietf subscription** *id* 例： `Device(config)# telemetry ietf subscription 8`	テレメトリのサブスクリプションを作成し、テレメトリサブスクリプション モードを開始します。
ステップ 4	**stream yang-push** 例： `Device(config-mdt-subs)# stream yang-push`	サブスクリプションのストリームを設定します。
ステップ 5	**filter xpath** *path* 例： `Device(config-mdt-subs)# filter xpath /iosxe-oper:ios-oper-db/hwidb-table`	サブスクリプションの XPath フィルタを指定します。

	コマンドまたはアクション	目的
ステップ 6	**update-policy** {**on-change** | **periodic** *period*} 例： `Device(config-mdt-subs)# update-policy on-change`	サブスクリプションの変更時更新ポリシーを設定します。
ステップ 7	**encoding encode-kvgpb** 例： `Device(config-mdt-subs)# encoding encode-kvgpb`	kvGPB エンコードを指定します。
ステップ 8	**receiver ip address** *ip-address receiver-port* **protocol** *protocol* **profile** *name* 例： `Device(config-mdt-subs)# receiver ip address 10.22.22.45 45000 protocol grpc_tls profile secure_profile`	通知の受信者の IP アドレス、プロトコル、およびプロファイルを設定します。
ステップ 9	**end** 例： `Device(config-mdt-subs)# end`	テレメトリサブスクリプションのコンフィギュレーションモードを終了し、特権 EXEC モードに戻ります。

応答コードの受信

サブスクリプションが正常に作成されると、デバイスはサブスクリプション結果 notif-bis:ok およびサブスクリプション ID で応答します。次に、動的サブスクリプションの応答 RPC メッセージの例を示します。

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
<subscription-result xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications"
xmlns:notif-bis="urn:ietf:params:xml:ns:yang:ietf-event-notifications">notif-bis:
ok</subscription-result>
<subscription-id
xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications">2147484201</subscription-id>
</rpc-reply>
```

NETCONF ダイヤルインのサブスクリプションプッシュ更新の受信

デバイスからプッシュされるサブスクリプション更新は XML RPC 形式であり、それらが作成された同じ NETCONF セッションにより送信されます。サブスクライブ対象情報の要素またはツリーは datastore-contents-xml タグ内で返されます。次に示すのは、サブスクライブ対象情報を提供するサンプル RPC メッセージです。

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <eventTime>2017-05-09T21:34:51.74Z</eventTime>
    <push-update xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
        <subscription-id>2147483650</subscription-id>
        <datastore-contents-xml>
            <cpu-usage
xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-process-cpu-oper"><cpu-utilization>
```

```
            <five-minutes>5</five-minutes></cpu-utilization></cpu-usage>
          </datastore-contents-xml>
    </push-update>
</notification>
```

サブスクリプションが行われる情報要素が空である場合、またはそれが動的（名前付きアクセスリストなど）であり存在しない場合、定期更新は空になり、自己終結 *datastore-contents-xml* タグを持つことになります。次に示すのは、定期更新が空である RPC メッセージの例です。

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <eventTime>2017-05-09T21:34:09.74Z</eventTime>
    <push-update xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
        <subscription-id>2147483649</subscription-id>
        <datastore-contents-xml />
    </push-update>
</notification>
```

サブスクリプションの詳細の取得

現在のサブスクリプションの一覧を取得するには、<get> RPC を Cisco-IOS-XE-mdt-oper モデルに送信します。現在のサブスクリプションの一覧を表示するには、**show telemetry ietf subscription** コマンドも使用できます。

次に、<get> RPC メッセージの例を示します。

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter>
      <mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
        <mdt-subscriptions/>
      </mdt-oper-data>
    </filter>
  </get>
</rpc>
```

次に、RPC 応答の例を示します。

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <data>
    <mdt-oper-data xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-mdt-oper">
      <mdt-subscriptions>
        <subscription-id>2147485164</subscription-id>
        <base>
          <stream>yang-push</stream>
          <encoding>encode-xml</encoding>
          <period>100</period>
          <xpath>/ios:native/router/ios-rip:rip/ios-rip:version</xpath>
        </base>
        <type>sub-type-dynamic</type>
        <state>sub-state-valid</state>
        <comments/>
        <updates-in>0</updates-in>
        <updates-dampened>0</updates-dampened>
```

```
        <updates-dropped>0</updates-dropped>
      </mdt-subscriptions>
    </mdt-oper-data>
  </data>
</rpc-reply>
```

次に、**show telemetry ietf subscription dynamic brief** コマンドの出力例を示します。

```
Device# show telemetry ietf subscription dynamic brief

Telemetry subscription brief

  ID               Type        State       Filter type
  ---------------------------------------------------------
  2147483667       Dynamic     Valid       xpath
  2147483668       Dynamic     Valid       xpath
  2147483669       Dynamic     Valid       xpath
```

次に、**show telemetry ietf subscription** *subscription-ID***detail** コマンドの出力例を示します。

```
Device# show telemetry ietf subscription 2147483667 detail

Telemetry subscription detail:

  Subscription ID: 2147483667
  State: Valid
  Stream: yang-push
  Encoding: encode-xml
  Filter:
    Filter type: xpath
    XPath: /mdt-oper:mdt-oper-data/mdt-subscriptions
  Update policy:
    Update Trigger: periodic
    Period: 1000
  Notes:
```

次に、**show telemetry ietf subscription all detail** コマンドの出力例を示します。

```
Device# show telemetry ietf subscription all detail

Telemetry subscription detail:

  Subscription ID: 101
  Type: Configured
  State: Valid
  Stream: yang-push
  Encoding: encode-kvgpb
  Filter:
    Filter type: xpath
    XPath: /iosxe-oper:ios-oper-db/hwidb-table
  Update policy:
    Update Trigger: on-change
    Synch on start: Yes
    Dampening period: 0
  Notes:
```

次の RPC の例は、RESTCONF を使用してサブスクリプションの詳細を取得する方法を示します。

```
Subscription details can also be retrieved through a RESTCONF GET request to the
Cisco-IOS-XE-mdt-oper database:
URI:
https://10.85.116.28:443/restconf/data/Cisco-IOS-XE-mdt-oper:
mdt-oper-data/mdt-subscriptions
Headers:
application/yang-data.collection+json, application/yang-data+json,
application/yang-data.errors+json
Content-Type:
application/yang-data+json
Returned output:
{
  "Cisco-IOS-XE-mdt-oper:mdt-subscriptions": [
    {
      "subscription-id": 101,
      "base": {
        "stream": "yang-push",
        "encoding": "encode-kvgpb",
        "source-vrf": "",
        "no-synch-on-start": false,
        "xpath": "/iosxe-oper:ios-oper-db/hwidb-table"
      },
      "type": "sub-type-static",
      "state": "sub-state-valid",
      "comments": "",
      "updates-in": "0",
      "updates-dampened": "0",
      "updates-dropped": "0",
      "mdt-receivers": [
        {
          "address": "5.28.35.35",
          "port": 57555,
          "protocol": "grpc-tcp",
          "state": "rcvr-state-connecting",
          "comments": "Connection retries in progress",
          "profile": ""
        }
      ]
    }
  ]
}
```

CLI を使用した名前付きプロトコルレシーバの設定

手順の概要

1. **enable**
2. **configure terminal**
3. **telemetry receiver protocol** *receiver-name*
4. **protocol** {**cloud-native** | **cntp-tcp** | **cntp-tls profile** *profile-name* | **grpc-tcp** | **grpc-tls profile** *profile-name* | **native** | **tls-native profile** *profile-name*}
5. **host** {**ip** *ip-address* | **name** *hostname*} *receiver-port*
6. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**telemetry receiver protocol** *receiver-name* 例： `Device(config)# telemetry receiver protocol receiver1`	名前付きプロトコルレシーバを設定し、テレメトリ プロトコルレシーバ コンフィギュレーション モードを開始します。
ステップ 4	**protocol** {**cloud-native** | **cntp-tcp** | **cntp-tls profile** *profile-name* | **grpc-tcp** | **grpc-tls profile** *profile-name* | **native** | **tls-native profile** *profile-name*} 例： `Device(config-mdt-protocol-receiver)# protocol grpc-tcp`	名前付きプロトコルレシーバ接続のプロトコルを設定します。
ステップ 5	**host** {**ip** *ip-address* | **name** *hostname*} *receiver-port* 例： `Device(config-mdt-protocol-receiver)# host name rcvr.test.com 45000`	名前付きプロトコルレシーバのホスト名を設定します。
ステップ 6	**end** 例： `Device(config-mdt-protocol-receiver)# end`	テレメトリ プロトコルレシーバ コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

名前付きレシーバを使用したサブスクリプションの設定（CLI を使用）

手順の概要

1. **enable**
2. **configure terminal**
3. **telemetry ietf subscription** *id*
4. **receiver-type protocol** }
5. **receiver name** *name*
6. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**telemetry ietf subscription** *id* 例： `Device(config)# telemetry ietf subscription 101`	テレメトリのサブスクリプションを作成し、テレメトリサブスクリプション モードを開始します。
ステップ 4	**receiver-type protocol** } 例： `Device(config-mdt-subs)# receiver-type protocol`	プロトコルタイプレシーバを設定します。
ステップ 5	**receiver name** *name* 例： `Device(config-mdt-subs)# receiver name receiver1`	通知ためのレシーバの名前を設定します。
ステップ 6	**end** 例： `Device(config-mdt-subs)# end`	テレメトリのテレメトリサブスクリプションモードを終了し、特権 EXEC モードに戻ります。

モデル駆動型テレメトリに関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
NETCONF-YANG パッチ	https://tools.ietf.org/wg/netconf/draft-ietf-netconf-yang-patch/
YANG エクスプローラ	https://github.com/CiscoDevNet/yang-explorer

標準および RFC

標準/RFC	タイトル
イベント通知のカスタム サブスクリプション *draft-ietf-netconf-subscribed-notifications-03*	https://tools.ietf.org/id/draft-ietf-netconf-subscribed-notifications-03.txt
イベント通知の *NETCONF* サポート	draft-ietf-netconf-netconf-event-notifications-01

標準/RFC	タイトル
RFC 5277	NETCONF イベント通知
RFC 6241	ネットワーク設定プロトコル（NETCONF）
RFC 7950	YANG 1.1 データ モデリング言語
RFC 8040	RESTCONF プロトコル
イベント通知への登録	draft-ietf-netconf-rfc5277bis-01
YANG データストア プッシュのサブスクライブ	draft-ietf-netconf-yang-push-04
YANG データストア プッシュ更新のサブスクライブ *draft-ietf-netconf-yang-push-07*	https://tools.ietf.org/id/draft-ietf-netconf-yang-push-07.txt

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

モデル駆動型テレメトリの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *39 :* モデル駆動型テレメトリの機能情報

機能名	リリース	機能情報
モデル駆動型テレメトリ NETCONF ダイヤルイン	Cisco IOS XE Everest 16.6.1	モデル駆動型テレメトリでは、ネットワーク デバイスからサブスクライバに、リアルタイムの設定や運用状態の情報を継続的にストリームすることができます。 • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ
	Cisco IOS XE Everest 16.6.2	• Cisco Catalyst 9400 シリーズ スイッチ
	Cisco IOS XE Fuji 16.7.1	• Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ（ASR1001-HX、ASR1001-X、ASR1002-HX、ASR1002-X）
	Cisco IOS XE Fuji 16.8.1	• Cisco 1000 シリーズ サービス統合型ルータ • Cisco ASR 1000 RP2 および RP3 シリーズ アグリゲーション サービス ルータ
	Cisco IOS XE Fuji 16.8.1a	• Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ

機能名	リリース	機能情報
	Cisco IOS XE Fuji 16.9.1	• Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco cBR-8 コンバージド ブロードバンド ルータ • Cisco Network Convergence System 4200 シリーズ
	Cisco IOS XE Gibraltar 16.9.2	• Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300L SKU
	Cisco IOS XE Gibraltar 16.10.1	• Cisco クラウド サービス ルータ 1000v • Cisco Network Convergence System 520 シリーズ
	Cisco IOS XE Gibraltar 16.11.1	• Cisco Catalyst 9600 シリーズ スイッチ

機能名	リリース	機能情報
モデル駆動型テレメトリ gNMI ダイヤルイン	Cisco IOS XE Gibraltar 16.12.1	イニシエータ/サブスクライバに送信されるテレメトリの更新は、ダイヤルと呼ばれます。 この機能は、次のプラットフォームに実装されていました。 • Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300 および 9300L シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 および 9500 ハイ パフォーマンス シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ • Cisco cBR-8 コンバージド ブロードバンド ルータ
	Cisco IOS XE Amsterdam 17.1.1	• Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco Network Convergence System 520 シリーズ • Cisco Network Convergence System 4200 シリーズ
	Cisco IOS XE Amsterdam 17.2.1	Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ

機能名	リリース	機能情報
モデル駆動型テレメトリ gRPC ダイヤルアウト	Cisco IOS XE Gibraltar 16.10.1	

機能名	リリース	機能情報
		設置済みサブスクリプションでは、パブリッシャが受信者への接続を開始し、それらの接続はダイヤルアウトと見なされます。 この機能は、次のプラットフォームに実装されていました。 • Cisco 1000 シリーズ サービス統合型ルータ • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300 および 9300L シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 および 9500 ハイ パフォーマンス シリーズ スイッチ • Cisco cBR-8 コンバージド ブロードバンド ルータ • Cisco Cloud Services Router 1000V シリーズ • Cisco Network Convergence System 520 シリーズ

機能名	リリース	機能情報
		• Cisco Network Convergence System 4200 シリーズ
	Cisco IOS XE Gibraltar 16.11.1	• Cisco Catalyst 9600 シリーズ スイッチ
モデル駆動型テレメトリ：サブスクリプションの kill	Cisco IOS XE Gibraltar 16.11.1	動的サブスクリプションを削除するには、CLI および kill-subscription RPC を使用できます。 • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ（RSP2） • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ • Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300 および 9300L シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 および 9500 ハイ パフォーマンス シリーズ スイッチ • Cisco Network Convergence System 520 シリーズ • Cisco Network Convergence System 4200 シリーズ

機能名	リリース	機能情報
TLDP 変更時の通知	Cisco IOS XE Amsterdam 17.2.1	TLDP 変更時の通知機能は、TLDPセッションが起動または停止したとき、および TLDP が設定またはディセーブルになったときにユーザに通知します。 この機能は、次のプラットフォームに実装されていました。 • Cisco 4000 シリーズ サービス統合型ルータ • Cisco Catalyst 9200 シリーズ スイッチ • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ

機能名	リリース	機能情報
GRPCダイヤルアウト用のTLS	Cisco IOS XE Amsterdam 17.1.1	

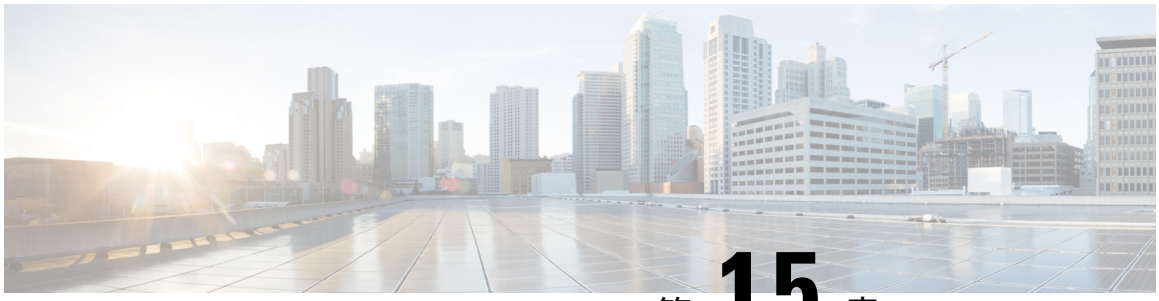
機能名	リリース	機能情報
		トランスポート層セキュリティは、gRPCダイヤルアウトでサポートされます。この機能は、次のプラットフォームでサポートされます。 • Cisco 1000 シリーズ サービス統合型ルータ • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300 および 9300L シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 および 9500 ハイ パフォーマンス シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ • Cisco Catalyst 9800-40 シリーズ ワイヤレス コントローラ • Cisco Catalyst 9800-80 シリーズ ワイヤレス コントローラ • Cisco cBR-8 コンバージド

機能名	リリース	機能情報
		ブロードバンド ルータ • Cisco Cloud Services Router 1000V シリーズ • Cisco Network Convergence System 520 シリーズ • Cisco Network Convergence System 4200 シリーズ

機能名	リリース	機能情報
gRPC サブスクリプションの FQDN サポート	Cisco IOS XE Bengaluru 17.6.1	

機能名	リリース	機能情報
		gRPCサブスクリプション機能の FQDN サポートの導入により、IPアドレスに加え、FQDN も gRPC サブスクリプションに使用できます。 • Cisco 1000 シリーズ サービス統合型ルータ • Cisco 4000 シリーズ サービス統合型ルータ • Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ • Cisco ASR 900 シリーズ アグリゲーション サービス ルータ • Cisco ASR 920 シリーズ アグリゲーション サービス ルータ • Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300 および 9300L シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ • Cisco Catalyst 9500 および 9500 ハイ パフォーマンス シリーズ スイッチ • Cisco Catalyst 9600 シリーズ スイッチ • Cisco Catalyst 9800-40 シリーズ ワイヤレス コントローラ • Cisco Catalyst 9800-80 シリーズ ワイヤレス コントローラ • Cisco cBR-8 コンバージド

機能名	リリース	機能情報
		ブロードバンド ルータ • Cisco Cloud Services Router 1000V シリーズ • Cisco Network Convergence System 520 シリーズ • Cisco Network Convergence System 4200 シリーズ



第 15 章

In-Service Model Update

このモジュールでは、In-Service Model Update によりデバイス上の YANG データ モデルを更新する方法を説明します。

In-Service Model Update の制約事項

- ハイ アベイラビリティまたは In-Service Software Upgrade（ISSU）はサポートされていません。スイッチオーバーの後、ユーザはスタンバイ デバイスにソフトウェア メンテナンス アップデート（SMU）をインストールする必要があります。

In-Service Model Update について

In-Service Model Update の概要

サービス中モデル更新プログラムは、既存のデータモデルに新しいデータ モデルまたは拡張機能を追加します。サービス中モデル更新プログラムは、リリース サイクル外の YANG モデルの拡張機能を提供します。更新プログラム パッケージはすべての既存のモデルの上位セットです。これには、更新された YANG モデルを始めとするすべての既存モデルが含まれています。

データ モデル インフラストラクチャは、Cisco IOS XE デバイス用の YANG モデル定義管理インターフェイスを実装します。データ モデル インフラストラクチャは、Cisco IOS XE デバイスからノースバウンドに NETCONF インターフェイスを公開します。サポートされているデータ モデルには、IETF などの業界標準モデルと、Cisco IOS XE デバイス固有のモデルが含まれます。

In-Service Model Update によって提供される機能は、その後の Cisco IOS XE ソフトウェア メンテナンス リリースに統合されます。データ モデル更新プログラム パッケージは、シスコ ソフトウェア ダウンロード センター からダウンロードできます。

In-Service Model Update パッケージの互換性

更新パッケージは、リリース単位で作成され、プラットフォームに固有になります。たとえば、Cisco ASR 1000 シリーズ アグリゲーション サービス ルータの更新パッケージを Cisco CSR 1000V シリーズ クラウド サービス ルータにインストールすることはできません。同様に、Cisco IOS XE Fuji 16.7.1 用に作成された更新パッケージを、Cisco IOS XE Everest 16.5.2 バージョンを実行しているデバイスに適用することはできません。

更新プログラム パッケージのすべてのコンテンツは、将来のメインライン リリースまたはメンテナンス リリースのイメージの一部になります。イメージとプラットフォームのバージョンは、パッケージの追加およびアクティブ化の際に、In-Service Model Update コマンドによってチェックされます。イメージまたはプラットフォームの不一致が発生すると、パッケージのインストールが失敗します。

更新プログラム パッケージの命名規則

In-Service Model Update は、.bin ファイルとしてパッケージ化されています。このファイルには、特定のリリースおよびプラットフォームのすべての更新プログラムと、Readme ファイルが含まれています。これらのファイルにはリリース日があり、追加モデルの更新をともなって定期的に更新されます。

データ モデルの更新プログラム パッケージの命名規則は、次の形式に従っています。プラットフォームの種類-ライセンス レベル.リリース バージョン.DDTS ID-ファイル。次に、データ モデル更新ファイルの例を示します。

- isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
- asr1000-universalk9.2017-08-23_17.48.0.CSCxxxxxxx.SSA.dmp.bin

Readme ファイルは、次の情報を提供します。

- データ モデルのアクティブ化または非アクティブ化中に表示されるコンソール メッセージおよびエラー メッセージ
- データ モデルのインストールによる影響
- 副作用と考えられる回避策
- In-Service Model Update によって影響を受けるパッケージ
- リスタートのタイプ

更新プログラム パッケージのインストール

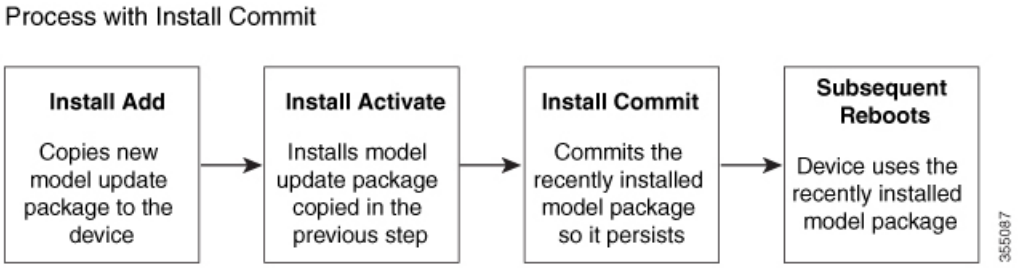
デバイスに In-Service Model Update パッケージをインストールするには、**install add**、 **install activate**、および **install commit** コマンドを特権 EXEC モードで使用します。

install add コマンドは、更新パッケージをリモートの場所からデバイスにコピーします。パッケージをコピーするには他の方法も使用できますが、その場合も、インストールしたプログラムを動作させるために **install add** コマンドを有効化する必要があります。 **install activate** コマンドを動作させるには、パッケージをデバイスのブートフラッシュで使用可能にする必要があります。**install commit** コマンドを有効化して、更新プログラムをリロード全体にわたって確定します。

更新プログラムをインストールすると、以前にインストールされたデータ モデルがある場合、それは置き換えられます。デバイスには常に、1 つの更新プログラムのみがインストールされます。データ モデル パッケージには、すべての更新された YANG モデルと、以前にデバイスにインストールされたすべての既存 YANG モデルが含まれています。

次のフロー チャートでは、モデル更新プログラム パッケージの動作を説明します。

図 *9*：モデル更新プログラム パッケージのコミット



パッケージをアクティブ化する際にNETCONF-YANGが有効化されていると、NETCONFプロセスがリスタートされます。すべてのアクティブなNETCONFセッションは、パッケージのアクティブ化中に破棄されます。パッケージの検証中にエラーが発生すると、アクティブ化プロセスは終了します。

更新プログラム パッケージの非アクティブ化

更新パッケージを非アクティブ化するには、**install deactivate** コマンドを使用します。変更を確定するには、**install commit** コマンドを有効化します。

表 *40*：モデル更新プログラム パッケージの非アクティブ化

操作	使用コマンド
パッケージの削除	**install remove** コマンドを使用します。 （注） パッケージを削除する前に非アクティブ化します。

操作	使用コマンド
パッケージの非アクティブ化	**install deactivate** コマンドを使用し、その後に **install commit** コマンドを使用します。 （注） **install commit** コマンドの使用が必要なのは、モデルパッケージの非アクティブ化をリロード全体にわたって確定するためです。非アクティブ化がコミットされていないと、その後にパッケージを削除しようとしても失敗します。

更新プログラムを非アクティブ化する際に、2つ以上のモデル更新プログラム パッケージがインストールされている場合、最近コミットされたモデル更新プログラム パッケージがデバイスによって使用されるモデル パッケージになります。以前にコミットされたその他のモデル パッケージがない場合、標準的なイメージとともに含まれているベース バージョンのデータ モデルが使用されるようになります。

更新プログラム パッケージのロールバック

ロールバックは、デバイスを更新前の動作状態に戻すメカニズムを提供します。ロールバック後は、変更が表示されるようになる前に NETCONF-YANG プロセスが再始動します。

更新は、**install rollback** コマンドを使用して、基本バージョン、最終コミット バージョン、または既知のコミット ID までロールバックできます。

In-Service Model Update の管理方法

更新プログラム パッケージの管理

手順の概要

1. **enable**
2. **install add file tftp:** *filename*
3. **install activate file bootflash:** *filename*
4. **install commit**
5. **install deactivate file bootflash:** *filename*
6. **install commit**
7. **install rollback to** {**base** | **committed** | **id** *commit-ID*}
8. **install remove** {**file bootflash:** *filename* | **inactive**}
9. **show install summary**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。

ステップ 2	**install add file tftp:** *filename*
	例：
	`Device# install add file tftp://172.16.0.1//tftpboot/folder1/ isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin`
	`Device# install add file tftp://172.16.0.1//tftpboot/folder1/ asr1000-universalk9.2017-08-23_17.48.0.CSCxxxxxxx.SSA.dmp.bin`
		リモート ロケーションから（FTP、TFTP 経由で）デバイスにモデル更新プログラム パッケージをコピーし、プラットフォームとイメージのバージョンの互換性チェックを実行します。
		• 他の方法を使用してリモートの場所からデバイスに更新パッケージをコピーすることもできます。ただし、その場合もパッケージをアクティブにする前に **install add** コマンドを実行する必要があります。

ステップ 3	**install activate file bootflash:** *filename*
	例：
	`Device# install activate file bootflash: isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin`
	`Device# install activate file bootflash: asr1000-universalk9.2017-08-23_17.48.0.CSCxxxxxxx.SSA.dmp.bin`
		更新パッケージが **install add** コマンドにより追加されていることを確認し、NETCONF プロセスを再開します。
		• 更新パッケージをアクティブにする前に **install add** 操作を実行します。

ステップ 4	**install commit** 例： `Device# install commit`	リロードが繰り返されても持続する変更を行います。 • NETCONF プロセスは再開されません。

ステップ 5	**install deactivate file bootflash:** *filename*
	例：
	`Device# install deactivate file bootflash: isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin`
	`Device# install deactivate file bootflash: asr1000-universalk9.2017-08-23_17.48.0.CSCxxxxxxx.SSA.dmp.bin`
		指定された更新プログラムパッケージを非アクティブにして、NETCONF プロセスを再開します。

ステップ 6	**install commit** 例： `Device# install commit`	リロードが繰り返されても持続する変更を行います。 • NETCONF プロセスは再開されません。
ステップ 7	**install rollback to** {**base** | **committed** | **id** *commit-ID*} 例： `Device# install rollback to base`	更新を基本バージョン、最後にコミットしたバージョン、または既知のコミット ID にロールバックし、NETCONF プロセスを再起動します。 • *commit-id* 引数の有効な値は 1 ～ 4294967295 です。

	コマンドまたはアクション	目的
		• データ モデル更新の古いバージョンが使用可能です。

ステップ 8	**install remove** {**file bootflash:** *filename* | **inactive**}
	例：
	`Device# install remove file bootflash: isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin`
	`Device# install remove file bootflash: asr1000-universalk9.2017-08-23_17.48.0.CSCxxxxxxx.SSA.dmp.bin`
		指定された更新プログラム パッケージをブートフラッシュから削除します。
		• パッケージは削除する前に非アクティブにする必要があります。

ステップ 9	**show install summary** 例： `Device# show install summary`	アクティブパッケージに関する情報を表示します。 • このコマンドの出力は、設定されている **install** コマンドに応じて変化します。

In-Service Model Update の設定例

例：更新プログラム パッケージの管理

次の例で使用しているのは、Cisco 4000 シリーズ サービス統合型ルータのサンプル イメージです。

次の例では、モデル更新プログラム パッケージ ファイルの追加方法を示しています。

```
Device# install add file tftp://172.16.0.1//tftpboot/folder1/
isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin

install_add: START Sun Feb 26 05:57:04 UTC 2017
Downloading file
tftp://172.16.0.1//tftpboot/folder1/isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Finished downloading file
tftp://172.16.0.1//tftpboot/folder1/isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
to bootflash:isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
SUCCESS: install_add /bootflash/isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Sun Feb 26 05:57:22 UTC 2017
Device#
```

次の例で使用しているのは、Cisco ASR1000 シリーズ アグリゲーション サービス ルータのサンプル イメージです。

次の例では、モデル更新プログラム パッケージ ファイルの追加方法を示しています。

```
Device# install add file tftp://172.16.0.1//tftpboot/folder1/
asr1000-universalk9.2017-08-23_17.48.0.CSCxxxxxxx.SSA.dmp.bin

install_add: START Sun Feb 26 05:57:04 UTC 2017
Downloading file
tftp://172.16.0.1//tftpboot/folder1/asr1000-universalk9.2017-08-23_17.48.0.CSCxxxxxxx.SSA.dmp.bin
Finished downloading file
```

```
tftp://172.16.0.1//tftpboot/folder1/asr1000-universalk9.2017-08-23_17.48.0.CSCxxxxxxx.SSA.dmp.bin
to bootflash: asr1000-universalk9.2017-08-23_17.48.0.CSCxxxxxxx.SSA.dmp.bin
SUCCESS: install_add
/bootflash/asr1000-universalk9.2017-08-23_17.48.0.CSCxxxxxxx.SSA.dmp.bin
Sun Feb 26 05:57:22 UTC 2017
Device#
```

次に、更新パッケージ ファイルをデバイスに追加した後の **show install summary** コマンドの出力例を示します。

```
Device# show install summary

Active Packages:
No packages
Inactive Packages:
bootflash: isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Committed Packages:
No packages
Uncommitted Packages:
No packages
Device#
```

次の例では、追加された更新プログラム パッケージ ファイルをアクティブにする方法を示しています。

```
Device# install activate file bootflash:
isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin

install_activate: START Sun Feb 26 05:58:41 UTC 2017
DMP package.
Netconf processes stopped
SUCCESS: install_activate /bootflash/isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Sun Feb 26 05:58:58 UTC 2017*Feb 26 05:58:47.655: %DMI-4-CONTROL_SOCKET_CLOSED:
SIP0: nesd: Confd control socket closed Lost connection to ConfD (45): EOF on socket to
 ConfD.
*Feb 26 05:58:47.661: %DMI-4-SUB_READ_FAIL: SIP0: vtyserverutild:
Confd subscription socket read failed Lost connection to ConfD (45):
EOF on socket to ConfD.
*Feb 26 05:58:47.667: %DMI-4-CONTROL_SOCKET_CLOSED: SIP0: syncfd:
Confd control socket closed Lost connection to ConfD (45): EOF on socket to ConfD.
*Feb 26 05:59:43.269: %DMI-5-SYNC_START: SIP0: syncfd:
External change to running configuration detected.
The running configuration will be synchronized to the NETCONF running data store.
*Feb 26 05:59:44.624: %DMI-5-SYNC_COMPLETE: SIP0: syncfd:
The running configuration has been synchronized to the NETCONF running data store.
Device#
```

次に示すのは、**show install summary** コマンドがモデル パッケージのステータスをアクティブでありコミット未完了と表示する場合の出力例です。

```
Device# show install summary

Active Packages:
bootflash:isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Inactive Packages:
No packages
Committed Packages:
No packages
Uncommitted Packages:
bootflash:isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Device#
```

次の例では、**install commit** コマンドの実行方法を示しています。

```
Device# install commit

install_commit: START Sun Feb 26 06:46:48 UTC 2017
SUCCESS: install_commit Sun Feb 26 06:46:52 UTC 2017
Device#
```

次に示すのは、**show install summary** コマンドが、更新パッケージがコミットされてリロードが繰り返されても持続することを表示する場合の出力例です。

```
Device# show install summary

Active Packages:
bootflash:isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Inactive Packages:
No packages
Committed Packages:
bootflash:isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Uncommitted Packages:
No packages
Device#
```

次の例は、更新プログラム パッケージを基本パッケージにロールバックする方法を示しています。

```
Device# install rollback to base

install_rollback: START Sun Feb 26 06:50:29 UTC 2017
7 install_rollback: Restarting impacted processes to take effect
7 install_rollback: restarting confd
*Feb 26 06:50:34.957: %DMI-4-CONTROL_SOCKET_CLOSED: SIP0: syncfd:
Confd control socket closed Lost connection to ConfD (45): EOF on socket to ConfD.
*Feb 26 06:50:34.962: %DMI-4-CONTROL_SOCKET_CLOSED: SIP0: nesd:
Confd control socket closed Lost connection to ConfD (45): EOF on socket to ConfD.
*Feb 26 06:50:34.963: %DMI-4-SUB_READ_FAIL: SIP0: vtyserverutild:
Confd subscription socket read failed Lost connection to ConfD (45):
EOF on socket to ConfD.Netconf processes stopped
7 install_rollback: DMP activate complete
SUCCESS: install_rollback Sun Feb 26 06:50:41 UTC 2017
*Feb 26 06:51:28.901: %DMI-5-SYNC_START: SIP0: syncfd:
External change to running configuration detected.
The running configuration will be synchronized to the NETCONF running data store.
*Feb 26 06:51:30.339: %DMI-5-SYNC_COMPLETE: SIP0: syncfd:
The running configuration has been synchronized to the NETCONF running data store.
Device#
```

次に、**show install package** コマンドの出力例を示します。

```
Device# show install package bootflash:
isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin

Name: isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin
Version: 16.5.1.0.199.1484082952..Everest
Platform: ISR4300
Package Type: dmp
Defect ID: CSCxxxxxxx
Package State: Added
Supersedes List: {}
Smu ID: 1
```

```
Device#
```

次の NETCONF hello メッセージの例では、新規データ モデル パッケージのバージョンを確認します。

```
Getting Capabilities: (admin @ 172.16.0.1:830)
PROTOCOL netconf
<?xml version="1.0" encoding="UTF-8"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<capabilities>
<capability>urn:ietf:params:netconf:base:1.0</capability>
<capability>urn:ietf:params:netconf:base:1.1</capability>
<capability>urn:ietf:params:netconf:capability:writable-running:1.0</capability>
<capability>urn:ietf:params:netconf:capability:xpath:1.0</capability>
<capability>urn:ietf:params:netconf:capability:validate:1.0</capability>
<capability>urn:ietf:params:netconf:capability:validate:1.1</capability>
<capability>urn:ietf:params:netconf:capability:rollback-on-error:1.0</capability>
<capability>urn:ietf:params:netconf:capability:notification:1.0</capability>
<capability>urn:ietf:params:netconf:capability:interleave:1.0</capability>
<capability>http://tail-f.com/ns/netconf/actions/1.0</capability>
<capability>http://tail-f.com/ns/netconf/extensions</capability>
<capability>urn:ietf:params:netconf:capability:with-defaults:1.0?basic-mode=
explicit&also-supported=report-all-tagged</capability>
<capability>urn:ietf:params:xml:ns:yang:ietf-netconf-with-defaults?
revision=2011-06-01&module=ietf-netconf-with-defaults</capability>
<capability>http://cisco.com/ns/yang/Cisco-IOS-XE-aaa?module=
Cisco-IOS-XE-aaa&revision=2017-02-07</capability>
<<capability>http://cisco.com/ns/yang/Cisco-IOS-XE-native?module=
Cisco-IOS-XE-native&revision=2017-01-07&features=virtual-
template,punt-num,multilink,eth-evc,esmc,efp,dot1x</capability>
Device#
```

次に、**show install log** コマンドの出力例を示します。

```
Device# show install log

[0|install_op_boot]: START Fri Feb 24 19:20:19 Universal 2017
[0|install_op_boot]: END SUCCESS Fri Feb 24 19:20:23 Universal 2017
[3|install_add]: START Sun Feb 26 05:55:31 UTC 2017
[3|install_add( FATAL)]: File path (scp) is not yet supported for this command
[4|install_add]: START Sun Feb 26 05:57:04 UTC 2017
[4|install_add]: END SUCCESS /bootflash/isr4300-universalk9.16.05.01.CSCxxxxxxx.dmp.bin

Sun Feb 26 05:57:22 UTC 2017
[5|install_activate]: START Sun Feb 26 05:58:41 UTC 2017
Device#
```

次の例で使用しているのは、Cisco Catalyst 3000 シリーズ スイッチのサンプル イメージです。

次の例では、モデル更新プログラム パッケージ ファイルの追加方法を示しています。

```
Device# install add file tftp://172.16.0.1//tftpboot/folder1/
cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.dmp.bin

install_add: START Sat Jul 29 05:57:04 UTC 2017
Downloading file tftp://172.16.0.1//tftpboot/folder1/
cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.dmp.bin
Finished downloading file tftp://172.16.0.1//tftpboot/folder1/
cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.SPA.smu.bin
```

```
to bootflash:cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.dmp.bin
SUCCESS: install_add /bootflash/cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.dmp.bin
Sat Jul 29 05:57:22 UTC 2017
Device#
```

次に示すのは、**show install summary** コマンドが、更新パッケージがコミットされてリロードが繰り返されても持続することを表示する場合の出力例です。

```
Device# show install summary

Active Packages:
bootflash:cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.dmp.bin
Inactive Packages:
No packages
Committed Packages:
bootflash:cat3k_caa-universalk9.16.06.01.CSCxxxxxxx.dmp.bin
Uncommitted Packages:
No packages
Device#
```

In-Service Model Update の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

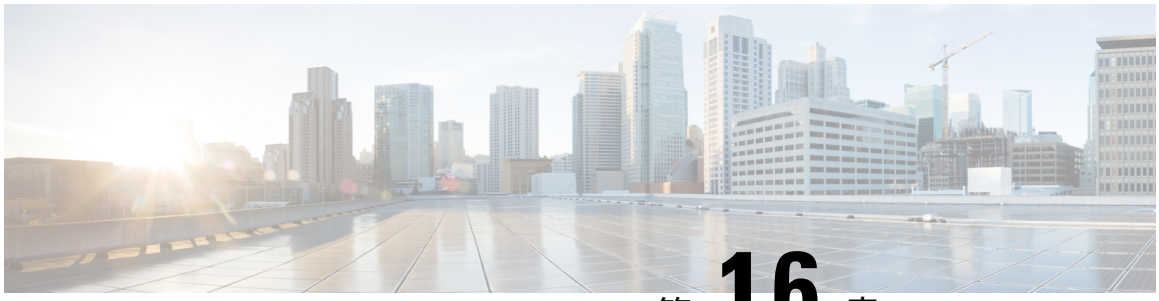
表 *41 : In-Service Model Update* の機能情報

機能名	リリース	機能情報
In-Service Model Update	Cisco IOS XE Everest 16.5.1a Cisco IOS XE Everest 16.5.1b	このモジュールでは、In-Service Model Update で YANG データ モデルを更新する方法を説明します。 Cisco IOS XE Everest 16.5.1a では、この機能は次のプラットフォームに実装されていました。 • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ Cisco IOS XE Everest 16.5.1b では、この機能は次のプラットフォームに実装されていました。 • Cisco 4000 シリーズ サービス統合型ルータ • Cisco クラウド サービス ルータ 1000v • Cisco サービス統合型仮想ルータ（ISRv） コマンド **install (Programmability)**、**show install (Programmability)** が導入または更新されました。
	Cisco IOS XE Everest 16.6.1	Cisco IOS XE Everest 16.5.1b では、この機能は次のプラットフォームに実装されていました。 • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ
	Cisco IOS XE Fuji 16.7.x	Cisco IOS XE Fuji 16.7.x では、この機能は次のプラットフォームに実装されていました。 • Cisco 1000 シリーズ アグリゲーション サービス ルータ

機能名	リリース	機能情報
	Cisco IOS XE Fuji 16.8.1a	Cisco IOS XE Fuji 16.8.1a では、この機能は Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチに実装されていました。

第 IV 部

アプリケーション ホスティング



第 16 章

アプリケーション ホスティング

ホステッドアプリケーションは Software as a Service（SaaS）ソリューションであり、コマンドを使用してリモート実行できます。アプリケーションのホスティングによって、管理者には独自のツールやユーティリティを利用するためのプラットフォームが与えられます。

（注） アプリケーション ホスティングは Docker アプリケーションのみをサポートします。

このモジュールでは、アプリケーションホスティング機能とその有効化の方法について説明します。

アプリケーション ホスティングの制約事項

- アプリケーション ホスティングは、Virtual Routing and Forwarding 認識（VRF 認識）ではありません。
- Cisco IOS XE Amsterdam 17.3.3 以前のリリースでは、アプリケーション ホスティングには専用ストレージの割り当てが必要であり、ブートフラッシュでは無効になっています。

  Cisco IOS XE Amsterdam 17.3.3 以降のリリースでは、アプリケーション ホスティングはブートフラッシュで有効ですが、シスコ署名済みアプリケーションのみがホストされます。
- 前面パネルの Universal Serial Bus（USB）スティックはサポートされていません。

Cisco Catalyst 9300 シリーズ スイッチは、背面パネルの シスコ認定 USB のみをサポートします。

- Cisco Catalyst 9500-High Performance シリーズ スイッチおよび Cisco Catalyst 9600 シリーズ スイッチは、前面パネルの USB のアプリケーション ホスティングをサポートしていません。
- Cisco Catalyst 9500 および 9500 ハイ パフォーマンス シリーズ スイッチ、および Cisco Catalyst 9600 シリーズ スイッチでは、AppGigabitEthernet インターフェイスはサポートされません。
- Cisco IOS XE Bengaluru 17.5.1 より前のリリースでは、Cisco Catalyst 9410R スイッチはアプリケーション ホスティングをサポートしていません。

  Cisco Catalyst 9410R スイッチでアプリケーション ホスティングを有効にするには、AppGigabitEthernetインターフェイスで **enable** コマンドを設定します。

アプリケーション ホスティングに関する情報

ここでは、アプリケーション ホスティングについて説明します。

アプリケーション ホスティングの必要性

仮想環境への移行により、再利用可能なポータブルかつスケーラブルなアプリケーションを構築する必要性が高まりました。アプリケーションのホスティングによって、管理者には独自のツールやユーティリティを利用するためのプラットフォームが与えられます。ネットワークデバイスでホスティングされているアプリケーションは、さまざまな用途に利用できます。これは、既存のツールのチェーンによる自動化から、設定管理のモニタリング、統合に及びます。

（注） このドキュメントでは、コンテナは Docker アプリケーションを指します。

Cisco IOx の概要

Cisco IOx（IOs＋linuX）はエンドツーエンドアプリケーションフレームワークであり、Cisco ネットワーク プラットフォーム上のさまざまなタイプのアプリケーションに対し、アプリケーション ホスティング機能を提供します。Cisco ゲストシェルは特殊なコンテナ展開であり、システムの開発に役立つアプリケーションの 1 つです。

Cisco IOx は、構築済みアプリケーションをパッケージ化し、それらをターゲットデバイス上にホストする開発者の作業を支援する一連のサービスを提供することにより、アプリケーションのライフサイクル管理とデータ交換を容易にします。IOxのライフサイクル管理には、アプリケーションおよびデータの配布、展開、ホスティング、開始、停止（管理）、およびモニタ

が含まれます。IOx サービスにはアプリケーションの配布および管理ツールも含まれており、ユーザがアプリケーションを発見して IOx フレームワークに展開するのに役立ちます。

Cisco IOx アプリケーション ホスティングは、次の機能を提供します。

- ネットワークの不均質性の遮蔽。
- デバイス上にホストされているアプリケーションのライフサイクルをリモートで管理する Cisco IOx アプリケーション プログラミング インターフェイス（API）。
- 一元化されたアプリケーションのライフ サイクル管理。
- クラウド ベースの開発。

アプリケーション ホスティングの概要

シスコのアプリケーション ホスティング フレームワークは、デバイス上で実行される仮想化アプリケーションやコンテナアプリケーションを管理する、IOx の Python プロセスです。

アプリケーション ホスティングは、次のサービスを提供します。

- コンテナ内の指定されたアプリケーションを起動する。
- 使用可能なリソース（メモリ、CPU、およびストレージ）を確認し、それらを割り当て、管理する。
- コンソール ロギングのサポートを提供する。
- REST API を介してサービスへのアクセスを提供する。
- CLI エンドポイントを提供する。
- Cisco Application Framework（CAF）と呼ばれるアプリケーション ホスティング インフラストラクチャを提供する。
- 管理インターフェイスを介したプラットフォーム固有のネットワーキング（パケットパス）のセットアップを支援する。

  データポートは、AppGigabitEthernet ポート機能を備えたプラットフォームでサポートされます。

アプリケーション ホスティングのコンテナは、ホスト オペレーティング システムでゲスト アプリケーションを実行するために提供される仮想環境と呼ばれています。Cisco IOS XE 仮想化サービスは、ゲストアプリケーションを実行するための管理性とネットワーキングモデルを提供します。仮想化インフラストラクチャにより、管理者はホストとゲスト間の接続を指定する論理インターフェイスを定義できます。Cisco IOx は、論理インターフェイスをゲストアプリケーションが使用する仮想ネットワーク インターフェイス カード（vNIC）にマッピングします。

コンテナに展開されるアプリケーションは、TAR ファイルとしてパッケージ化されます。これらのアプリケーションに固有の設定は、TAR ファイルの一部としてもパッケージ化されています。

デバイス上の管理インターフェイスは、アプリケーション ホスティング ネットワークを Cisco IOS 管理インターフェイスに接続します。ゲストアプリケーションのレイヤ 3 インターフェイスは、Cisco IOS 管理インターフェイスからレイヤ 2 ブリッジトラフィックを受信します。管理インターフェイスは、管理ブリッジを使用してコンテナインターフェイスに接続します。IP アドレスは、管理インターフェイス IP アドレスと同じサブネット上にある必要があります。

（注） すべての Cisco Catalyst スタックおよび StackWise 仮想モデル（すべてのソフトウェアバージョン）で、ゲストシェルおよび AppGigabitEthernet インターフェイスはスタック内のアクティブスイッチでのみ動作します。したがって、AppGigabitEthernet インターフェイスの設定は、スタック内のすべてのスイッチの AppGigabitEthernet インターフェイスに適用する必要があります。この設定が適用されていないスイッチがある場合、スイッチオーバー後にそのスイッチでは AppGigabitEthernet インターフェイスが機能しません。

Cisco Catalyst 9000 シリーズスイッチは、アプリケーションが SSDでホストされている場合、複数のアプリケーションをサポートします。アプリケーションは、次の条件を満たす必要があります。

- シスコの署名がある。
- 次のスイッチング インフラストラクチャ要件を満たしている。
  - AppGigabitEthernet ポート上のネットワーク設定で、アプリケーション間の競合が発生しない。
  - アプリケーションを実行するのに十分なリソースがある。

1 つのアプリケーションが使用可能なすべてのアプリケーション ホスティング リソースを消費する場合、複数のアプリケーションを導入することはできません。たとえば、1 つのアプリケーションがすべてのコンピューティングリソースとランタイムリソースを消費している場合、他のアプリケーションはデバイスにインストールできなくなります。

前面パネルトランクおよび VLAN ポートのアプリケーション ホスティング

アプリケーション ホスティングでは前面パネル VLAN ポートおよびトランクポートがサポートされています。レイヤ 2 トラフィックは、これらのポートを介して、Cisco IOS デーモンの外部で動作するソフトウェアコンポーネントに配信されます。

アプリケーション ホスティングの場合、前面パネルポートをトランクインターフェイスまたは VLAN 固有のインターフェイスとして設定できます。トランクインターフェイスとして使用する場合、前面パネルポートはレイヤ 2 トランクポートとして機能するように拡張され、ポートで受信したすべてのトラフィックがアプリケーションで使用可能になります。ポートを VLAN インターフェイスとして使用する場合、アプリケーションは特定の VLAN ネットワークに接続されます。

（注） 背面パネルの USB または M2 SATA ドライブをアプリケーション ホスティングに使用する場合、ストレージメディアは *ext4* ファイルシステムとしてフォーマットする必要があります。

Cisco Catalyst 9300 シリーズ スイッチ のアプリケーション ホスティング

ここでは、Cisco Catalyst 9300 シリーズ スイッチでのアプリケーション ホスティングについて説明します。

アプリケーション ホスティングの場合、Cisco Catalyst 9300 シリーズ スイッチは管理インターフェイスと前面パネルポートをサポートします。

USB 3.0 SSD は Cisco Catalyst 9300 シリーズ スイッチで有効になっています。USB 3.0 SSD は、アプリケーションをホストするための追加の 120 GB ストレージを提供します。詳細については、『*Interfaces and Hardware Configuration Guide*』の「Configuring USB 3.0 SSD」の章を参照してください。

次の 2 種類のネットワーキング アプリケーションがサポートされています。

- コントロールプレーン：管理インターフェイスにアクセスするアプリケーション。
- データプレーン：前面パネルのポートにアクセスするアプリケーション。

Cisco Catalyst 9300X シリーズ スイッチの前面パネル アプリケーション ホスティング

Cisco IOS XE Bengaluru 17.6.1 では、前面パネルのアプリケーション ホスティングが Cisco Catalyst 9300X シリーズ スイッチで有効になっています。

アプリケーションは、ホスティングに専用の前面パネルポートを使用できます。**app-vnic AppGigabitEthernet port** コマンドを使用して、アプリケーション ホスティングに使用するポートを指定します。両方の前面パネルポートを同じレイヤ2アプリケーションに接続できます。

これらのスイッチは、アクセスモードとトランクモードの両方でアプリケーション ホスティングをサポートします。アプリケーション ホスティングは、両方のモードで同時に有効にできます。

（注） **app-vnic** コマンドで行われた設定は、アクティベーション中に拒否される可能性があります。

表 *42 :* アクセスモードとトランクモードでのアプリケーション ホスティングの設定シナリオ例

シナリオ	サポート対象/非サポート対象
単一のアプリケーションとアクセスモードの 2 つの前面パネルポート。	サポート。 重複する VLAN はありません。
単一のアプリケーションとトランクモードの 2 つの前面パネル。	サポート。 重複する VLAN はありません。
単一のアプリケーションとトランクモードおよびアクセスモードの 2 つの前面パネルポート。	サポート。 重複する VLAN はありません。
単一のアプリケーションと、デフォルトのアプリケーション ゲートウェイが設定されたトランクモードの 2 つの前面パネルポート。	サポート。 同一のアプリケーションと2つのインターフェイスが異なるサブネットに設定されていまが、デフォルトゲートウェイは、外部接続を持つ 1 つの VLAN に接続されます。
単一のアプリケーションと、VLAN が重複しているトランクモードとアクセスモードの 2 つの前面パネル。	有効な設定ではありません。 VLAN が両方のポートで重複しています。
アクセスモードでの単一アプリケーションと、同一 VLAN に設定された 2 つの前面パネルポート。	有効な設定ではありません。
トランクモードでの単一アプリケーションと、重複する VLAN 範囲で設定された 2 つの前面パネルポート。	有効な設定ではありません。 トラフィックが分離されていおらず、VLAN 範囲が重複しています。
トランクモードでの単一アプリケーションと、重複する VLAN 範囲で設定された 2 つの前面パネルポート。	有効な設定ではありません。 この設定はアクティベーション中に拒否されます。 両方の前面パネルポートがトランクモードであるため、任意の VLAN を使用できます。ただし、両方のポートに同じ VLAN が設定されているため、VLAN は両方のポートで重複します。 （注） 同じシナリオがアクセスモードにも適用されます。

シナリオ	サポート対象/非サポート対象
トランクモードとアクセスモードでの単一のアプリケーション、および VLAN が重複している前面パネルポート。	有効な設定ではありません。 トランクモードとアクセスモードで同じVLANが設定されています。この設定により、VLANは両方のポートで重複しています。
トランクモードでの複数のアプリケーション。	有効な設定ではありません。 トラフィックが分離されません。
トランクモードとアクセスモードでの 2 つのアプリケーション。	有効な設定ではありません。 VLAN が重複しています。

Cisco Catalyst 9300X シリーズ スイッチのハイアベイラビリティ

Cisco Catalyst 9300X シリーズ スイッチで使用可能な混合モードスタックでは、アクティブデバイスとスタンバイデバイスが、アプリケーション ホスティング用に 1 + 1 冗長性を使用します。混合モードのサポートとは、異なるモデルバリアントと異なるネットワークモジュールがスタックで使用される場合です。

Cisco Catalyst 9300X シリーズ スイッチと Cisco Catalyst 9300 シリーズ スイッチをスタックすると、Cisco Catalyst 9300X シリーズ スイッチの 2 つの前面パネルポートのいずれか 1 つが動的に無効になります。AppGigabitEthernet 1/0/1 インターフェイスのみが有効として表示されます。

このセクションでは、いくつかのハイアベイラビリティのシナリオについて説明します。

スタックモード	機能	使用されるポート	動作
Cisco Catalyst 9300X シリーズスイッチ（アクティブ） + Cisco Catalyst 9300X シリーズスイッチ（スタンバイ）	2	1	サポート対象
Cisco Catalyst 9300X シリーズスイッチ（アクティブ） + Cisco Catalyst 9300X シリーズスイッチ（スタンバイ）	1	1	サポート対象

スタックモード	機能	使用されるポート	動作
Cisco Catalyst 9300X シリーズスイッチ（アクティブ）+ Cisco Catalyst 9300 シリーズスイッチ（スタンバイ）	1	1 ポート 1 を使用した場合のみ。	サポート。 この設定は、**app-vnic AppgigabitEthernet port 1 trunk** コマンドまたは **app-vnic AppgigabitEthernet trunk** コマンドを使用してポート 1 が設定されている場合にサポートされます。 ポート番号が指定されていない場合は、スイッチオーバーが発生したときにデフォルトのポート 1 が使用されます。
Cisco Catalyst 9300X シリーズスイッチ（アクティブ）+ Cisco Catalyst 9300 シリーズスイッチ（スタンバイ）	1	2	未サポート このシナリオでは、スイッチオーバーが発生すると、新しいアクティブには前面パネルポートが 2 つないため、アプリケーションの設定が失敗します。 スイッチオーバー後、Cisco Catalyst 9300 シリーズ スイッチではアプリケーションが再起動されません。これは、前面パネルのポートが 1 つだけ設定されており、この設定が失敗するためです。使用可能な前面パネルポートを使用してアプリケーションを再設定する必要があります。

スタックモード	機能	使用されるポート	動作
Cisco Catalyst 9300X シリーズスイッチ（アクティブ）+ Cisco Catalyst 9300 シリーズスイッチ（スタンバイ）	2	2	未サポート （注） たとえば、app1 と app2 の 2 つのアプリケーションが実行されており、各アプリケーションがそれぞれ異なる前面パネルポート（port1 と port2 など）を使用しているとします。 スイッチオーバー後、前面パネル port1 上の app1 が実行状態の Cisco Catalyst 9300 シリーズ スイッチで開始されます。ただし app2 は、前面パネルに port2 がないため、Cisco Catalyst 9300 シリーズ スイッチでは開始されません。
Catalyst 9300 シリーズスイッチ（アクティブ）+Catalyst 9300X シリーズスイッチ（スタンバイ）	1 つ以上	1	サポート。 （注） スイッチオーバー後、アプリケーションは Catalyst 9300X シリーズ スイッチで前面パネルのポートを使用して再開始されます。

Cisco Catalyst 9400 シリーズ スイッチでのアプリケーション ホスティング

ここでは、Cisco Catalyst 9400 シリーズ スイッチでのアプリケーション ホスティングについて説明します。

アプリケーション ホスティングの場合、Cisco Catalyst 9400 シリーズ スイッチは管理インターフェイスと前面パネルポートをサポートします。アプリケーションは、C9400-SSD-240GB、C9400-SSD-480GB、および C9400-SSD-960GB ソリッドステートドライブ（SSD）でホストできます。

これらのスイッチは、アプリケーション ホスティングに M2 SATA モジュールを使用します。詳細については、『*Interfaces and Hardware Configuration Guide*』の「M2 SATA Module」の章を参照してください。

Cisco Catalyst 9400 シリーズ スイッチでは、アプリケーションはアクティブなスーパーバイザでのみホストできます。スイッチオーバー後、新しくアクティブになったスーパーバイザの AppGigbitEthernet インターフェイスがアクティブになり、アプリケーション ホスティングに使用できるようになります。

Cisco Catalyst 9410 シリーズ スイッチでのアプリケーション ホスティング

Cisco IOS XE Bengaluru 17.5.1 では、アプリケーション ホスティングが Cisco Catalyst 9410 シリーズ スイッチでサポートされています。アプリケーション ホスティング用に AppGigabitEthernet インターフェイスを有効にするには、インターフェイス コンフィギュレーション モードで **enable** コマンドを設定します。

（注） **enable** コマンドは、Cisco Catalyst 9410 シリーズ スイッチでのみ使用できます。

スロット 4 の 48 ポートラインカードをアプリケーション ホスティングに使用する場合、そのポートはデフォルトのシャットダウンモードである必要があります。スロット 4 の 48 ポートラインカードがアクティブな場合、アプリケーション ホスティングは拒否されます。ラインカードポートが無効な場合、スロット 4 の 48 ポートラインカードが非アクティブとしてマークされます。

スロット 4 に 48 ポートラインカードが装着されている場合、ポート 4/0/48 はアップ状態になりません。ラインカード4が空の場合、または24ポートラインカードの場合、無効になるポートはありません。

ポート（4/0/48）を有効にするには、**no iox** コマンドを使用してアプリケーション ホスティングを無効にします。ポートが有効または無効の場合、コンソールにシステムメッセージは表示されません。

インサービス ソフトウェア アップグレード（ISSU）の実行中は、AppGigabitEthernet インターフェイスを有効にする必要があるため、ラインカードポートは自動的には無効になりません。ソフトウェアのダウングレードの前に、AppGigabitEthernet インターフェイスを無効にして、前面パネルポートを無効にする必要があります。

ホットスワップ（OIR）

表 *43* : 活性挿抜（*OIR*）のシナリオ

OIR のシナリオ	アクション
スロット 4 のラインカードが空で、AppGigabitEthernetインターフェイスが有効になっている。	無効なポートはありません。

OIR のシナリオ	アクション
スロット 4 のラインカードが 48 ポートラインカードで、AppGigabitEthernet インターフェイスが有効になっている。	スロット 4 のポート 48 は無効です。ポートが無効になった後は、ポートに設定は適用されません。ポート 48 は非アクティブとしてマークされます。
スロット 4 のラインカードは 24 ポートラインカードである。	スロット 4 のポートは無効になりません。
スロット 4 のラインカードが 48 ポートのラインカードで、それが 24 ポートのラインカードに置き換えられ、AppGigabitEthernet インターフェイスが有効化された。	スロット 4 のポートは無効になりません。
スロット 4 のラインカードが 24 ポートのラインカードであり、48 ポートのラインカードに置き換えられ、AppGigabitEthernet インターフェイスが有効になっている。	スロット 4 のポート 48 は無効です。
OIR の操作中にスタンバイスーパーバイザが新しいアクティブになり、新しいアクティブの前面パネルポートがアプリケーション ホスティングに使用される。	スロット 4 のポート 48 の状態は変化しません。スタンバイスーパーバイザの OIR は、アクティブスーパーバイザの前面パネルポートには影響しません。

Cisco StackWise Virtual

ここでは、デュアルスーパーバイザのアップリンクポートを StackWise Virtual リンクとして使用する場合のシナリオについて説明します。

- アプリケーション ホスティングが有効で、ラインカード 4 のポート 48 がアップ状態ではない場合、アクティブシャーシとスタンバイシャーシの両方で無効になります。
- アクティブまたはスタンバイシャーシのラインカード 4 のポート 48 でリンクがアップ状態である場合、**enable** コマンドは拒否されます。
- ラインカード 4 のポート 48 をデュアルアクティブ検出（DAD）リンクとして使用する場合は、DAD リンクを削除し、これを別のポートで設定します。
- ラインカード 4 のポート 48 を StackWise Virtual リンクとして使用し、前面パネルポートを有効にする必要がある場合は、ポート 48 の StackWise Virtual リンクを削除し、別のポートを StackWise Virtual リンクとして使用します。ラインカード 4 のポート 48 は、StackWise Virtual または DAD リンクとして使用できません。

Cisco Catalyst 9500 シリーズ スイッチでのアプリケーション ホスティング

Cisco Catalyst 9500-High Performance シリーズ スイッチは、M2 SATA モジュール、SSD-240G、SSD-480G、および SSD-960（C9k-F1-SSD-240GB）のみをサポートします。前面パネルの USB はサポートされていません。

詳細については、『Cisco IOS XE Amsterdam 17.2.x（Catalyst 9500 スイッチ）インターフェイスおよびハードウェア コンポーネント コンフィギュレーション ガイド』の「M2 SATA モジュール」を参照してください。

Cisco IOS XE Cupertino 17.7.1 では、Cisco Catalyst 9500X シリーズ スイッチは、AppGigabitEthernet インターフェイスでのアプリケーション ホスティングをサポートしています。アプリケーション ホスティングは、次の M2 SATA モジュールでサポートされています。SSD-240G、SSD-480G、および SSD-960（C9k-F1-SSD-240GB）。

Cisco Catalyst 9600 シリーズ スイッチでのアプリケーション ホスティング

Cisco Catalyst 9600 シリーズ スイッチは、アプリケーション ホスティングのために M2 SATA モジュールのみをサポートします。前面パネルの USB はサポートされていません。次の M2 SATA モジュール（SSD-240G、SSD-480G、および SSD-960（C9k-F2-SSD-240GB））がサポートされています。

詳細については、『Cisco IOS XE Amsterdam 17.2.x（Catalyst 9600 スイッチ）インターフェイスおよびハードウェア コンポーネント コンフィギュレーション ガイド』の「M2 SATA モジュール」を参照してください。

内部フラッシュから SSD へのアプリケーションの自動転送および自動インストール

IOx が有効である場合、使用可能な最適なメディアが選択され、そのメディアを使用して IOx サービスが開始されます。IOx は、その起動時にアプリケーションを実行するメディアも選択します。

IOx が再起動して別のメディアが選択された場合は、すべてのアプリケーション（Docker アプリケーションのみサポートされます）を新しいメディアに移行し、コンテナを変更前と同じ状態に復元する必要があります。アプリケーションに関連付けられているているすべての永続データとボリュームも移行する必要があります。

再起動中、IOx は次の優先順位でメディアを選択します。

1. ハードディスク
2. フラッシュ

フラッシュはゲストシェルのみをサポートします。他のアプリケーションは許可されません。

ユースケース

このセクションでは、アプリケーションの自動転送および自動インストール中のいくつかのユースケースについて説明します。

表 *44*：アプリケーションの自動転送および自動インストールのユースケース

使用例	結果
IOx がフラッシュで実行している間に SSD が接続される。	IOx の実行中に SSD が接続される場合、実行中のアプリケーションまたは IOx に影響はありません。IOx が SSD に移行されるのは、IOx を無効にして再起動し、 CLI を介して有効にした場合、またはシステムの再起動の場合のみです。
IOx データが新しいメディアにコピーされている間に、システムがリブートします。	あるメディアから別のメディアへの IOx データの移行中にシステムが再起動した場合は、システムの再起動時に移行プロセスが続行されます。古いメディアのデータは、コピー操作が完了したときにのみ削除されます。

ThousandEyes Enterprise Agent の概要

ThousandEyes Enterprise Agent は、エンタープライズ ネットワーク監視ツールであり、ビジネスに影響を与えるネットワークとサービス全体のエンドツーエンドのビューを提供します。内部、外部、キャリア、およびインターネットネットワーク全体のネットワーク トラフィック パスをリアルタイムでモニタして、ネットワーク パフォーマンス データを提供します。Enterprise Agent は、WAN やインターネットの接続状態を詳細に把握するために、ブランチサイトやデータセンターにインストールするのが最も一般的です。

以前の Cisco IOS XE リリースでは、ThousandEyes は SSD 上のサードパーティ製カーネルベース仮想マシン（KVM）アプライアンスとしてサポートされていました。

Cisco IOS XE Amsterdam 17.3.3 では、ThousandEyes Enterprise Agent の新しいバージョンであるバージョン 3.0 が導入されました。これは、アプリケーションホスティング機能を使用してシスコデバイスで実行される組み込み型の Docker ベースアプリケーションです。Enterprise Agent は SSD とブートフラッシュの両方で使用でき、ブラウザテスト（ページロードとトランザクション）を除くすべてのテストをサポートします。ブラウザテストは、Cisco IOS XE Bengaluru 17.6.1 以降のリリースの Enterprise Agent バージョン 4.0 で使用できます。

ThousandEyes Enterprise Agent は次の機能を提供します。

- ネットワークとアプリケーションのパフォーマンスベンチマーク。

- 詳細なホップバイホップメトリック。
- ブランチまたはキャンパスからデータセンターまたはクラウドへのエンドツーエンドのパスの可視化。
- 機能停止の検出と解決。
- ユーザーエクスペリエンス分析。
- トラフィックフローパターンの可視化。

Cisco IOS XE Bengaluru 17.6.1 で利用可能な ThousandEyes Enterprise Agent バージョン 4.0 は、ThousandEyes Agent バージョン 3.0 では利用できない次の追加機能をサポートしています。

- BrowserBot のサポート（背面パネル SSD が使用可能な場合）。
- DNAC アプリケーションのアイコンおよび説明
- Docker ヘルスモニタリング。
- ThousandEyes Enterprise Agent をアップグレードするための **app-hosting upgrade URL** コマンド。

ThousandEyes Enterprise Agent の前提条件

- ThousandEyes サイトで入手可能な ThousandEyes Enterprise Agent イメージは、HTTPS ダウンロード用に www.cisco.com で使用される認証局（CA）と同じ認証局によって署名される必要があります。ユーザー名とパスワードは使用されません。
- Enterprise Agent をインストールするには、インターネット接続またはプロキシサーバーが必要です。詳細については、https://docs.thousandeyes.com/product-documentation/enterprise-agents にある ThousandEyes のドキュメントを参照してください。
- Enterprise Agent アプリケーションは、ユーザーのライセンス権限が検証された後にのみ使用できます。
- Docker ベースのアプリケーションのみがサポートされます。
- 1:1 スタックモードは、ThousandEyes ステートフル スイッチオーバー（SSO）をサポートするための必須条件です。

  1:1 モードとは、スタック内の特定のデバイスにアクティブロールとスタンバイロールが割り当てられる場合です。これは、スタック内の任意のスイッチをアクティブまたはスタンバイにすることができる従来の N+1 ロール選択アルゴリズムより優先されます。

ThousandEyes Enterprise Agent に必要なリソース

次の表に、ThousandEyes Enterprise Agent のインストールに必要なリソースを示します。

表 *45 : ThousandEyes Enterprise Agent* に必要なリソース

アプリケーションメディア	最大リソース	サポートされるリリース
SSD （注） 120G SSD のみがサポートされます。	• CPU：2 vCPU • メモリ：2G RAM • ストレージ：SSD 上の制限なし	Cisco IOS XE Amsterdam 17.3.3 • Cisco Catalyst 9300 および 9300L シリーズ スイッチ Cisco IOS XE Bengaluru 17.5.1 • Cisco Catalyst 9400 シリーズ スイッチ Cisco IOS XE Bengaluru 17.6.1 • Cisco Catalyst 9300X シリーズ スイッチ
フラッシュ	• CPU：2 vCPU • メモリ：2G RAM • ストレージ：フラッシュ ファイルシステムの 4G パーティションのうち、アプリケーションによる永続的なロギング用に 1G。ストレージは IOx メタデータと共有されます。	Cisco IOS XE Amsterdam 17.3.3 • Cisco Catalyst 9300 および 9300L シリーズ スイッチ Cisco IOS XE Bengaluru 17.5.1 • Cisco Catalyst 9400 シリーズ スイッチ Cisco IOS XE Bengaluru 17.6.1 • Cisco Catalyst 9300X シリーズ スイッチ

Cisco IOS XE Bengaluru 17.6.1では、アドオンモードはCisco Catalyst 9300、9300L、および9300X シリーズ スイッチ、および Cisco Catalyst 9400 シリーズ スイッチでサポートされています。

ThousandEyes Enterprise Agent のダウンロード

ThousandEyes Enterprise Agent には、ブラウンフィールドとグリーンフィールドの 2 つのタイプがあります。既存のデバイスの場合は、ThousandEyes Web サイトからブラウンフィールドバージョンをダウンロードできます。一方、新しいデバイスは、グリーンフィールド アプリケーションがブートフラッシュにロードされた状態で出荷されます。

次の表に、エージェントで使用可能なダウンロードオプションを示します。

表 *46 : ThousandEyes Enterprise Agent* のダウンロードオプション

ブラウンフィールド	グリーンフィールド
• ファイルをダウンロードします。https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-3.0.cat9k.tar ファイルは、HTTPS ダウンロード用に www.cisco.com で使用される認証局（CA）と同じ認証局によって署名されます。ユーザー名とパスワードは使用されません。 • アプリケーションをダウンロードして展開するには、**install** コマンドを使用します。	• ブートフラッシュの /apps フォルダにあります。デバイスに付属しています。 • アプリケーションをダウンロードして展開するには、**install** コマンドを使用します。

このセクションでは、エージェントの実行に必要な最大リソースについて説明します。

- CPU：vCPU x 2
- メモリ：2G
- ストレージ：フラッシュファイルシステムの 4G パーティションのうち、アプリケーションによる永続的なロギング用に 1G。このストレージは IOx メタデータによって共有されます。
- メディアストレージ：
  - Cisco IOS XE Amsterdam 17.3.3 の Cisco Catalyst 9300 および Cat9300 L シリーズ スイッチ用 120G SSD。
  - Cisco IOS XE Bengaluru 17.5.1 の Cisco Catalyst 9400 シリーズ スイッチ用の 240/480/960GB M2-SATA-HDD

Enterprise Agent がダウンロードされると、必要なアプリケーション設定を提供する ThousandEyes クラウドベースポータルへのセキュアチャネルを作成するためのコールが開始され、アプリケーションデータが収集されます。TE ポータルへのリンクは https://app.thousandeyes.com です。

ThousandEyes BrowserBot

ThousandEyes Enterprise Agent バージョン 4.0 には、トランザクション スクリプト テスト用の BrowserBot が用意されています。BrowserBot は、ページロードおよびトランザクションテストを管理する Enterprise Agent のコンポーネントです。BrowserBot を使用すると、ThousandEyes クラウドポータルでの Web ブラウザのアクションを模倣するカスタマイズされた JavaScript テストを有効にできます。ホスト オペレーティング システムを誤った JavaScript 操作から保護するために、ThousandEyes Agent はJavaScript を実行するサンドボックスコンテナを作成します。

制限がないディスクがアプリケーションで使用される場合、ThousandEyes Agent は初期化中に BrowserBot パッケージを動的にインストールします。これにより、ポータル トランザクション スクリプト テストを設定できます。

（注） BrowserBot のサポートは、ThousandEyes Agent バージョン 3.0 では適用されません。

BrowserBot は、大量のハードウェアリソースを消費します。2 GB のシステムメモリと 2 つの VCPU 負荷が、すべての IOx アプリケーションに割り当てられる最大 IOx システムメモリと CPU 負荷です。ブートフラッシュで複数のアプリケーションを同時に実行できるようにするには、エージェントをアクティブ化する前に、デフォルトの package.yaml BrowserBot のリソースを削減します。**app-resource profile custom** コマンドを使用して、デフォルトの package.yaml 設定を上書きします。

- CPU：1850 CPU ユニット（1/4 VCPU）
- メモリ：500MB

トランザクションのスクリプト化の詳細については、次のリンクを参照してください。

- https://docs.thousandeyes.com/product-documentation/tests/transaction-scripting-guide
- https://docs.thousandeyes.com/product-documentation/tests/transaction-scripting-reference

トランザクションのスクリプト化の例については、
https://github.com/thousandeyes/transaction-scripting-examples を参照してください。

ThousandEyes Agent のアップグレードとダウングレード

ThousandEyes Agent のアップグレード

Cisco IOS XE Amsterdam 17.3.3 および Bengaluru 17.5.1で使用可能な ThousandEyes Enterprise Agent 3.0 は、Cisco IOS XE Bengaluru17.6.1で使用可能な Agent 3.0 または Agent 4.0 にアップグレードできます。Agent 3.0 は、アップグレード後に操作によって復元されます。

Agent 4.0 は Cisco IOS XE Bengaluru 17.6.1 で使用でき、エージェントは、自動アップグレードにより起動時に最新の Agent 4.0 バイナリに更新されます。現在、Agent 4.0 に対するアップグレードはありません。

アプリケーションのアップグレードは、次の方法で実行できます。

- ThousandEyes エージェントの自動アップグレード：アプリケーションの起動時に自動的に実行されます。実行中のコンテナ内のエージェントバイナリはアップグレードされますが、アプリケーションパッケージはアップグレードされません。
- **app-hosting upgrade** コマンドを使用する。
- DNAC アプリのアップグレード。

ThousandEyes Agent のダウングレード

Cisco IOS XE Amsterdam 17.3.3、Cisco IOS XE Bengaluru 17.5.1、および Cisco IOS XE 17.6.1で使用可能なAgent 3.0 はダウングレードできません。

Cisco IOS XE Bengaluru 17.6.1 で使用可能な Agent 4.0 は、Cisco IOS XE Bengaluru 17.6.1 で使用可能な Agent 3.0 にダウングレードできます。他のダウングレードはできません。

ダウングレードするときに、アプリケーションが以前のリリースと同じ状態にならない場合は、アプリケーションを非アクティブ化またはアンインストールしてから、インストールまたは再起動します。

ネイティブ Docker コンテナ：アプリケーションの自動再起動

アプリケーションの自動再起動機能を使用すると、プラットフォームに導入されたアプリケーションは、システムのスイッチオーバーまたは再起動時に最後に設定された動作状態を維持できます。基盤となるホスティングフレームワークは、スイッチオーバー中も保持されます。この機能はデフォルトで有効であり、ユーザが無効にすることはできません。

アプリケーションの永続データは同期されません。 Cisco Application Framework（CAF）が認識しているセキュアデータストレージと永続データのみが同期されます。

スイッチオーバーまたはシステムの再起動時にIOxを同じ状態で再起動するには、アクティブデバイスとスタンバイデバイスにある IOx メディアが同期している必要があります。

Cisco Catalyst 9300 シリーズ スイッチは、アプリケーション ホスティングで SSD のみをサポートします。新しい SSD を挿入したら、他の SSD と同じ同期状態にする必要があります。アプリケーションの自動再起動同期を機能させるには、スタンバイデバイスにIOxと互換性のある SSD が必要です。

show iox-service コマンドの出力は同期の状態を表示します。

アプリケーションの自動再起動機能は、Cisco Catalyst 9300 シリーズ スイッチでのみサポートされます。

アプリケーションの自動再起動のシナリオ

ここでは、さまざまなアプリケーションの自動再起動のシナリオについて説明します。

表 *47:* アプリケーションの自動再起動のシナリオ

シナリオ	アクティブデバイスの単一メディア	アクティブデバイスとスタンバイデバイスのメディア
システムブートアップ	システムブートアップ時にIOxとアプリケーションを起動します。USB SSD はローカルデバイスであるため、すぐに表示されます。この時点では同期は行われません。	システムのブートアップ時にIOxとアプリケーションを起動します。既存の情報をスタンバイデバイスに一括同期します。

シナリオ	アクティブデバイスの単一メディア	アクティブデバイスとスタンバイデバイスのメディア
スイッチオーバー	新しいアクティブデバイスでメディアが見つかりません。IOx は、以前にインストールされたアプリケーションがなく、最小限の機能を持つシステムフラッシュで起動します。	システムスイッチオーバー（SSO）後に、新しいアクティブデバイスでIOxとアプリケーションを以前の状態で起動します。新しいスタンバイデバイスがブートアップした後に、情報の一括同期を実行します。
ブートアップまたはスイッチオーバー：USB SSD がメンバーデバイスに存在します。	メンバーデバイスに存在する SSD の同期はありません。メンバー SSD は IOx およびアプリケーションのホストには使用されません。	メンバーデバイスに存在する SSD の同期はありません。メンバー SSD は IOx およびアプリケーションのホストには使用されません。
デバイスの削除：アクティブデバイスからローカル USB SSD が削除されます。	ローカル USB SSD が削除されると、IOx がグレースフル終了を処理します。 SSD がアクティブデバイスに差し戻されたら、ユーザがトリガーする IOx の再起動が必要です。	IOx はグレースフル終了を処理します。IOx はローカルディスク上でのみ動作するため、スタンバイ SSD は IOx の起動に使用されません。 SSD がアクティブデバイスに差し戻されたら、ユーザがトリガーする IOx の再起動が必要です。
デバイスの削除：USB SSD がスタンバイデバイスから削除されます。	該当なし	IOx 同期操作が失敗します。IOx は SSO 対応ではなくなりました。
デバイスの削除：リモート USB SSD がリモートメンバーデバイスから削除されます。	IOx はメンバー SSD を使用しないため、影響はありません。	IOx はメンバー SSD を使用しないため、影響はありません。
デバイスのダウン：IOx が実行されているアクティブなデバイスがダウンします。	新しいアクティブデバイスでメディアが見つかりません。IOx は、以前にインストールされたアプリケーションがなく、最小限の機能を持つ状態でシステムフラッシュで起動します。	新しいアクティブデバイスで、SSO 前の状態で IOx とアプリケーションを起動します。新しいスタンバイデバイスがブートアップすると、情報の一括同期を実行します。

シナリオ	アクティブデバイスの単一メディア	アクティブデバイスとスタンバイデバイスのメディア
指定されたアクティブ/スタンバイデバイスの変更（スタック環境 1:1）	変更はリブート後に反映されます。リブート後、新しいアクティブデバイスから IOx が起動します。	変更はリブート後に反映されます。リブート後、新しいアクティブデバイスから IOx が起動します。

Cisco Catalyst 9300 シリーズ スイッチでのアプリケーション自動再起動

ここでは、マルチメンバースタックの Cisco Catalyst 9300 シリーズ スイッチでアプリケーションの自動再起動がどのように機能するかについて説明します。

Cisco Catalyst 9300 シリーズ スイッチでは、アプリケーションの自動再起動は、スタック内の特定のデバイスにアクティブロールとスタンバイロールを割り当てる、1+1 スイッチ冗長モードまたは StackWise Virtual モードでサポートされます。

スイッチスタックが N+1 モードの場合、アプリケーションの自動再起動はサポートされません。デバイスが N+1 モードの場合、次のログメッセージがコンソールに表示されます。

```
Feb 5 20:29:17.022: %IOX-3-IOX_RESTARTABITLITY: Switch 1 R0/0: run_ioxn_caf:Stack is in
 N+1 mode,
disabling sync for IOx restartability
```

IOxは、背面パネルのUSBポートでシスコ認定のUSB3.0フラッシュドライブをアプリケーションホスティング用のストレージとして使用します。このメディアは、すべてのスタックメンバーに存在するわけではありません。

データは、rsync ユーティリティを使用してアクティブデバイスからスタンバイデバイスに同期されます。

サポート対象ネットワークタイプ

ここでは、Cisco Catalyst スイッチでサポートされるネットワークのタイプを示します。

表 *48* : サポート対象ネットワークタイプ

ネットワークタイプ	サポートされているプラットフォームとリリース
管理ポート	• Cisco IOS XE Gibraltar 16.12.1 の Catalyst 9300 シリーズ スイッチおよび C9300L • Cisco IOS XE Amsterdam 17.1.1 の Catalyst 9400 シリーズ スイッチ • Cisco IOS XE Amsterdam 17.2.1 の Catalyst 9500 シリーズ スイッチおよび Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ • Cisco IOS XE Amsterdam 17.2.1 の Catalyst 9600 シリーズ スイッチ
前面パネルポート（トランクおよび VLAN）	• Cisco IOS XE Gibraltar 16.12.1 の Catalyst 9300 シリーズ スイッチおよび C9300L • Cisco IOS XE Amsterdam 17.1.1 の Catalyst 9400 シリーズ スイッチ • Cisco IOS XE Amsterdam 17.5.1 の Catalyst 9600 シリーズ スイッチ • Cisco IOS XE Bengaluru 17.6.1 の Catalyst 9300X シリーズ スイッチ （注） Catalyst 9300X シリーズ スイッチは、複数の AppGigabitEthernet ポートをサポートします。
Cisco IOS ネットワークアドレス変換（NAT）	• Cisco IOS XE Gibraltar 16.12.1 の Catalyst 9300 シリーズ スイッチおよび C9300L • Cisco IOS XE Amsterdam 17.1.1 の Catalyst 9400 シリーズ スイッチ これらのプラットフォームの両方で、前面パネルのデータポートおよび AppGigabitEthernet ポートに適用されるハードウェアデータポート機能によって NAT がサポートされます。
Cisco IOx NAT	サポート対象外

仮想ネットワーク インターフェイス カード

アプリケーションコンテナのライフサイクルを管理するには、内部論理インターフェイスごとに 1 つのコンテナをサポートするレイヤ 3 ルーティングモデルが使用されます。これは、各アプリケーションに対して仮想イーサネットペアが作成されることを意味します。このペアのうち仮想ネットワーク インターフェイス カードと呼ばれるインターフェイスは、アプリケーション コンテナの一部です。

NIC は、コンテナ内の標準イーサネットインターフェイスで、プラットフォームデータ プレーンに接続してパケットを送受信します。Cisco IOx は、コンテナ内の各 vNIC について、IP アドレスおよび一意の MAC アドレス割り当てを行います。

コンテナ内の vNIC は、標準のイーサネット インターフェイスと見なされます。

アプリケーション ホスティングの設定方法

ここでは、アプリケーション ホスティングの設定を構成するさまざまな作業について説明します。

Cisco IOx の有効化

このタスクを実行して Cisco IOx へのアクセスを有効にすることで、CLI ベースのユーザインターフェイスでホストシステム上のアプリケーションの管理、制御、モニタ、トラブルシューティング、および関連するさまざまなアクティビティを実行できます。

手順の概要

1. **enable**
2. **configure terminal**
3. **iox**
4. **username** *name* **privilege** *level* **password** {**0** | **7** | *user-password*}*encrypted-password*
5. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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。

	コマンドまたはアクション	目的
ステップ 3	**iox** 例： `Device(config)# iox`	Cisco IOx をイネーブルにします。
ステップ 4	**username** *name* **privilege** *level* **password** {**0** | **7** | *user-password*}*encrypted-password* 例： `Device(config)# username cisco privilege 15 password 0 ciscoI`	ユーザー名ベースの認証システムとユーザーの権限レベルを確立します。 • ユーザー名の特権レベルは 15 に設定する必要があります。
ステップ 5	**end** 例： `Device(config)# end`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

前面パネル VLAN ポートのアプリケーション ホスティングの設定

（注） このタスクは、Cisco IOS XE Amsterdam 17.1.1 以降のリリースに適用されます。

アプリケーション ホスティング トランク コンフィギュレーション モードでは、許可されるすべての AppGigabitEthernet VLAN ポートがコンテナに接続されます。ネイティブおよび VLAN タグ付きフレームは、コンテナ ゲスト インターフェイスによって送受信されます。AppGigabitEthernet トランクポートにマッピングできるコンテナ ゲスト インターフェイスは 1 つだけです。

トランクポートと VLAN アクセスポートの両方の同時設定がサポートされます。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface** *AppGigabitEthernet number*
4. **switchport trunk allowed vlan** *vlan-ID*
5. **switchport mode trunk**
6. **exit**
7. **app-hosting appid** *name*
8. **app-vnic AppGigabitEthernet trunk**
9. **vlan** *vlan-ID* **guest-interface** *guest-interface-number*
10. **guest-ipaddress** *ip-address* **netmask** *netmask*
11. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**interface** *AppGigabitEthernet number* 例： `Device(config)# interface AppGigabitEthernet 1/0/1`	AppGigabitEthernet を設定し、インターフェイス コンフィギュレーション モードを開始します。 • スタック可能スイッチの場合、*number* 引数は *switch-number/0/1* です。
ステップ 4	**switchport trunk allowed vlan** *vlan-ID* 例： `Device(config-if)# switchport trunk allowed vlan 10-12,20`	トランク上で許可される VLAN のリストを設定します。
ステップ 5	**switchport mode trunk** 例： `Device(config-if)# switchport mode trunk`	インターフェイスを永続的なトランキングモードに設定して、ネイバーリンクのトランクリンクへの変換をネゴシエートします。
ステップ 6	**exit** 例： `Device(config-if)# exit`	インターフェイス コンフィギュレーション モードを終了し、グローバルコンフィギュレーションモードに戻ります。
ステップ 7	**app-hosting appid** *name* 例： `Device(config)# app-hosting appid iox_app`	アプリケーションを設定し、アプリケーション ホスティング コンフィギュレーション モードを開始します。
ステップ 8	**app-vnic AppGigabitEthernet trunk** 例： `Device(config-app-hosting)# app-vnic AppGigabitEthernet trunk`	トランクポートをアプリケーションの前面パネル ポートとして設定し、アプリケーション ホスティング トランク コンフィギュレーション モードを開始します。
ステップ 9	**vlan** *vlan-ID* **guest-interface** *guest-interface-number* 例： `Device(config-config-app-hosting-trunk)# vlan 10 guest-interface 2`	VLAN ゲストインターフェイスを設定し、アプリケーション ホスティング VLAN アクセス IP コンフィギュレーション モードを開始します。 • 複数の VLAN からゲストインターフェイスへのマッピングがサポートされます。

	コマンドまたはアクション	目的
ステップ 10	**guest-ipaddress** *ip-address* **netmask** *netmask* 例： `Device(config-config-app-hosting-vlan-access-ip)# guest-ipaddress 192.168.0.2 netmask 255.255.255.0`	（オプション）静的 IP を設定します。
ステップ 11	**end** 例： `Device(config-config-app-hosting-vlan-access-ip)# end`	アプリケーション ホスティング VLAN アクセス IP コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

前面パネルトランクポートのアプリケーション ホスティングの設定

アプリケーション ホスティング トランク コンフィギュレーション モードでは、許可されるすべての AppGigabitEthernet VLAN ポートがコンテナに接続されます。ネイティブおよび VLAN タグ付きフレームは、コンテナ ゲスト インターフェイスによって送受信されます。AppGigabitEthernet トランクポートにマッピングできるコンテナ ゲスト インターフェイスは 1 つだけです。

Cisco IOS XE Gibraltar 16.2.1 では、アプリケーション ID は、アプリケーション ホスティング トランク コンフィギュレーション モードまたはアプリケーション ホスティング VLAN アクセス コンフィギュレーション モードで設定できますが、両方のモードで設定することはできません。

Cisco IOS XE Amsterdam 17.1.1 以降のリリースでは、トランクポートと VLAN アクセスポートの両方の同時設定がサポートされています。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface** *AppGigabitEthernet number*
4. **switchport trunk allowed vlan** *vlan-ID*
5. **switchport mode trunk**
6. **exit**
7. **app-hosting appid** *name*
8. **app-vnic AppGigabitEthernet trunk**
9. **guest-interface** *guest-interface-number*
10. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**interface** *AppGigabitEthernet number* 例： `Device(config)# interface AppGigabitEthernet 1/0/1`	AppGigabitEthernet を設定し、インターフェイス コンフィギュレーション モードを開始します。 • スタック可能スイッチの場合、*number* 引数は *switch-number/0/1* です。
ステップ 4	**switchport trunk allowed vlan** *vlan-ID* 例： `Device(config-if)# switchport trunk allowed vlan 10-12,20`	トランク上で許可される VLAN のリストを設定します。
ステップ 5	**switchport mode trunk** 例： `Device(config-if)# switchport mode trunk`	インターフェイスを永続的なトランキングモードに設定して、ネイバーリンクのトランクリンクへの変換をネゴシエートします。
ステップ 6	**exit** 例： `Device(config-if)# exit`	インターフェイス コンフィギュレーション モードを終了し、グローバルコンフィギュレーションモードに戻ります。
ステップ 7	**app-hosting appid** *name* 例： `Device(config)# app-hosting appid iox_app`	アプリケーションを設定し、アプリケーション ホスティング コンフィギュレーション モードを開始します。
ステップ 8	**app-vnic AppGigabitEthernet trunk** 例： `Device(config-app-hosting)# app-vnic AppGigabitEthernet trunk`	トランクポートをアプリケーションの前面パネルポートとして設定し、アプリケーション ホスティング トランク コンフィギュレーション モードを開始します。
ステップ 9	**guest-interface** *guest-interface-number* 例： `Device(config-config-app-hosting-trunk)# guest-interface 2`	AppGigabitEthernet インターフェイストランクに接続されているアプリケーションのインターフェイスを設定します。

	コマンドまたはアクション	目的
ステップ 10	**end** 例： `Deviceconfig-config-app-hosting-trunk)# end`	アプリケーション ホスティング トランク コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

コンフィギュレーション モードでのアプリケーションの起動

アプリケーションホスティングコンフィギュレーションモードの**start**コマンドは、**app-hosting activate appid** および **app-hosting start appid** コマンドと同等です。

アプリケーション ホスティング コンフィギュレーション モードの **no start** コマンドは、**app-hosting stop appid** および **app-hosting deactivate appid** コマンドと同等です。

（注） アプリケーションをインストールする前に **start** コマンドを設定してから **install** コマンドを設定すると、Cisco IOx は自動的に内部 **activate** アクションと **start** アクションを実行します。これにより、**install** コマンドを設定することでアプリケーションを自動的に起動できます。

手順の概要

1. **enable**
2. **configure terminal**
3. **app-hosting appid** *application-name*
4. **start**
5. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**app-hosting appid** *application-name* 例： `Device(config)# app-hosting appid iox_app`	アプリケーションを設定し、アプリケーション ホスティング コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 4	**start** 例： `Device(config-app-hosting)# start`	（任意）アプリケーションを起動して実行します。 • アプリケーションを停止するには、**no start** コマンドを使用します。
ステップ 5	**end** 例： `Device(config-app-hosting)# end`	アプリケーション ホスティング コンフィギュレーション モードを終了し、特権EXECモードに戻ります。

アプリケーションのライフサイクル

次の EXEC コマンドを使用すると、アプリケーションのライフサイクルを確認できます。

（注） アプリケーションのインストール後に設定の変更が行われた場合、実行状態のアプリケーションにはこれらの変更が反映されません。設定の変更を有効にするには、アプリケーションを明示的に停止して非アクティブにし、再度アクティブにして再起動する必要があります。

手順の概要

1. **enable**
2. **app-hosting install appid** *application-name* **package** *package-path*
3. **app-hosting activate appid** *application-name*
4. **app-hosting start appid** *application-name*
5. **app-hosting stop appid** *application-name*
6. **app-hosting deactivate appid** *application-name*
7. **app-hosting uninstall appid** *application-name*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**app-hosting install appid** *application-name* **package** *package-path* 例： `Device# app-hosting install appid iox_app package usbflash1:my_iox_app.tar`	指定した場所からアプリケーションをインストールします。 • アプリケーションは、flash、bootflash、usbflash0、usbflash1、harddisk などのローカル ストレージの場所からインストールできます。

	コマンドまたはアクション	目的
ステップ 3	**app-hosting activate appid** *application-name* 例： `Device# app-hosting activate appid iox_app`	アプリケーションをアクティブ化します。 • このコマンドは、すべてのアプリケーション リソース要求を検証し、すべてのリソースが使用可能な場合はアプリケーションがアクティブになります。それ以外の場合は、アクティベーションが失敗します。
ステップ 4	**app-hosting start appid** *application-name* 例： `Device# app-hosting start appid iox_app`	アプリケーションを起動します。 • アプリケーションの起動スクリプトがアクティブ化されます。
ステップ 5	**app-hosting stop appid** *application-name* 例： `Device# app-hosting stop appid iox_app`	（任意）アプリケーションを停止します。
ステップ 6	**app-hosting deactivate appid** *application-name* 例： `Device# app-hosting deactivate appid iox_app`	（任意）アプリケーションに割り当てられているすべてのリソースを無効にします。
ステップ 7	**app-hosting uninstall appid** *application-name* 例： `Device# app-hosting uninstall appid iox_app`	（任意）アプリケーションをアンインストールします。 • 保存されているすべてのパッケージとイメージをアンインストールします。アプリケーションに対するすべての変更と更新も削除されます。

Docker ランタイムオプションの設定

最大 30 行のランタイムオプションを追加できます。システムは、1 行目から 30 行目までの連結文字列を生成します。文字列には、複数の Docker ランタイムオプションを指定できます。

ランタイムオプションが変更された場合は、アプリケーションを停止、非アクティブ化、アクティブ化、および起動して、新しいランタイムオプションを有効にします。

手順の概要

1. **enable**
2. **configure terminal**
3. **app-hosting appid** *application-name*
4. **app-resource docker**
5. **run-opts** *options*
6. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**app-hosting appid** *application-name* 例： `Device(config)# app-hosting appid iox_app`	アプリケーションを設定し、アプリケーション ホスティング コンフィギュレーション モードを開始します。
ステップ 4	**app-resource docker** 例： `Device(config-app-hosting)# app-resource docker`	アプリケーション ホスティング Docker コンフィギュレーション モードを開始して、アプリケーション リソースの更新を指定します。
ステップ 5	**run-opts** *options* 例： `Device(config-app-hosting-docker)# run-opts 1 "-v $(APP_DATA):/data"`	Docker ランタイム オプションを指定します。
ステップ 6	**end** 例： `Device(config-app-hosting-docker)# end`	アプリケーション ホスティング Docker コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

コンテナの静的 IP アドレスの設定

コンテナに静的 IP アドレスを設定する場合は、次のガイドラインが適用されます。

- 最後に設定されたデフォルト ゲートウェイ設定のみが使用されます。
- 最後に設定されたネーム サーバ設定のみが使用されます。

Cisco IOS CLI を使用して、コンテナの IP アドレスを設定できます。

手順の概要

1. **enable**
2. **configure terminal**
3. **app-hosting appid** *name*
4. **name-server#** *ip-address*
5. **app-vnic management guest-interface** *interface-number*
6. **guest-ipaddress** *ip-address* **netmask** *netmask*

7. **exit**
8. **app-default-gateway** *ip-address* **guest-interface** *network-interface*
9. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**app-hosting appid** *name* 例： `Device(config)# app-hosting appid iox_app`	アプリケーションを設定し、アプリケーション ホスティング コンフィギュレーション モードを開始します。
ステップ 4	**name-server#** *ip-address* 例： `Device(config-app-hosting)# name-server0 10.2.2.2`	ドメインネームシステム（DNS）サーバを設定します。
ステップ 5	**app-vnic management guest-interface** *interface-number* 例： `Device(config-app-hosting)# app-vnic management guest-interface 0`	仮想ネットワーク インターフェイスおよびゲスト インターフェイスの管理ゲートウェイを設定し、アプリケーションホスティング ゲートウェイ コンフィギュレーション モードを開始します。
ステップ 6	**guest-ipaddress** *ip-address* **netmask** *netmask* 例： `Device(config-app-hosting-mgmt-gateway)# guest-ipaddress 172.19.0.24 netmask 255.255.255.0`	管理ゲストインターフェイスの詳細を設定します。
ステップ 7	**exit** 例： `Device(config-app-hosting-mgmt-gateway)# exit`	アプリケーション ホスティング管理ゲートウェイ コンフィギュレーション モードを終了し、アプリケーション ホスティング コンフィギュレーション モードに戻ります。
ステップ 8	**app-default-gateway** *ip-address* **guest-interface** *network-interface* 例： `Device(config-app-hosting)# app-default-gateway 172.19.0.23 guest-interface 0`	デフォルトの管理ゲートウェイを設定します。

	コマンドまたはアクション	目的
ステップ 9	**end** 例： `Device(config-app-hosting)# end`	アプリケーション ホスティング コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

管理ポートでのアプリケーション ホスティングの設定

手順の概要

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet0/0**
4. **vrf forwarding** *vrf-name*
5. **ip address** *ip-address mask*
6. **exit**
7. **app-hosting appid** *name*
8. **app-vnic management guest-interface** *network-interface*
9. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**interface gigabitethernet0/0** 例： `Device(config)# interface gigabitethernet0/0`	インターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。 • Cisco Catalyst 9000 シリーズ スイッチでは、管理インターフェイスは GigabitEthernet0/0 です。
ステップ 4	**vrf forwarding** *vrf-name* 例： `Device(config-if)# vrf forwarding Mgmt-vrf`	インターフェイスまたはサブインターフェイスに Virtual Routing and Forwarding（VRF）インスタンスまたは仮想ネットワークを関連付けます。 • *Mgmt-vrf* は、Cisco Catalyst 9000 シリーズ スイッチの管理インターフェイスに自動的に設定されます。

	コマンドまたはアクション	目的
ステップ 5	**ip address** *ip-address mask* 例： `Device(config-if)# ip address 198.51.100.1 255.255.255.254`	インターフェイスに IP アドレスを設定します。
ステップ 6	**exit** 例： `Device(config-if)# exit`	インターフェイス コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードに戻ります。
ステップ 7	**app-hosting appid** *name* 例： `Device(config)# app-hosting appid iox_app`	アプリケーションを設定し、アプリケーション ホスティング コンフィギュレーション モードを開始します。
ステップ 8	**app-vnic management guest-interface** *network-interface* 例： `Device(config-app-hosting)# app-vnic management guest-interface 1`	ゲストインターフェイスを管理ポートに接続し、アプリケーションホスティング管理ゲートウェイ コンフィギュレーション モードを開始します。 • **management** キーワードは、コンテナに接続されている Cisco IOS 管理 GigabitEthernet0/0 インターフェイスを指定します。 • **guest-interface** *network-interface* のキーワード引数ペアは、Cisco IOS 管理インターフェイスに接続されているコンテナの内部イーサネット インターフェイス番号を指定します。この例では、コンテナのイーサネット 1 インターフェイスに対して *guest-interface 1* を使用しています。
ステップ 9	**end** 例： `Device(config-app-hosting-mgmt-gateway)# end`	アプリケーションホスティング管理ゲートウェイ コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

アプリケーションの IP アドレスの手動設定

次の方法を使用して、コンテナの IP アドレスを設定できます。

- コンテナにログインし、**ifconfig** Linux コマンドを設定します。

  1. 次のコマンドを使用して、アプリケーションにログインします。

     ```
     app-hosting connect appid APPID {session | console}
     ```

  2. アプリケーションの Linux サポートに基づいて、標準の Linux インターフェイス コンフィギュレーション コマンドを使用します。

     ```
     - ifconfig dev IFADDR/subnet-mask-length
     ```

または

```
- ip address {add|change|replace} IFADDR dev IFNAME [ LIFETIME ] [ CONFFLAG-LIST
 ]
```

- コンテナで Dynamic Host Configuration Protocol（DHCP）を有効にし、Cisco IOS の設定で DHCP サーバとリレーエージェントを設定します。
  - Cisco IOx は、アプリケーション DHCP インターフェイスに使用されるアプリケーションコンテナ内で実行する DHCP クライアントを提供します。

アプリケーションのリソース設定の上書き

リソースの変更を有効にするには、最初に **app-hosting stop** および **app-hosting deactivate** コマンドを使用してアプリケーションを停止して非アクティブ化し、次に **app-hosting activate** および **app-hosting start** コマンドを使用してアプリケーションを再起動する必要があります。

アプリケーション ホスティング コンフィギュレーション モードで **start** コマンドを使用している場合は、**no start** および **start** コマンドを設定します。

これらのコマンドを使用して、リソースと app-hosting appid iox_app 設定の両方をリセットできます。

手順の概要

1. **enable**
2. **configure terminal**
3. **app-hosting appid** *name*
4. **app-resource profile** *name*
5. **cpu** *unit*
6. **memory** *memory*
7. **vcpu** *number*
8. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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。

	コマンドまたはアクション	目的
ステップ 3	**app-hosting appid** *name* 例： `Device(config)# app-hosting appid iox_app`	アプリケーションホスティングをイネーブルにし、アプリケーション ホスティング コンフィギュレーション モードを開始します。
ステップ 4	**app-resource profile** *name* 例： `Device(config-app-hosting)# app-resource profile custom`	カスタム アプリケーション リソース プロファイルを設定し、カスタム アプリケーション リソース プロファイル コンフィギュレーション モードを開始します。 • カスタムプロファイル名のみがサポートされています。
ステップ 5	**cpu** *unit* 例： `Device(config-app-resource-profile-custom)# cpu 7400`	アプリケーションのデフォルトのCPU割り当てを変更します。 • リソース値はアプリケーション固有のため、これらの値を変更した場合、アプリケーションが変更後も確実に稼働できることを確認する必要があります。
ステップ 6	**memory** *memory* 例： `Device(config-app-resource-profile-custom)# memory 2048`	デフォルトのメモリ割り当てを変更します。
ステップ 7	**vcpu** *number* 例： `Device(config-app-resource-profile-custom)# vcpu 2`	アプリケーションの仮想 CPU（vCPU）割り当てを変更します。
ステップ 8	**end** 例： `Device(config-app-resource-profile-custom)# end`	カスタム アプリケーション リソース プロファイル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

ThousandEyes Enterprise Agent のインストール

Enterprise Agent をインストールするには、次の手順を実行します。

1. IOx を設定します。詳細については、Cisco IOx の有効化に関する項を参照してください。
2. アプリケーション ホスティングを設定する。
3. AppGigabitEthernet ポートを設定する。
4. ThousandEyes Enterprise Agent をインストールする。

ThousandEyes Enterprise Agent のアプリケーション ホスティングの設定

手順の概要

1. **enable**
2. **configure terminal**
3. **app-hosting appid** *application-name*
4. **app-vnic AppGigabitEthernet trunk**
5. **vlan** *vlan-ID* **guest-interface** *guest-interface-number*
6. **guest-ip** *ip-address* **netmask** *netmask*
7. **exit**
8. **exit**
9. **app-default-gateway** *ip-address* **guest-interface** *network-interface*
10. **nameserver#** *ip-address*
11. **app-resource docker**
12. **run-opts** *options*
13. **prepend-pkg-opts**
14. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを開始します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**app-hosting appid** *application-name* 例： `Device(config)# app-hosting appid appid 1keyes`	アプリケーションを設定し、アプリケーション ホスティング コンフィギュレーション モードを開始します。
ステップ 4	**app-vnic AppGigabitEthernet trunk** 例： `Device(config-app-hosting)# app-vnic AppGigabitEthernet trunk`	トランクポートをアプリケーションの前面パネル ポートとして設定し、アプリケーション ホスティング トランク コンフィギュレーション モードを開始します。
ステップ 5	**vlan** *vlan-ID* **guest-interface** *guest-interface-number* 例： `Device(config-config-app-hosting-trunk)# vlan 10 guest-interface 2`	VLAN ゲストインターフェイスを設定し、アプリケーション ホスティング VLAN アクセス IP コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 6	**guest-ip** *ip-address* **netmask** *netmask* 例： `Device(config-config-app-hosting-vlan-access-ip)# guest-ipaddress 172.19.0.24 netmask 255.255.255.0`	ゲストインターフェイスのスタティック IP アドレスを設定します。
ステップ 7	**exit** 例： `Device(config-config-app-hosting-vlan-access-ip)# exit`	アプリケーション ホスティング VLAN アクセス IP コンフィギュレーション モードを終了し、アプリケーション ホスティング トランク コンフィギュレーション モードに戻ります。
ステップ 8	**exit** 例： `Device(config-config-app-hosting-trunk)# exit`	アプリケーション ホスティング トランク コンフィギュレーション モードを終了し、アプリケーション ホスティング コンフィギュレーション モードに戻ります。
ステップ 9	**app-default-gateway** *ip-address* **guest-interface** *network-interface* 例： `Device(config-app-hosting)# app-default-gateway 172.19.0.23 guest-interface 0`	デフォルトの管理ゲートウェイを設定します。
ステップ 10	**nameserver#** *ip-address* 例： `Device(config-app-hosting)# name-server0 10.2.2.2`	DNS サーバを設定します。
ステップ 11	**app-resource docker** 例： `Device(config-app-hosting)# app-resource docker`	アプリケーション ホスティング Docker コンフィギュレーションモードを開始して、アプリケーションリソースの更新を指定します。
ステップ 12	**run-opts** *options* 例： `Device(config-app-hosting-docker)# run-opts 1 "-e TEAGENT_ACCOUNT_TOKEN=[account-token]"`	Docker ランタイムオプションを指定します。
ステップ 13	**prepend-pkg-opts** 例： `Device(config-app-hosting-docker)# prepend-pkg-opts`	パッケージオプションを Docker ランタイムオプションとマージします。 • 重複する変数は上書きされます。
ステップ 14	**end** 例： `Device(config-app-hosting-docker)# end`	アプリケーション ホスティング Docker コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

ThousandEyes Enterprise Agent の AppGigabitEthernet インターフェイスの設定

手順の概要

1. **enable**
2. **configure terminal**
3. **interface appgigabitethernet** *number*
4. **switchport trunk allowed vlan** *vlan-ID*
5. **switchport mode trunk**
6. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを開始します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**interface appgigabitethernet** *number* 例： `Device(config)# interface AppGigabitEthernet 1/0/1`	AppGigabitEthernet を設定し、インターフェイス コンフィギュレーション モードを開始します。 • スタック可能スイッチの場合、*number* 引数は *switch-number/0/1* です。
ステップ 4	**switchport trunk allowed vlan** *vlan-ID* 例： `Device(config-if)# switchport trunk allowed vlan 10-12,20`	トランク上で許可される VLAN のリストを設定します。
ステップ 5	**switchport mode trunk** 例： `Device(config-if)# switchport mode trunk`	インターフェイスを永続的なトランキングモードに設定して、ネイバーリンクのトランクリンクへの変換をネゴシエートします。
ステップ 6	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

ThousandEyes Enterprise Agent のインストール

始める前に

ThousandEyes Enterprise Agent は、以下の URL またはフラッシュファイルシステムからインストールできます。

手順の概要

1. **enable**
2. **app-hosting install appid** *application-name* **package** *package-path*
3. **app-hosting start appid** *application-name*
4. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを開始します。

ステップ 2	**app-hosting install appid** *application-name* **package** *package-path*
	例：
	`Device# app-hosting install 1keyes https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-3.0.cat9k.tar`
	または
	`Device# app-hosting install appid 1keyes package flash:/apps/[greenfield-app-tar]`
		指定した場所からアプリケーションをインストールします。

ステップ 3	**app-hosting start appid** *application-name* 例： `Device# app-hosting start appid 1keyes`	（オプション）アプリケーションを開始します。
ステップ 4	**end** 例： `Device# end`	アプリケーション ホスティング コンフィギュレーション モードを終了し、特権EXECモードに戻ります。

次に、**show app-hosting list** コマンドの出力例を示します。

```
Device# show app-hosting list

App id                                   State
---------------------------------------------------------
1keyes                                   RUNNING
```

アプリケーション ホスティング設定の確認

show コマンドを使用して設定を確認します。コマンドはどの順序で使用してもかまいません。

手順の概要

1. **enable**
2. **show iox-service**
3. **show app-hosting detail**
4. **show app-hosting device**
5. **show app-hosting list**
6. **show interfaces trunk**
7. **show controller ethernet-controller AppGigabitEthernet** *interface-number*

手順の詳細

ステップ 1 enable

特権 EXEC モードを有効にします。

- パスワードを入力します（要求された場合）。

例：

```
Device> enable
```

ステップ 2 show iox-service

すべての Cisco IOx サービスのステータスを表示します。

例：

```
Device# show iox-service

IOx Infrastructure Summary:
---------------------------
IOx service (CAF)          : Not Running
IOx service (HA)           : Not Running
IOx service (IOxman)       : Not Running
IOx service (Sec storage)  : Not Running
Libvirtd                   : Running
Dockerd                    : Not Running
Application DB Sync Info   : Not available
```

ステップ 3 show app-hosting detail

アプリケーションに関する詳細情報を表示します。

例：

```
Device# show app-hosting detail

State                       : Running
Author                      : Cisco Systems, Inc
Application
 Type                       : vm
 App id                     : Wireshark
 Name                       : Wireshark
 Version                    : 3.4
 Activated Profile Name     : custom
 Description                : Ubuntu based Wireshark
Resource Reservation
 Memory                     : 1900 MB
 Disk                       : 10 MB
 CPU                        : 4000 units
 VCPU                       : 2
Attached devices
Type          Name          Alias
---------------------------------------------------------------------
Serial/shell
Serial/aux
Serial/Syslog               serial2
Serial/Trace                serial3
Network Interfaces
---------------------------------------------------------------
eth0:
 MAC address                : 52:54:dd:80:bd:59
 IPv4 address
eth1:
 MAC address                : 52:54:dd:c7:7c:aa
 IPv4 address
```

ステップ 4 show app-hosting device

USB デバイスに関する情報を表示します。

例：

```
Device# show app-hosting device

USB port Device name Available
1 Front_USB_1 true


app-hosting appid testvm
app-vnic management guest-interface 0
app-device usb-port 1
```

ステップ 5 show app-hosting list

アプリケーションとそれらのステータスの一覧を表示します。

例：

```
Device# show app-hosting list

App id              State
------------------------------------------
Wireshark           Running
```

ステップ 6 show interfaces trunk

トランクインターフェイス情報を表示します。

例：

```
Device# show interfaces trunk

Port Mode Encapsulation Status Native vlan
Gi3/0/1 on 802.1q trunking 1
Ap3/0/1 on 802.1q trunking 1

Port Vlans allowed on trunk
Gi3/0/1 1-4094
Ap3/0/1 1-4094

Port Vlans allowed and active in management domain
Gi3/0/1 1,8,10,100
Ap3/0/1 1,8,10,100

Port Vlans in spanning tree forwarding state and not pruned
Gi3/0/1 1,8,10,100
Ap3/0/1 1,8,10,100

Device# show runnning-config interface AppGigabitEthernet 3/0/1

Building configuration...

Current configuration : 64 bytes
!
interface AppGigabitEthernet3/0/1
switchport mode trunk
end
```

ステップ 7 **show controller ethernet-controller AppGigabitEthernet** *interface-number*

ハードウェアから読み込んだ AppGigabitEthernet インターフェイスの送受信に関する統計情報を表示します。

例：

```
Device# show controller ethernet-controller AppGigabitEthernet 1/0/1

Transmit                  AppGigabitEthernet1/0/1   Receive
            0 Total bytes                           0 Total bytes
            0 Unicast frames                        0 Unicast frames
            0 Unicast bytes                         0 Unicast bytes
            0 Multicast frames                      0 Multicast frames
            0 Multicast bytes                       0 Multicast bytes
            0 Broadcast frames                      0 Broadcast frames
            0 Broadcast bytes                       0 Broadcast bytes
            0 System FCS error frames               0 IpgViolation frames
            0 MacUnderrun frames                    0 MacOverrun frames
            0 Pause frames                          0 Pause frames
            0 Cos 0 Pause frames                    0 Cos 0 Pause frames
            0 Cos 1 Pause frames                    0 Cos 1 Pause frames
            0 Cos 2 Pause frames                    0 Cos 2 Pause frames
            0 Cos 3 Pause frames                    0 Cos 3 Pause frames
            0 Cos 4 Pause frames                    0 Cos 4 Pause frames
            0 Cos 5 Pause frames                    0 Cos 5 Pause frames
            0 Cos 6 Pause frames                    0 Cos 6 Pause frames
            0 Cos 7 Pause frames                    0 Cos 7 Pause frames
            0 Oam frames                            0 OamProcessed frames
            0 Oam frames                            0 OamDropped frames
            0 Minimum size frames                   0 Minimum size frames
```

```
0 65 to 127 byte frames
0 128 to 255 byte frames
0 256 to 511 byte frames
0 512 to 1023 byte frames
0 1024 to 1518 byte frames
0 1519 to 2047 byte frames
0 2048 to 4095 byte frames
0 4096 to 8191 byte frames
0 8192 to 16383 byte frames
0 16384 to 32767 byte frame
0 > 32768 byte frames
0 Late collision frames
0 Excess Defer frames
0 Good (1 coll) frames
0 Good (>1 coll) frames
0 Deferred frames
0 Gold frames dropped
0 Gold frames truncated
0 Gold frames successful
0 1 collision frames
0 2 collision frames
0 3 collision frames
0 4 collision frames
0 5 collision frames
0 6 collision frames
0 7 collision frames
0 8 collision frames
0 9 collision frames
0 10 collision frames
0 11 collision frames
0 12 collision frames
0 13 collision frames
0 14 collision frames
0 15 collision frames
0 Excess collision frame
```

```
0 65 to 127 byte frames
0 128 to 255 byte frames
0 256 to 511 byte frames
0 512 to 1023 byte frames
0 1024 to 1518 byte frames
0 1519 to 2047 byte frames
0 2048 to 4095 byte frames
0 4096 to 8191 byte frames
0 8192 to 16383 byte frames
0 16384 to 32767 byte frame
0 > 32768 byte frames
0 SymbolErr frames
0 Collision fragments
0 ValidUnderSize frames
0 InvalidOverSize frames
0 ValidOverSize frames
0 FcsErr frames
```

アプリケーション ホスティングの設定例

次に、アプリケーション ホスティング機能の設定に関するさまざまな例を示します。

例：Cisco IOx の有効化

次に、Cisco IOxを有効にする例を示します。

```
Device> enable
Device# configure terminal
Device(config)# iox
Device(config)# username cisco privilege 15 password 0 ciscoI
Device(config)# end
```

例：前面パネル VLAN ポートのアプリケーション ホスティングの設定

（注） このセクションは、Cisco IOS XE Amsterdam 17.1.1 以降のリリースに適用されます。

次に、前面パネルのVLANポートでアプリケーションホスティングを設定する例を示します。

```
Device# configure terminal
Device(config)# interface AppGigabitEthernet 1/0/1
Device(config-if)# switchport trunk allowed vlan 10-12,20
Device(config-if)# switchport mode trunk
Device(config-if)# exit
Device(config)# app-hosting appid iox_app
Device(config-app-hosting)# app-vnic AppGigabitEthernet trunk
Device(config-config-app-hosting-trunk)# vlan 10 guest-interface 2
Device(config-config-app-hosting-vlan-access-ip)# guest-ipaddress 192.168.0.1
netmask 255.255.255.0
Device(config-config-app-hosting-vlan access-ip)# end
```

例：前面パネルトランクポートのアプリケーションホスティングの設定

次に、前面パネルのトランクポートでアプリケーションホスティングを設定する例を示します。

```
Device# configure terminal
Device(config)# interface AppGigabitEthernet 3/0/1
Device(config-if)# switchport trunk allowed vlan 10-12,20
Device(config-if)# switchport mode trunk
Device(config-if)# exit
Device(config)# app-hosting appid iox_app
Device(config-app-hosting)# app-vnic AppGigabitEthernet trunk
Device(config-config-app-hosting-trunk)# guest-interface 2
Device(config-config-app-hosting-trunk)# end
```

例： disk0: からアプリケーションをインストール

次に、disk0: からアプリケーションをインストールする例を示します。

```
Device> enable
Device# app-hosting install appid iperf3 package disk0:iperf3.tar

Installing package 'disk0:iperf3.tar' for 'iperf3'. Use 'show app-hosting list' for
progress.

Device# show app-hosting list
App id                                   State
---------------------------------------------------------
```

```
iperf3                                    DEPLOYED

Switch#app-hosting activate appid iperf3
iperf3 activated successfully
Current state is: ACTIVATED
Switch#
Switch#show app-hosting list
App id                                    State
---------------------------------------------------------
iperf3                                    ACTIVATED

Switch#app-hosting start appid iperf3
iperf3 started successfully
Current state is: RUNNING
Switch#show app-hosting list
App id                                    State
---------------------------------------------------------
iperf3                                    RUNNING

Device#
```

例：アプリケーションの起動

この例では、アプリケーションを起動する方法を示します。

```
Device> enable
Device# configure terminal
Device(config)# app-hosting appid iox_app
Device(config-app-hosting)# start
Device(config-app-hosting)# end
```

例：アプリケーションのライフサイクル

次に、アプリケーションをインストールおよびアンインストールする例を示します。

```
Device> enable
Device# app-hosting install appid iox_app package usbflash1:my_iox_app.tar.tar
Device# app-hosting activate appid iox_app
Device# app-hosting start appid iox_app
Device# app-hosting stop appid iox_app
Device# app-hosting deactivate appid iox_app
Device# app-hosting uninstall appid iox_app
```

例：Docker ランタイムオプションの設定

この例では、Docker ランタイムオプションを設定する方法を示します。

```
Device> enable
Device# configure terminal
Device(config)# app-hosting appid iox_app
Device(config-app-hosting)# app-resource docker
```

```
Device(config-app-hosting-docker)# run-opts 1 "-v $(APP_DATA):/data"
Device(config-app-hosting-docker)# run-opts 3 "--entrypoint '/bin/sleep 1000000'"
Device(config-app-hosting-docker)# end
```

例：コンテナの静的 IP アドレスの設定

次に、コンテナの静的 IP アドレスを設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# app-hosting appid iox_app
Device(config-app-hosting)# name-server0 10.2.2.2
Device(config-app-hosting)# app-vnic management guest-interface 0
Device(config-app-hosting-mgmt-gateway)# guest-ipaddress 172.19.0.24 netmask 255.255.255.0
Device(config-app-hosting-mgmt-gateway)# exit
Device(config-app-hosting)# app-default-gateway 172.19.0.23 guest-interface 0
Device(config-app-hosting)# end
```

例：管理ポートでのアプリケーション ホスティングの設定

この例では、アプリケーションの IP アドレスを手動で設定する方法を示します。

```
Device# configure terminal
Device(config)# interface gigabitethernet 0/0
Device(config-if)# vrf forwarding Mgmt-vrf
Device(config-if)# ip address 198.51.100.1 255.255.255.254
Device(config-if)# exit
Device(config)# app-hosting appid iox_app
Device(config-app-hosting)# app-vnic management guest-interface 1
Device(config-app-hosting-mgmt-gateway)# end
```

例：アプリケーションのリソース設定の上書き

この例では、アプリケーションのリソース設定を上書きする方法を示します。

```
Device# configure terminal
Device(config)# app-hosting appid iox_app
Device(config-app-hosting)# app-resource profile custom
Device(config-app-resource-profile-custom)# cpu 7400
Device(config-app-resource-profile-custom)# memory 2048
Device(config-app-resource-profile-custom)# vcpu 2
Device(config-app-resource-profile-custom)# end
```

例：ThousandEyes Enterprise Agent のインストール

次の例は、次の方法を示します。

- IOx を有効化する。
- アプリケーション ホスティングを設定する。
- AppGigabitEthernet ポートを設定する。
- ThousandEyes Enterprise Agent をインストールする。

次の例は、IOx を有効化する方法を示しています。

```
Device> enable
Device# configure terminal
Device(config)# iox
Device(config)# username cisco privilege 15 password 0 ciscoI
Device(config)# end
```

次の例は、AppHosting を設定する例を示しています。

```
Device> enable
Device# configure terminal
Device(config)# app-hosting appid appid 1keyes
Device(config-app-hosting)# app-vnic AppGigabitEthernet trunk
Device(config-config-app-hosting-trunk)# vlan 10 guest-interface 2
Device(config-config-app-hosting-vlan-access-ip)# guest-ipaddress 172.19.0.24
netmask 255.255.255.0
Device(config-config-app-hosting-vlan-access-ip)# exit
Device(config-config-app-hosting-trunk)# exit
Device(config-app-hosting)# app-default-gateway 172.19.0.23
guest-interface 0
Device(config-app-hosting)# name-server0 10.2.2.2
Device(config-app-hosting)# app-resource docker
Device(config-app-hosting-docker)# run-opts 1
"-e TEAGENT_ACCOUNT_TOKEN=[account-token]"
Device(config-app-hosting-docker)# prepend-pkg-opts
Device(config-app-hosting-docker)# end
```

次の例は、Appgigabitethernet インターフェイスを設定する方法を示しています。

```
Device> enable
Device# configure terminal
Device(config)# interface AppGigabitEthernet 1/0/1
Device(config-if)# switchport trunk allowed vlan 10-12,20
Device(config-if)# switchport mode trunk
Device(config-if)# end
```

次の例は、ThousandEyes Enterprise Agent をインストールする方法を示しています。

（注） ブラウンフィールドアプリケーションを ThousandEyes Web サイトからダウンロードするか、パッケージ化されたグリーンフィールドアプリケーションをフラッシュファイルシステムからインストールできます。

```
Device> enable
Device# Device# app-hosting install 1keyes https://downloads.thousandeyes.com/
enterprise-agent/thousandeyes-enterprise-agent-3.0.cat9k.tar
OR
Device# app-hosting install appid 1keyes package flash:/apps/[greenfield-app-tar]
Device# app-hosting start appid 1keyes
Device# end
```

ThousandEyes Enterprise Agent の設定例

次に、**show app-hosting detail** コマンドの出力例を示します。

```
Device# show app-hosting detail

App id                 : 1keyes
Owner                  : iox
State                  : RUNNING
Application
  Type                 : docker
  Name                 : thousandeyes/enterprise-agent
  Version              : 3.0
  Description          :
  Path                 : flash:thousandeyes-enterprise-agent-3.0.cat9k.tar
  URL Path             :
Activated profile name : custom

Resource reservation
  Memory               : 0 MB
  Disk                 : 1 MB
  CPU                  : 1850 units
  CPU-percent          : 25 %
  VCPU                 : 1

Attached devices
  Type              Name               Alias
  ---------------------------------------------
  serial/shell     iox_console_shell   serial0
  serial/aux       iox_console_aux     serial1
  serial/syslog    iox_syslog          serial2
  serial/trace     iox_trace           serial3

Network interfaces
   ---------------------------------------
eth0:
   MAC address         : 52:54:dd:c0:a2:ab
   IPv4 address        : 10.0.0.110
   IPv6 address        : ::
   Network name        : mgmt-bridge-v14


Docker
------
Run-time information
  Command              :
  Entry-point          : /sbin/my_init
  Run options in use   : -e TEAGENT_ACCOUNT_TOKEN=TOKEN_NOT_SET --hostname=$(SYSTEM_NAME)
 --cap-add=NET_ADMIN
                          --mount type=tmpfs,destination=/var/log/agent,tmpfs-size=140m
                          --mount
type=tmpfs,destination=/var/lib/te-agent/data,tmpfs-size=200m
                     -v $(APP_DATA)/data:/var/lib/te-agent -e TEAGENT_PROXY_TYPE=DIRECT
```

```
                          -e TEAGENT_PROXY_LOCATION= -e TEAGENT_PROXY_USER= -e
TEAGENT_PROXY_AUTH_TYPE=
                          -e TEAGENT_PROXY_PASS= -e TEAGENT_PROXY_BYPASS_LIST= -e
TEAGENT_KDC_USER=
                         -e TEAGENT_KDC_PASS= -e TEAGENT_KDC_REALM= -e TEAGENT_KDC_HOST=
 -e TEAGENT_KDC_PORT=88
                          -e TEAGENT_KERBEROS_WHITELIST= -e TEAGENT_KERBEROS_RDNS=1 -e
PROXY_APT=
                          -e APT_PROXY_USER= -e APT_PROXY_PASS= -e APT_PROXY_LOCATION=
-e TEAGENT_AUTO_UPDATES=1
                          -e TEAGENT_ACCOUNT_TOKEN=r3d29srpebr4j845lvnamwhswlori2xs
                          --hostname=cat9k-9300-usb --memory=1g
  Package run options  : -e TEAGENT_ACCOUNT_TOKEN=TOKEN_NOT_SET --hostname=$(SYSTEM_NAME)
 --cap-add=NET_ADMIN
                          --mount type=tmpfs,destination=/var/log/agent,tmpfs-size=140m
                          --mount
type=tmpfs,destination=/var/lib/te-agent/data,tmpfs-size=200m
                       -v $(APP_DATA)/data:/var/lib/te-agent -e TEAGENT_PROXY_TYPE=DIRECT

                          -e TEAGENT_PROXY_LOCATION= -e TEAGENT_PROXY_USER= -e
TEAGENT_PROXY_AUTH_TYPE=
                          -e TEAGENT_PROXY_PASS= -e TEAGENT_PROXY_BYPASS_LIST= -e
TEAGENT_KDC_USER=
                         -e TEAGENT_KDC_PASS= -e TEAGENT_KDC_REALM= -e TEAGENT_KDC_HOST=

                          -e TEAGENT_KDC_PORT=88 -e TEAGENT_KERBEROS_WHITELIST= -e
TEAGENT_KERBEROS_RDNS=1
                          -e PROXY_APT= -e APT_PROXY_USER= -e APT_PROXY_PASS= -e
APT_PROXY_LOCATION=
                          -e TEAGENT_AUTO_UPDATES=1
Application health information
  Status               : 0
  Last probe error     :
  Last probe output    :
```

次の **show running-configuration** コマンドの出力例は、静的 IP アドレスの設定を示しています。

```
Device# show running-config | section app-hosting

app-hosting appid 1keyes
 app-vnic AppGigabitEthernet trunk
  vlan 14 guest-interface 0
   guest-ipaddress 10.0.0.110 netmask 255.255.255.0
app-default-gateway 10.0.0.1 guest-interface 0
app-resource docker
  prepend-pkg-opts
  run-opts 1 "-e TEAGENT_ACCOUNT_TOKEN=r3d29srpebr4j845lvnamwhswlori2xs"
  run-opts 2 "--hostname=cat9k-9300-usb --memory=1g"
name-server0 10.0.0.1
start
```

次の **show running-configuration** コマンドの出力例は、静的 IP アドレスの設定とプロキシサーバーの情報を示しています。

```
Device# show running-config | section app-hosting

app-hosting appid 1keyes
 app-vnic AppGigabitEthernet trunk
  vlan 14 guest-interface 0
```

```
  guest-ipaddress 172.27.0.137 netmask 255.240.0.0
 app-default-gateway 172.27.0.129 guest-interface 0
 app-resource docker
  run-opts 1 "-e TEAGENT_ACCOUNT_TOKEN=r3d29srpebr4j845lvnamwhswlori2xs"
 run-opts 3 "-e TEAGENT_PROXY_TYPE=STATIC"
 run-opts 4 "-e TEAGENT_PROXY_LOCATION='proxy-wsa.esl.cisco.com:80'"
 prepend-pkg-opts
 name-server0 172.16.0.2
 start
```

次に、Docker ランタイムオプションとマージされた app-resource Docker パッケージを実行した場合の出力例を示します。

```
// Example of "prepend-package-opts" merging
app-hosting appid TEST
app-vnic management guest-interface 3
app-resource docker
prepend-package-opts !!!
run-opts 1 "--entrypoint '/bin/sleep 1000000'"
run-opts 2 "-e TEST=1 "

# Specify runtime and startup
startup:
runtime_options: "--env MYVAR2=foo --cap-add=NET_ADMIN"

Merged docker run-opts passed to CAF's activation payload:
{"auto_deactivate": false, "resources": {"profile": "custom", "cpu":
"1000", "memory": "1024", "rootfs_size": "0", "vcpu": 1, "disk": 10,"network":
[{"interface-name": "eth3", "network-name": "mgmt-bridge100"}, {"interface-name":
"eth4", "network-type": "vlan", "mode": "static", "ipv4": {"ip": "10.2.0.100",
"prefix": "24", "default": false, "gateway": "" },"network-info": { "vlan-id": "10" },
"mac_forwarding": "no", "mirroring": "no"}, {"interface-name": "eth0",
"network-type": "vlan", "network-info": { "vlan-id": "12" }, "mac_forwarding": "no",
"mirroring": "no"}, {"interface-name": "eth2", "network-type": "vlan", "networkinfo":
{"vlan-id": "22" }, "mac_forwarding": "no", "mirroring": "no"},
{"interface-name
": "eth1", "network-type": "vlan", "network-info": {"vlan-id": "all" },
"mac_forwarding": "no", "mirroring": "no"}]},

"startup":{"runtime_options":"--env MYVAR2=foo --cap-add=NET_ADMIN --
entrypoint'/bin/sleep 1000000' -e TEST=1"}}

// Example of no "prepend-package-opts" which is the current behavior since
16.12 where pkg.yml default runoptions are ignored.
app-hosting appid TEST
app-vnic management guest-interface 3
app-resource docker !!!
run-opts 1 "--entrypoint '/bin/sleep 1000000'"
run-opts 2 "-e TEST=1 "

# Specify runtime and startup
startup:
runtime_options: "--env MYVAR2=foo --cap-add=NET_ADMIN"

Merged docker run-opts passed to CAF's activation payload:
{"auto_deactivate": false, "resources": {"profile": "custom", "cpu":
"1000", "memory": "1024", "rootfs_size": "0", "vcpu": 1, "disk": 10,"network":
[{"interface-name": "eth3", "network-name": "mgmt-bridge100"}, {"interface-name":
"eth4", "network-type": "vlan", "mode": "static", "ipv4": {"ip": "10.2.0.100",
"prefix": "24", "default": false, "gateway": "" },"network-info": { "vlan-id": "10" },
"mac_forwarding": "no", "mirroring": "no"}, {"interface-name": "eth0",
"network-type": "vlan", "network-info": { "vlan-id": "12" }, "mac_forwarding": "no",
"mirroring": "no"}, {"interface-name": "eth2", "network-type": "vlan", "networkinfo":
```

```
{"vlan-id": "22" }, "mac_forwarding": "no", "mirroring": "no"},
{"interface-name": "eth1", "network-type": "vlan", "network-info": {"vlan-id": "all" },
"mac_forwarding": "no", "mirroring": "no"}]},

"startup":{"runtime_options":"--entrypoint '/bin/sleep 1000000' -e
TEST=1"}}


// Config 1 : default behavior when "app-resource docker" is not
configured.
app-hosting appid TEST
app-vnic management guest-interface 3

// Config 2: no docker run-opts specified
app-hosting appid TEST
app-vnic management guest-interface 3
app-resource docker
prepend-package-opts
```

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
プログラマビリティ コマンド	プログラマビリティ コマンド リファレンス
DevNet	https://developer.cisco.com/docs/app-hosting/
Cisco Catalyst 9400 シリーズ スイッチの M2 SATA	*M2 SATA* モジュール
Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチの M2 SATA	*M2 SATA* モジュール
Cisco Catalyst 9600 シリーズ スイッチの M2 SATA	*M2 SATA* モジュール
Cisco Catalyst 9300 シリーズ スイッチの USB3.0 SSD	*USB 3.0 SSD* の設定
Cisco Catalyst 9500 シリーズ スイッチの USB3.0 SSD	*USB 3.0 SSD* の設定
ThousandEyes URL	https://app.thousandeyes.com

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

アプリケーション ホスティングに関する機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *49* : アプリケーション ホスティングに関する機能情報

機能名	リリース	機能情報
アプリケーション ホスティング	Cisco IOS XE Gibraltar 16.12.1 Cisco IOS XE Amsterdam 17.1.1 Cisco IOS XE Amsterdam 17.2.1 Cisco IOS XE Bengaluru 17.5.1 Cisco IOS XE Cupertino 17.7.1	ホステッドアプリケーションはSoftware as a Service（SaaS）ソリューションであり、ユーザはこのソリューションの実行と運用を完全にクラウドから行うことができます。このモジュールでは、アプリケーション ホスティング機能とその有効化の方法について説明します。 • この機能は、Cisco IOS XE Gibraltar 16.12.1 で、Cisco Catalyst 9300 シリーズ スイッチに実装されました。 • この機能は、Cisco IOS XE Amsterdam 17.1.1 で、Cisco Catalyst 9400 シリーズ スイッチに実装されました。 • Cisco IOS XE Amsterdam 17.2.1 では、この機能は Cisco Catalyst 9500 ハイパフォーマンス シリーズ スイッチ、および Cisco Catalyst 9600 シリーズ スイッチに実装されました。 • この機能は、Cisco IOS XE Bengaluru 17.5.1 で、Cisco Catalyst 9410 シリーズ スイッチに実装されました。 • この機能は、Cisco IOS XE Cupertino 17.7.1 で、Cisco Catalyst 9500X シリーズ スイッチに実装されました。

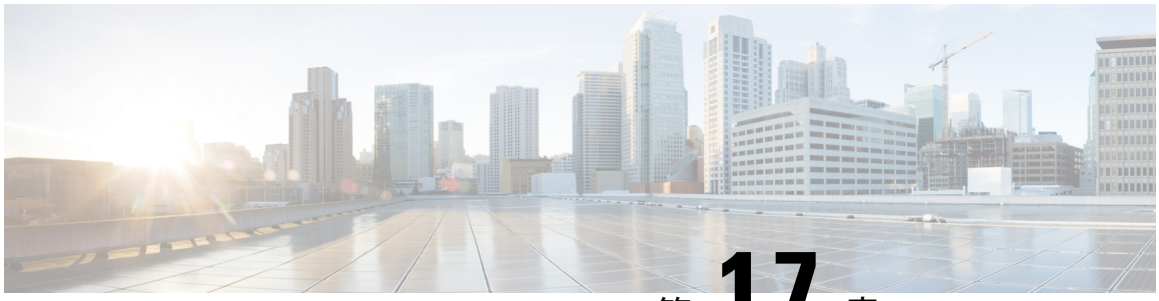
機能名	リリース	機能情報
アプリケーション ホスティング：内部フラッシュから SSD へのアプリケーションの自動転送および自動インストール	Cisco IOS XE Bengaluru 17.6.1	IOxが再起動して別のメディアが選択された場合は、すべてのアプリケーションを新しいメディアに移行し、コンテナを変更前と同じ状態に復元する必要があります。 Cisco IOS XE Bengaluru 17.6.1では、この機能は次のプラットフォームで導入されました。 • Cisco Catalyst 9300 および 9300L シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ
アプリケーション ホスティング：前面パネルのネットワークポートアクセス	Cisco IOS XE Gibraltar 16.12.1 Cisco IOS XE Amsterdam 17.1.1	アプリケーション ホスティング コンテナと前面パネルのネットワークポート間のデータパス接続を導入します。また、前面パネルのネットワークで ZTP 機能が有効になります。 • この機能は、Cisco IOS XE Gibraltar 16.12.1 で、Cisco Catalyst 9300 シリーズ スイッチに実装されました。 • この機能は、Cisco IOS XE Amsterdam 17.1.1 で、Cisco Catalyst 9400 シリーズ スイッチに実装されました。
アプリケーション ホスティング：前面パネルの USB ポートアクセス	Cisco IOS XE Gibraltar 16.12.1 Cisco IOS XE Amsterdam 17.1.1	アプリケーション ホスティング コンテナと前面パネルの USB ポート間のデータパス接続を導入します。 • この機能は、Cisco IOS XE Gibraltar 16.12.1 で、Cisco Catalyst 9300 シリーズ スイッチに実装されました。 • この機能は、Cisco IOS XE Amsterdam 17.1.1 で、Cisco Catalyst 9400 シリーズ スイッチに実装されました。

機能名	リリース	機能情報
アプリケーション ホスティング：ThousandEyes の統合	Cisco IOS XE Amsterdam 17.3.3 Cisco IOS XE Bengaluru 17.5.1 Cisco IOS XE Bengaluru 17.6.1	ThousandEyes は、クラウド対応のエンタープライズ ネットワーク監視ツールであり、ネットワークとサービス全体のエンドツーエンドのビューを提供します。 • この機能は、Cisco IOS XE Amsterdam 17.3.3 で、Cisco Catalyst 9300 および 9300L シリーズ スイッチに実装されました。 • この機能は、Cisco IOS XE Bengaluru 17.5.1 で、Cisco Catalyst 9400 シリーズ スイッチに実装されました。 • この機能は、Cisco IOS XE Bengaluru 17.6.1 で、Cisco Catalyst 9300X シリーズ スイッチに実装されました。 （注） ThousandEyes 統合機能は、Cisco IOS XE Bengaluru 17.4.x リリースではサポートされていません。
ThousandEyes BrowserBot	Cisco IOS XE Bengaluru 17.6.1	ThousandEyes アドオンエージェントモードがサポートされています。アドオンモードは、トランザクションのスクリプト化テスト用の BrowserBot を提供します。 Cisco IOS XE Bengaluru 17.6.1では、この機能は次のプラットフォームで導入されました。 • Cisco Catalyst 9300、9300L、および 9300X シリーズ スイッチ • Cisco Catalyst 9400 シリーズ スイッチ

機能名	リリース	機能情報
ネイティブ Docker コンテナ：アプリケーションの自動再起動	Cisco IOS XE Amsterdam 17.2.1 Cisco IOS XE Bengaluru 17.5.1	アプリケーションの自動再起動機能を使用すると、プラットフォームに導入されたアプリケーションは、システムのスイッチオーバーまたは再起動時に最後に設定された動作状態を維持できます。この機能はデフォルトで有効であり、ユーザが無効にすることはできません。 • この機能は、Cisco IOS XE Amsterdam 17.1.1 で、Cisco Catalyst 9400 シリーズ スイッチに実装されました。 • この機能は、Cisco IOS XE Bengaluru 17.5.1 で、Cisco Catalyst 9410 シリーズ スイッチに実装されました。

第 V 部

OpenFlow



第 17 章

OpenFlow

このモジュールでは、デバイスで OpenFlow を有効化して設定する方法について説明します。

OpenFlow の前提条件

デバイスを OpenFlow モードで起動する必要があります。（OpenFlow モードは、デバイスで **boot mode openflow** コマンドを設定すると有効になります。すべてのポートがこのモードになり、デバイスは通常の Cisco IOS XE 機能をサポートしなくなります）。

OpenFlow の制約事項

- デバイスで OpenFlow モードを有効にする場合は、以前の設定をすべて消去し、フラッシュファイルシステムから *vlan.dat* ファイルと *stby-vlan.dat* ファイルを削除します。
- デバイスが Openflow モードの場合、デバイスが通常モードのときに機能する他のコントロール プレーン プロトコル、ボーダー ゲートウェイ プロトコル（BGP）、スパニングツリープロトコル（STP）、ポートチャネル、StackWise Virtual などを有効にしないでください。

OpenFlow について

機能に関する詳細については、次の各項を参照してください。

OpenFlow の概要

OpenFlow は Open Networking Foundation（ONF）の仕様で、フローベースの転送インフラストラクチャと、標準化されたアプリケーション プログラム インターフェイスを定義します。OpenFlow では、セキュアなチャネルを介して、デバイスのフォワーディング機能を方向付けすることができます。

OpenFlowは、コントローラ（コントロールプレーン）とイーサネットスイッチ（データプレーン）の間のプロトコルです。スイッチには、パイプラインに配置されたフローテーブルがあります。フローとは、これらのテーブルに到達するパケットを調べるためのルールです。

スイッチ上のOpenFlowエージェントは、OpenFlowプロトコルを使用してコントローラと通信します。エージェントは、OpenFlow 1.0（有線プロトコル 0x1）と OpenFlow 1.3（有線プロトコル 0x4）の両方をサポートしています。最大 8 つのコントローラを接続できます。これらの接続はスイッチオーバー後は維持されず、コントローラはエージェントに再接続する必要があります。

Cisco Catalyst 9400 シリーズ スイッチでの OpenFlow の実装はステートレスです。ノンストップフォワーディング（NSF）はサポートされていません。スタンバイのスーパーバイザは、フローデータベースと同期しません。

Openflow コントローラ

Openflow コントローラは、Openflow プロトコルを使用して Openflow スイッチとやり取りするエンティティです。ほとんどの場合、コントローラは多数の Openflow 論理スイッチを管理するソフトウェアです。コントローラではネットワークを一元的に表示でき、管理者はこれを使用して、ネットワーク トラフィックの処理方法について基盤となるシステム（スイッチおよびルータ）に指示を出すことができます。通常、コントローラは Linux サーバで実行され、OpenFlow 対応スイッチに IP 接続できる必要があります。

コントローラはスイッチを管理し、スイッチ上でフローを挿入および削除します。これらのフローは、OpenFlow 1.3 および 1.0 の「照合」と「アクション」の基準のサブセットをサポートしています。

スイッチは、管理ポートを使用してコントローラに接続します。管理ポートは管理用の Virtual Routing and Forwarding（VRF）インスタンスの中にあり、そのためコントローラへのセキュアな接続を提供します。コントローラをスイッチに接続するには、コントローラへの到達が可能な IP アドレスとポート番号を設定します。

フローの管理

フロー エントリは、パケットの照合と処理に使用されるフロー テーブル内の要素です。これには、照合設定の優先順位レベル、パケットを照合するための一連の照合フィールド、適用する一連の命令、パケット カウンタ、およびバイト カウンタが含まれています。また、フローごとにタイムアウト（ハード タイムアウトまたは非アクティブ タイムアウト）も関連付けられており、フローの自動削除に使用されます。

サポートされるフローテーブルは、最大 32 です。

各フローは次の情報を提供します。

- 優先順位：優先順位の高いフローが先に照合されます。フローの更新では、設定された優先順位に基づいて、すべてのフローに優先順位を付ける必要があります。
- 照合フィールド：パケットを照合する際のフローエントリの一部。照合フィールドは、さまざまなパケット ヘッダー フィールドと照合できます。フィールドに照合情報が指定されていない場合は、ワイルドカードが使用されます。
- アクション：パケットに対して作用する操作。

OpenFlow パイプライン

OpenFlow パイプラインは、リンクされたフローテーブルのセットで、OpenFlow スイッチでの照合、転送、およびパケット変更を提供します。ポートは、パケットがパイプラインに出入りする場所です。

パケットは入力ポートで受信され、出力ポートに転送される OpenFlow パイプラインによって処理されます。パケット入力ポートは、パイプライン全体でパケットによって所有され、パケットがスイッチに受信されたポートを表します。入力ポートは、フローの一致フィールドとしても使用できることに注意してください。

フローアクションを使用すると、パケットをパイプライン内の後続のテーブルに送信して処理したり、テーブル間で情報をやり取りしたりすることができます。一致するフローエントリに関連付けられたアクションが次のテーブルを指定しない場合、パイプライン処理は停止します。この時点で、パケットは通常変更され、転送されます。パケットはドロップすることもできます。

OpenFlow スイッチのフローテーブルには、0 から順に番号が付けられます。パイプライン処理は常に、フローテーブル 0 のフローエントリに対してパケットを照合することから始まります。最初のテーブルの一致とアクションの結果に応じて、他のフローテーブルを使用できます。その結果、後続のテーブルのフローエントリとパケットが一致する可能性があります。

サポートされる Match フィールドとアクション

Match フィールドは、パケットヘッダーと入力ポートを含む、パケットが照合されるフィールドです。Match フィールドは、ワイルドカード（任意の値に一致）にすることができ、フィールドの選択されたビットに一致するビットマスクを指定できます。

アクションは、パケットをポートまたは後続のテーブルに転送する操作、またはパケットフィールドを変更する操作です。アクションは、フローエントリに関連付けられた命令の一部、またはグループエントリに関連付けられたアクションバケットとして指定できます。グループエントリは、複数のフローで共有できるアクションの集合です。

1 つ以上のフローエントリで指定されたアクションは、グループアクションと呼ばれる基本アクションにパケットを転送できます。グループアクションの目的は、複数のフロー間で一連のアクションを共有することです。グループは1つ以上のバケットで構成され、バケットは一連のアクション（set、pop、または output）を持つことができます。Cisco Catalyst 9000 シリーズスイッチは、グループタイプ *all* および *indirect* をサポートします。

次に、サポートされる match フィールドとアクションの一覧を示します。

表 *50 :* サポートされる *match* フィールド

ヘッダー フィールド	前提条件	マスク可能なエントリ	値の例
イーサネットの宛先 MAC アドレス	—	あり	01:80:c2:00:00:00/ ff:ff:ff:00:00:00（マスクあり） de:f3:50:c7:e2:b2（マスクなし）
イーサネットの送信元 MAC アドレス	—	あり	0e:00:00:00:00:019（マスクなし）
イーサネットの種類	—	—	ARP（0x0806）、IPv4（0x0800）、IPv6（0x86dd）など
VLAN ID	—		0x13f
ARP ターゲット プロトコル アドレス	イーサネットタイプは 0x0806 に設定する必要があります	あり	—
IP プロトコル	イーサネットタイプは 0x0800 または 0x86dd に設定する必要があります	—	ICMP（0x01）、TCP（0x06）、UDP（0x11）など
IPv4 発信元アドレス	イーサネットタイプは 0x0800 に設定する必要があります	あり	10.0.0.0/24（マスクあり）
IPv4 宛先アドレス	イーサネットタイプは 0x0800 に設定する必要があります	あり	10.0.0.254（マスクなし）

ヘッダー フィールド	前提条件	マスク可能なエントリ	値の例
IPv6 送信元アドレス	イーサネットタイプは 0x08dd に設定する必要があります	あり	2001:DB8::1（マスクなし）
IPv6 宛先アドレス	イーサネットタイプは 0x08dd に設定する必要があります	あり	2001:DB8:0:ABCD::1/48（マスクあり）
ネイバー探索ターゲット	イーサネットタイプは 0x08dd に設定し、IP プロトコルは 0x01 に設定する必要があります	—	ND ターゲット
ICMPv6 タイプ	イーサネットタイプは 0x08dd に設定し、IP プロトコルは 0x01 に設定する必要があります	—	—
UDP/TCP 送信元ポート	イーサネットタイプは 0x0800 または 0x86dd に設定し、プロトコルは 0x06 または 0x11 に設定する必要があります	—	—
UDP/TCP 宛て先ポート	イーサネットタイプは 0x0800 または 0x86dd に設定し、プロトコルは 0x06 または 0x11 に設定する必要があります	—	—
着信インターフェイス	—	—	—

サポートされるアクション

フローは次の宛先にパケットを送信できます。

- コントローラ
- スイッチの任意のインターフェイス（着信インターフェイスを含む）。
- 別のルックアップ用の後続のフローテーブル（テーブル 0 の後）。
- グループ。

- ローカル処理用のスイッチ CPU。ローカル処理のために送信できるのは、Cisco Discovery Protocol および Link Layer Discovery Protocol（LLDP）パケットのみです。

フローによって VLAN タグを追加（push）または削除（pop）できます。パケットが IP パケットの場合は、フローによって存続可能時間（TTL）ヘッダーフィールドの値を減らすことができます。

この機能は、パケットフィールドが *Set-Field* アクションとして定義されるように変更します。フローは、パケットの次のヘッダーフィールドも変更できます。

表 *51* : ヘッダーフィールドでサポートされる書き換え回数

ヘッダー フィールド	スケール
イーサネットの宛先 MAC アドレス	1k
イーサネットの送信元 MAC アドレス	256
VLAN ID	4k

フィールド書き換え

Cisco IOS XE Bengaluru 17.4.1 では、次のフィールドを書き換えるサポートが追加されました。

表 *52* : フィールド書き換え

フィールド	スケール
ipv4_src	4k
ipv4_dst	4k
icmpv4_type	256
tcp_src/udp_src	4k（両方のフィールドで共有）
tcp_dst/udp_dst	4k（両方のフィールドで共有）
ip_dscp	64

IP_DSCP フィールドは、IPv4 タイプオブサービス（ToS）フィールドと IPv6 トラフィック クラス フィールドの一部です。

OpenFlow スケール情報

表 *53 :* サポートされるプラットフォームのスケール情報

	Cisco Catalyst 9300 シリーズ スイッチ	Cisco Catalyst 9400 シリーズスイッチ、および Cisco Catalyst 9500 シリーズ スイッチ	Cisco Catalyst 9500 ハイパフォーマンスシリーズ スイッチ
フローの合計数	18K/9K	54K/27K	54K/27K

フローの操作

ここでは、フローが OpenFlow デバイスでプログラムされるようにコントローラから送信されるときに実行される操作について説明します。

通常デバイスには、パイプラインに配置されたフローテーブルがあります。パイプライン機能情報は、パイプラインの構造を指定します。たとえば、テーブルまたはステージの数、各ステージが実行できる機能（照合またはアクション）、各テーブルのサイズなどがあります。

コントローラがフロー要求を送信すると、OpenFlow エージェントは、ハードウェアがフローを処理できるかどうかを確認します。エージェントは、スイッチの起動時に定義されるハードウェアの機能とフローとを比較します。フローが有効であれば、該当するフローテーブルにプログラムされます。

新しいパイプラインが検証された場合（ハードウェアがパイプラインをサポートできるかどうか）、そのパイプラインは、フローをインストールできるかどうかのチェックに使用される新しい機能セットになります。

パイプラインがインスタンス化され、フローがインストールされると、パケットがスイッチから転送されます。優先順位の最も高い、一致するフローエントリが見つかるまで、入力パケットが各フローテーブル内のフローと照合されます。パケットの照合は、完全一致の場合もあれば（テーブルのすべてのフィールドが正確に一致する）、部分一致の場合もあります（一部または全部のフィールドに一致し、ビットマスクを持つフィールドが部分的に一致する場合がある）。設定されたアクションに基づいて、パケットが変更されるか転送される場合があります。アクションは、パイプライン内でいつでも適用できます。アクションによって、次の照合対象のフローテーブル、パケットの出力ポートのセット、およびパケットをコントローラにルーティングするかどうかが決まる場合があります。

OpenFlow テーブル パイプライン

OpenFlow テーブル機能要求メッセージを使用すると、OpenFlow コントローラから、OpenFlow が管理するデバイスについて既存のフローテーブルの機能を照会したり、指定した設定と一致するようにこれらのテーブルを設定したりできます。

テーブルはすべて、照合およびアクション機能のサブセットを使用して設定できます。テーブルのサイズを実行時に変更することもできます。新しいフローテーブル設定が正常に適用され

ると、古いフロー テーブルのフロー エントリが通知なく削除されます。動的に設定されたフローテーブルは、再起動後は維持されません。デバイスが起動するとデフォルトのパイプラインが起動します。

OpenFlow コントローラからの要求に基づいて新しいフローテーブルを設定している間は、既存のフローの中を流れる進行中のトラフィックがあると、いずれもドロップされます。

ブレークアウトポートのサポート

ブレークアウトポートは、単一の 40G Quad Small Form-Factor Pluggable+（QSFP+）インターフェイスを 4 つの 10G SFP+ インターフェイスに分割し、単一の 100G QSFP28 インターフェイスを 4 つの 25G SFP28 インターフェイスに分割できます。ブレークアウトポートのサポートは、通常モードのブレークアウトポートをサポートするプラットフォームの OpenFlow モードで使用できます。Cisco IOS XE Bengaluru 17.5.1 では、ブレークアウトポートのサポートは、Cisco Catalyst 9500 および9500 ハイパフォーマンス シリーズ スイッチで使用できます。

ブレークアウトポートに関連付けられている OpenFlow ポート番号を表示するには、**show openflow switch 1 ports** コマンドを使用します。ブレークアウト インターフェイス名から OpenFlow ポート番号を計算するための特定のルールはありません。

OpenFlow Power over Ethernet

OpenFlow は Power Over Ethernet（PoE）をサポートします。PoE を機能させるには、デバイスで Cisco Discovery Protocol または LLDP を設定し、Cisco Discovery Protocol パケットまたは LLDP パケットがデバイスによって処理（および送信）されるようにします。PoE を OpenFlow で機能させるために、OpenFlow 固有の設定は必要ありません。

OpenFlow コントローラで、*output-to-local* アクションを使用してフローを設定し、パケットがローカル処理のためにデバイス CPU に送信されるようにします。

PoE の詳細については、「*POE* の設定」の章を参照してください。

OpenFlow の設定方法

ここでは、OpenFlow のさまざまな設定作業について説明します。

デバイスでの OpenFlow モードの有効化

スイッチが通常モードで動作している場合は、以前の設定を削除するように **write erase** コマンドを設定することをお勧めします。

手順の概要

1. **enable**
2. **configure terminal**
3. **boot mode openflow**

4. **exit**
5. **write erase**
6. • **delete flash:***vlan.dat*
   • **delete flash:***stby-vlan.dat*
7. **reload**
8. **enable**
9. **show boot mode**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**boot mode openflow** 例： `Device(config)# boot mode openflow`	OpenFlow 転送モードをイネーブルにします。
ステップ 4	**exit** 例： `Device(config)# exit`	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードを開始します。
ステップ 5	**write erase** 例： `Device# write erase`	NVRAM 内のすべてのファイルを消去します。 • デバイスが以前に通常モードで動作していた場合は、すべてのファイルを消去することをお勧めします。

ステップ 6	• **delete flash:***vlan.dat*
	• **delete flash:***stby-vlan.dat*
	例：
	`Device# delete flash:vlan.dat`
	`Device# delete flash:stby-vlan.dat`
		• VLAN 情報を保存する vlan.dat ファイルを削除します。
		• スタンバイデバイスがある場合は、stby-vlan.dat ファイルを削除します。

ステップ 7	**reload** 例： `Device# reload`	スイッチをリロードし、スイッチの OpenFlow フォワーディング モードを有効にします。

	コマンドまたはアクション	目的
ステップ 8	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 9	**show boot mode** 例： `Device# show boot mode`	デバイスのフォワーディングモードに関する情報を表示します。

例

次の **show boot mode** コマンドの出力例は、デバイスが OpenFlow モードであることを示しています。

```
Device# show boot mode

System initialized in openflow forwarding mode
System configured to boot in openflow forwarding mode
```

次のタスク

通常モードに戻るには、**no boot mode openflow**コマンドを設定して、デバイスをリロードします。

OpenFlow の設定

手順の概要

1. **enable**
2. **configure terminal**
3. **feature openflow**
4. **openflow**
5. **switch 1 pipeline 1**
6. **controller ipv4** *ip-address* **port** *port-number* **vrf** *vrf-name* **security** {**none** | **tls**}
7. **datapath-id** *ID*
8. **tls trustpoint local** *name* **remote** *name*
9. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例：	特権 EXEC モードをイネーブルにします。

	コマンドまたはアクション	目的
	`Device> enable`	• プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**feature openflow** 例： `Device(config)# feature openflow`	OpenFlow 機能をイネーブルにします。
ステップ 4	**openflow** 例： `Device(config)# openflow`	OpenFlow 設定をイネーブルにし、OpenFlow コンフィギュレーション モードを開始します。
ステップ 5	**switch 1 pipeline 1** 例： `Device(config-openflow)# switch 1 pipeline 1`	論理スイッチとパイプラインを設定し、OpenFlow のスイッチ コンフィギュレーション モードを開始します。
ステップ 6	**controller ipv4** *ip-address* **port** *port-number* **vrf** *vrf-name* **security** {**none** | **tls**} 例： `Device(config-openflow-switch)# controller ipv4 10.2.2.2 port 6633 vrf Mgmt-vrf security tls`	コントローラに接続します。 • OpenFlow コントローラの接続セキュリティオプションとして TLS を設定している場合は、**tls trustpoint** コマンドを設定する必要があります。 • OpenFlow コントローラのセキュリティオプションを設定していない場合は、**tls trustpoint** コマンドを設定する必要はありません。
ステップ 7	**datapath-id** *ID* 例： `Device(config-openflow-switch)# datapath-id 0x12345678`	（任意）OpenFlow の論理スイッチ ID を設定します。 • *ID* 引数にはスイッチ ID を指定します。これは 16 進値です。
ステップ 8	**tls trustpoint local** *name* **remote** *name* 例： `Device(config-openflow-switch)# tls trustpoint local trustpoint1 remote trustpoint1`	（任意）OpenFlow スイッチの Transport Layer Security（TLS）トラストポイントを設定します。
ステップ 9	**end** 例： `Device(config-openflow-switch)# end`	OpenFlow スイッチのコンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

OpenFlow モードでのインターフェイスの設定

OpenFlow モードでは、レイヤ 2 またはレイヤ 3 インターフェイスを設定できます。レイヤ 3 インターフェイスを使用する場合は、インターフェイス コンフィギュレーション モードで **no switchport** コマンドを設定します。レイヤ 2 インターフェイスを使用する場合は、次の作業を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **feature openflow**
4. **interface** *type number*
5. **switchport mode trunk**
6. **switchport nonnegotiate**
7. **no keepalive**
8. **spanning-tree bpdufilter enable**
9. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**feature openflow** 例： `Device(config)# feature openflow`	OpenFlow 機能をイネーブルにします。
ステップ 4	**interface** *type number* 例： `Device(config)# interface gigabitethernet 1/0/3`	インターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 5	**switchport mode trunk** 例： `Device(config-if)# switchport mode trunk`	レイヤ 2 スイッチドインターフェイスのトランキングモードをトランクに設定します。
ステップ 6	**switchport nonnegotiate** 例：	装置がこのインターフェイスのネゴシエーション プロトコルに関係しないように指定します。

	コマンドまたはアクション	目的
	`Device(config-if)# switchport nonnegotiate`	
ステップ 7	**no keepalive** 例： `Device(config-if)# no keepalive`	キープアライブ パケットをディセーブルにします。
ステップ 8	**spanning-tree bpdufilter enable** 例： `Device(config-if)# spanning-tree bpdufilter enable`	インターフェイスでブリッジ プロトコル データ ユニット（BPDU）フィルタリングをイネーブルにします。
ステップ 9	**end** 例： `Device(config-if)# end`	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

OpenFlow の確認

OpenFlow の設定を確認するには、次のコマンドを使用します。コマンドはどの順序で使用してもかまいません。

手順の概要

1. **enable**
2. **show openflow hardware capabilities**
3. **show openflow switch 1 controller**
4. **show openflow switch 1 ports**
5. **show openflow switch 1 flows list**

手順の詳細

ステップ 1 enable

特権 EXEC モードを有効にします。

- パスワードを入力します（要求された場合）。

例：

```
Device> enable
```

ステップ 2 show openflow hardware capabilities

OpenFlow デバイスのハードウェア機能を表示します。

例：

```
Device# show openflow hardware capabilities
```

```
  Max Interfaces: 1000
  Aggregated Statistics: YES


  Pipeline ID: 1
    Pipeline Max Flows: 2322
    Max Flow Batch Size: 100
    Statistics Max Polling Rate (flows/sec): 10000
    Pipeline Default Statistics Collect Interval: 5

Flow table ID: 0

    Max Flow Batch Size: 100
    Max Flows: 1022
    Bind Subintfs: FALSE
    Primary Table: TRUE
    Table Programmable: TRUE
    Miss Programmable: TRUE
    Number of goto tables: 1
    Goto table id: 1
    Number of miss goto tables: 1
    Miss Goto table id: 1
    Stats collection time for full table (sec): 1

.
.
.
```

ステップ 3 show openflow switch 1 controller

スイッチに接続されているコントローラに関する情報を表示します。

例：

```
Device# show openflow switch 1 controller

Logical Switch Id: 1
Total Controllers: 1
Controller: 1
10.10.23.200:6633
Protocol: tcp
VRF: Mgmt-vrf
Connected: Yes
Role: Equal
Negotiated Protocol Version: OpenFlow 1.3
Last Alive Ping: 2018-06-04 17:59:20 PDT
state: ACTIVE
sec_since_connect: 50
```

ステップ 4 show openflow switch 1 ports

OpenFlow スイッチのポートに関する情報を表示します。

例：

```
Device# show openflow switch 1 ports
Logical Switch Id: 1
Port    Interface Name   Config-State     Link-State  Features
   1           Gi1/0/1        PORT_UP        LINK_UP  1GB-FD
   2           Gi1/0/2        PORT_UP        LINK_UP  1GB-FD
   3           Gi1/0/3        PORT_UP        LINK_UP  1GB-FD
   4           Gi1/0/4        PORT_UP        LINK_UP  1GB-FD
   5           Gi1/0/5        PORT_UP      LINK_DOWN  1GB-HD
   6           Gi1/0/6        PORT_UP      LINK_DOWN  1GB-HD
```

```
 7           Gi1/0/7        PORT_UP      LINK_DOWN  1GB-HD
 8           Gi1/0/8        PORT_UP      LINK_DOWN  1GB-HD
 9           Gi1/0/9        PORT_UP        LINK_UP  1GB-FD
10          Gi1/0/10        PORT_UP        LINK_UP  1GB-FD
11          Gi1/0/11        PORT_UP        LINK_UP  1GB-FD
12          Gi1/0/12        PORT_UP        LINK_UP  1GB-FD
13          Gi1/0/13        PORT_UP      LINK_DOWN  1GB-HD
14          Gi1/0/14        PORT_UP      LINK_DOWN  1GB-HD
15          Gi1/0/15        PORT_UP      LINK_DOWN  1GB-HD
16          Gi1/0/16        PORT_UP      LINK_DOWN  1GB-HD
17          Gi1/0/17        PORT_UP      LINK_DOWN  1GB-HD
18          Gi1/0/18        PORT_UP      LINK_DOWN  1GB-HD
19          Gi1/0/19        PORT_UP        LINK_UP  1GB-FD
20          Gi1/0/20        PORT_UP        LINK_UP  1GB-FD
21          Gi1/0/21        PORT_UP        LINK_UP  1GB-FD
22          Gi1/0/22        PORT_UP        LINK_UP  1GB-FD
23          Gi1/0/23        PORT_UP      LINK_DOWN  1GB-HD
24          Gi1/0/24        PORT_UP      LINK_DOWN  1GB-HD
25           Gi1/1/1        PORT_UP      LINK_DOWN  1GB-HD
26           Gi1/1/2        PORT_UP      LINK_DOWN  1GB-HD
27           Gi1/1/3        PORT_UP      LINK_DOWN  1GB-HD
28           Gi1/1/4        PORT_UP      LINK_DOWN  1GB-HD
29           Te1/1/1        PORT_UP      LINK_DOWN  10GB-FD
30           Te1/1/2        PORT_UP      LINK_DOWN  10GB-FD
31           Te1/1/3        PORT_UP      LINK_DOWN  10GB-FD
32           Te1/1/4        PORT_UP      LINK_DOWN  10GB-FD
33           Te1/1/5        PORT_UP      LINK_DOWN  10GB-FD
34           Te1/1/6        PORT_UP      LINK_DOWN  10GB-FD
35           Te1/1/7        PORT_UP      LINK_DOWN  10GB-FD
36           Te1/1/8        PORT_UP      LINK_DOWN  10GB-FD
37           Fo1/1/1        PORT_UP      LINK_DOWN  40GB-FD
38           Fo1/1/2        PORT_UP      LINK_DOWN  40GB-FD
39          Twe1/1/1        PORT_UP      LINK_DOWN  10GB-FD
40          Twe1/1/2        PORT_UP      LINK_DOWN  10GB-FD
```

ステップ 5 show openflow switch 1 flows list

OpenFlow のエントリを表示します。

次の出力例は、テーブル 0 で利用可能なフローを示しています。*match any* はテーブル 1 に移動します（「match any」とは、すべてのパケットがテーブル 1 に移動するという意味です）。テーブル 1 では、宛先 MAC アドレス 00:00:01:00:00:01 が照合され、出力ポートが 36 に設定されます。

例：

```
Device# show openflow switch 1 flows list

Logical Switch Id: 1
Total flows: 8

Flow: 1 Match: any Actions: goto_table:1, Priority: 9000, Table: 0, Cookie: 0x1,
Duration: 2382.117s, Packets: 34443, Bytes: 3359315

Flow: 2 Match: any Actions: drop, Priority: 0, Table: 0, Cookie: 0x0,
Duration: 2382.118s, Packets: 294137, Bytes: 28806211

Flow: 3 Match: any Actions: drop, Priority: 0, Table: 1, Cookie: 0x0,
Duration: 2382.118s, Packets: 34443, Bytes: 3359315

Flow: 4 Match: dl_dst=00:00:01:00:00:01 Actions: output:36, Priority: 9000,
```

```
Table: 1, Cookie: 0x1, Duration: 2382.116s, Packets: 0, Bytes: 0
```

OpenFlow の設定例

例：デバイスでの OpenFlow の有効化

次に、OpenFlow を有効にする例を示します。

```
Device> enable
Device# configure terminal
Device(config)# boot mode openflow
Device(config)# exit
Device# write erase
Device# delete flash:vlan.dat
Device# reload
Device> enable
Device# show boot mode
```

例：OpenFlow の設定

次に、OpenFlow を設定する例を示します。

```
Device# configure terminal
Device(config)# feature openflow
Device(config)# openflow
Device(config-openflow)# switch 1 pipeline 1
Device(config-openflow-switch)# controller ipv4 10.2.2.2 port 6633 vrf Mgmt-vrf security
 tls
Device(config-openflow-switch)# datapath-id 0x12345678
Device(config-openflow-switch)# tls trustpoint local trustpoint1 remote trustpoint1
Device(config-openflow-switch)# end
```

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
OpenFlow のコマンド	プログラマビリティ コマンド リファレンス
Open Network Foundation	https://www.opennetworking.org/

関連項目	マニュアル タイトル
Faucet OpenFlow コントローラ	• https://faucet.nz/ • https://docs.faucet.nz/en/latest/
PoE	• Cisco Catalyst 9300 シリーズ スイッチでの「Power over Ethernet の設定」 • Cisco Catalyst 9400 シリーズ スイッチでの「PoE の設定」

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

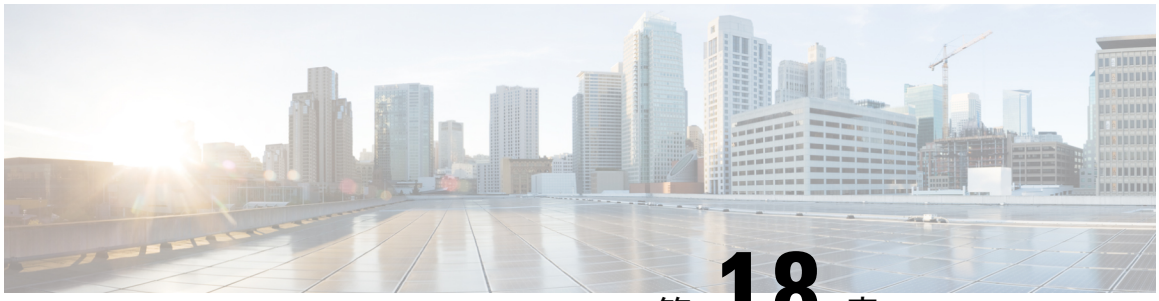
OpenFlow の機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *54 : OpenFlow* の機能情報

機能名	リリース	機能情報
OpenFlow	Cisco IOS XE Fuji 16.9.1	OpenFlow は Software Defined Networking（SDN）の標準規格であり、SDN 環境での通信プロトコルを定義します。これにより、SDN コントローラは、スイッチやルータなどのネットワーク デバイスのフォワーディング プレーンと直接やり取りできるようになります。 この機能は、次のプラットフォームに実装されていました。 • Catalyst 9300 シリーズ スイッチ • Catalyst 9400 シリーズ スイッチ • Catalyst 9500 シリーズ スイッチ • Catalyst 9500 シリーズ ハイ パフォーマンス スイッチ
	Cisco IOS XE Gibraltar 16.10.1	Catalyst 9500 シリーズ ハイ パフォーマンス スイッチでのテーブル機能メッセージのサポートが実装されました。
OpenFlow Power over Ethernet	Cisco IOS XE Gibraltar 16.12.1	PoE は OpenFlow ポートでサポートされます。 この機能は、次のプラットフォームに実装されていました。 • Catalyst 9300 シリーズ スイッチ • Catalyst 9400 シリーズ スイッチ
OpenFlow ブレークアウトポートのサポート	Cisco IOS XE Bengaluru 17.5.1	ブレークアウトケーブルは、単一の 40G Quad Small Form-Factor Pluggable+（QSFP+）インターフェイスを 4 つの 10G SFP+ インターフェイスに分割し、単一の 100G QSFP28 インターフェイスを 4 つの 25G SFP28 インターフェイスに分割できます。 この機能が次のプラットフォームで追加されました。 • Catalyst 9500 および 9500 ハイパフォーマンス シリーズ スイッチ



第 18 章

オープンフローモードでのハイアベイラビリティ

OpenFlow モードのハイアベイラビリティは、ステートフル スイッチオーバー（SSO）およびノンストップ フォワーディング（NSO）をサポートします。SSO は、NSF と連動して、スイッチオーバー後にユーザーがネットワークを使用できない時間を最小限に抑えます。

OpenFlow モードでのハイアベイラビリティの制約事項

- ステートフル スイッチオーバー（SSO）は、Transport Layer Security（TLS）ではサポートされません。
- OpenFlow コントローラで TCP 接続とセキュアソケットレイヤ（SSL）接続の両方を設定することはできません。

OpenFlow について

機能に関する詳細については、次の各項を参照してください。

オープンフローモードでのハイアベイラビリティ

Cisco IOS XE Bengaluru 17.5.1では、Cisco Catalyst 9400 シリーズ スイッチで OpenFlow モードでのハイアベイラビリティをサポートします。シャーシベースのプラットフォームである Cisco Catalyst 9400 シリーズ スイッチは、デュアルスーパーバイザをサポートします。スーパーバイザの一方がアクティブとして機能し、もう一方がスタンバイとして機能します。

この機能が導入される前は、スイッチオーバー時に OpenFlow コントローラがデバイスにインストールされていたすべてのフローを削除してから、転送トラフィックの中断の原因となるすべてのフローを再送信していました。また、スイッチオーバー中に、OpenFlow コントローラ接続がリセットされて再確立され、コントローラがインストールされているすべてのフローを削除していました。

ハイアベイラビリティ機能を使用すると、アクティブなスーパーバイザが OpenFlow コントローラとの接続を確立し、コントローラから送信されるすべてのフローがアクティブなスーパーバイザによってデバイスにプログラムされます。ソフトウェアまたはハードウェアの障害が原因でアクティブスーパーバイザに障害が発生した場合、またはアクティブスーパーバイザからスタンバイスーパーバイザへの手動スイッチオーバーがトリガーされた場合、すべてのフローが保持されます。新しいアクティブスーパーバイザ上の OpenFlow エージェントは OpenFlow コントローラとの TCP セッションを継続し、接続が OpenFlow エージェントによって終了されることはありません。

ステートフル スイッチオーバー

ステートフルスイッチオーバー（SSO）は、ステートフルなプロトコルおよびアプリケーション情報を保持し、スイッチオーバーの間、ユーザーセッション情報を維持します。コントローラによって OpenFlow デバイスに送信されたフローは、アクティブスーパーバイザからスタンバイスーパーバイザへのスイッチオーバー時に保持されるため、コントローラはフローを再インストールする必要がありません。高いシステムアベイラビリティと比較しても、高速なスイッチオーバーを実現します。

デュアルスーパーバイザをサポートするデバイス上で、SSO はスーパーバイザの冗長性を活用してネットワークのアベイラビリティを向上させます。SSO はスーパーバイザの一方をアクティブプロセッサとして設定し、もう一方をスタンバイとして設定したあと、これらの間で重要なステート情報を同期します。2 つのスーパーバイザ間の初回同期後、SSO はこれらの間のステート情報を動的に維持します。

一般的に、SSO は Cisco ノンストップ フォワーディング（NSF）とともに使用されます。

NSF では、スイッチオーバー中も、パケットは OpenFlow コントローラによってプログラムされたフローエントリに基づいて転送されます。

対称ハイアベイラビリティ

対称ハイアベイラビリティは、アクティブな OpenFlow エージェントが OpenFlow コントローラとのOpenFlow TCP 接続を確立する前に、アクティブスーパーバイザとスタンバイスーパーバイザの両方が稼働している状況です。

対称ハイアベイラビリティモードでは、アクティブスーパーバイザとスタンバイスーパーバイザの両方が独立して動作します。アクティブスーパーバイザのみがコントローラと OpenFlow プロトコルメッセージを交換します。アクティブスーパーバイザが OpenFlow コントローラから受信したすべての TCP パケットは、スタンバイスーパーバイザに複製されます。アクティブスーパーバイザとスタンバイスーパーバイザの OpenFlow ハードウェアテーブル設定、グループテーブルエントリ、およびフローエントリが同期されます。

非対称ハイアベイラビリティ

非対称ハイアベイラビリティでは、アクティブな OpenFlow エージェントがコントローラとの OpenFlow TCP 接続を確立した後にのみ、スタンバイスーパーバイザが起動します。スタンバイが起動すると、アクティブスーパーバイザ上のコントローラによってインストールされたフローと、TCP コントローラ接続がスタンバイで同期されません。アクティブスーパーバイザ上のハイ アベイラビリティ プロセスは、コントローラの TCP 接続を同期するために一括同期を実行し、アクティブスーパーバイザにインストールされているフロー、グループ、OpenFlow テーブル機能メッセージをスタンバイスーパーバイザに送信します。次に、スタンバイスーパーバイザの統計情報カウンタが同期されるため、スタンバイはコントローラが送信したパケットの複製を受信できます。

スタンバイスーパーバイザが、アクティブから送信されたテーブル機能メッセージのインストールに失敗すると、スタンバイスーパーバイザはアクティブにこの失敗を通知します。失敗情報を受信すると、アクティブスーパーバイザは以降のスタンバイとの同期を開始しません。アクティブスーパーバイザは、インストールの失敗を一括同期の失敗としてマークし、エラーメッセージをログに記録し、スタンバイスーパーバイザに通知します。スタンバイスーパーバイザは、メッセージを受信するとリロードします。group mod や flow mod が失敗した場合は、同じプロセスが繰り返されます。

統計情報も、一括同期中にアクティブスーパーバイザからスタンバイに同期されます。統計情報は一括同期後に数秒ごとに動的に同期されるため、統計情報の同期の失敗は無視されます。

プローブ間隔

アクティブスーパーバイザは、管理インターフェイス GigabitEthernet 0/0 を介してコントローラとのOpenFlow TCP 接続を維持し、この接続がスタンバイスーパーバイザと同期されます。アクティブスーパーバイザは、設定されたプローブ間隔に基づいてコントローラ接続をプローブします。

スイッチオーバー後、新しいアクティブの管理インターフェイスが動作可能になるまでに少なくとも 13 秒かかります。それまでにコントローラによって送信されたパケットは受信されず、OpenFlow TCP 接続が切断される可能性があります。プローブ間隔による OpenFlow エージェントのタイムアウトを回避するために、アクティブスーパーバイザではデフォルト値の 40 秒が自動的に設定されます。

OpenFlow モードでのハイアベイラビリティの設定方法

OpenFlow モードでのハイアベイラビリティの設定

手順の概要

1. **enable**
2. **configure terminal**

3. **openflow**
4. **switch 1 pipeline 1**
5. **controller ipv4** *ip-address* **port** *port-number* **vrf** *vrf-name*
6. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	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 3	**openflow** 例： `Device(config)# openflow`	OpenFlow設定をイネーブルにし、OpenFlow コンフィギュレーション モードを開始します。
ステップ 4	**switch 1 pipeline 1** 例： `Device(config-openflow)# switch 1 pipeline 1`	論理スイッチとパイプラインを設定し、OpenFlow のスイッチ コンフィギュレーション モードを開始します。
ステップ 5	**controller ipv4** *ip-address* **port** *port-number* **vrf** *vrf-name* 例： `Device(config-openflow-switch)# controller ipv4 10.2.2.2 port 6633 vrf Mgmt-vrf`	OpenFlow コントローラに接続します。 （注） TLS ではハイアベイラビリティはサポートされません。
ステップ 6	**end** 例： `Device(config-openflow-switch)# end`	OpenFlow スイッチのコンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

OpenFlow モードでのハイアベイラビリティの設定例

例：OpenFlow モードでのハイアベイラビリティの設定

次に、OpenFlow を設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# openflow
Device(config-openflow)# switch 1 pipeline 1
Device(config-openflow-switch)# controller ipv4 10.2.2.2 port 6633 vrf Mgmt-vrf
```

```
Device(config-openflow-switch)# end
```

ハイアベイラビリティ設定の確認

次に、**show openflow switch** *switch-number* **controller** コマンドの出力例を示します。出力フィールドで、Connected が Yes、state が ACTIVE である必要があり、Negotiated Protocol Version はスタンバイスーパーバイザの値と同じである必要があります。

```
Device# show openflow switch 1 controller

Logical Switch Id: 1
Total Controllers: 1

  Controller: 1
    172.16.18.85:6636
    Protocol: tcp
    VRF: Mgmt-vrf
    Connected: Yes
    Role: Equal
    Negotiated Protocol Version: OpenFlow 1.3
    Last Alive Ping: 2021-01-29 08:44:59 UTC
    state: ACTIVE
    sec_since_connect: 4893
```

次に、**show tcp ha connection** コマンドの出力例を示します。アクティブとスタンバイの両方のスーパーバイザで、state が ESTAB と表示されている必要があります。

```
Device# show tcp ha connection

 SSO enabled for 1 connections
TCB           Local Address         Foreign Address       (state)    Conn Id
7F53B1ADE1E0  172.21.18.87.23401    172.16.18.85.6636      ESTAB      1
```

OpenFlow モードでのハイアベイラビリティの機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェアリリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 *55 : OpenFlow* モードでのハイアベイラビリティの機能情報

機能名	リリース	機能情報
オープンフローモードでのハイアベイラビリティ	Cisco IOS XE Bengaluru 17.5.1	OpenFlow モードでのハイアベイラビリティは、SSO および NSO をサポートします。 Cisco IOS XE Bengaluru 17.5.1では、この機能は次のプラットフォームで導入されました。 • Catalyst 9400 シリーズ スイッチ

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