



Cisco IOS XE Amsterdam 17.3.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：2020年7月31日

最終更新：2023年4月13日

シスコシステムズ合同会社

〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー

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

お問い合わせ先：シスコ コンタクトセンター

0120-092-255（フリーコール、携帯・PHS含む）

電話受付時間：平日 10:00～12:00、13:00～17:00

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

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

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

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

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

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

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

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

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

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

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

© 2020 Cisco Systems, Inc. All rights reserved.



目次

Full Cisco Trademarks with Software License ?

第 1 章

新機能および変更された機能に関する情報 1

新機能および変更された機能に関する情報 1

第 1 部 :

プロビジョニング 33

第 2 章

ゼロ タッチ プロビジョニング 35

ゼロタッチプロビジョニングの制約事項 35

ゼロ タッチ プロビジョニングについて 35

ゼロ タッチ プロビジョニングの概要 35

ゼロ タッチ プロビジョニングのための DHCP サーバの設定 36

DHCPv6 のサポート 36

ゼロ タッチ プロビジョニングの構成例 37

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 37

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 38

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 38

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 38

Linux Ubuntu デバイス上でのサンプル DHCP サーバの構成 39

TFTP コピーを使用する管理ポートでの DHCPv6 サーバ設定の例 39

サンプルの Python プロビジョニング スクリプト 40

Cisco 4000 シリーズ サービス統合型ルータの起動ログ 40

Cisco Catalyst 9000 シリーズ スイッチの起動ログ 42

ゼロ タッチ プロビジョニングの機能情報 66

第 3 章

iPXE 71

- iPXE について 71
 - iPXE について 71
 - iPXE の概要 72
 - IPv6 iPXE ネットワーク ブート 75
 - ROMmon モードでの IPv6 アドレスの割り当て 77
 - サポートされる ROMmon 変数 78
 - iPXE がサポートする DHCP オプション 78
 - DHCPv6 固有識別子 80
- iPXE の設定方法 81
 - iPXE の設定 81
 - デバイス ブートの設定 82
- iPXE の設定例 83
 - 例 : iPXE 構成 83
 - サンプルの iPXE ブート ログ 83
 - iPXE 用のサンプル DHCPv6 サーバ構成 84
 - iPXE のトラブルシューティングのヒント 85
 - iPXE に関する追加情報 87
 - iPXE の機能情報 87

第 II 部 :

シェルとスクリプト化 91

第 4 章

ゲスト シェル 93

- ゲストシェルの制約事項 93
- ゲスト シェルについて 93
 - ゲスト シェルの概要 93
 - ゲストシェルのソフトウェア要件 94
 - ゲストシェルのセキュリティ 95
 - ゲストシェルのハードウェア要件 95
 - ゲストシェルのストレージ要件 96

ゲスト シェルの有効化と実行	97
ゲスト シェルの無効化と破棄	98
デバイスでのゲスト シェルへのアクセス	98
管理ポートを介してのゲスト シェルへのアクセス	98
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	99
ゲスト シェルでのスタッキング	99
Cisco IOx の概要	99
IOx のトレースとロギングの概要	100
IOXMAN 構造体	100
ロギングとトレースのシステム フロー	101
メッセージのロギングとトレース	103
ゲスト シェルを有効にする方法	105
IOx の管理	105
ゲスト シェルの管理	106
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	108
ゲストシェルの AppGigabitEthernet インターフェイスの設定	109
管理インターフェイスでのゲストシェルの有効化	112
Python インタープリタのアクセス	113
ゲスト シェルの設定例	114
例：ゲスト シェルの管理	114
VirtualPortGroup 設定の例	115
例：ゲストシェルの AppGigabitEthernet インターフェイスの設定	116
例：管理インターフェイスでのゲストシェルの有効化	117
例：ゲスト シェルの使用	117
例：ゲスト シェルのネットワーキング設定	118
ゲスト シェルの DNS 設定の例	118
例：プロキシ環境変数の設定	118
例：プロキシ設定用の Yum および PIP の構成	119
ゲスト シェルに関するその他の参考資料	119
ゲスト シェルの機能情報	120

第 5 章	Python API 125
	Python の使用 125
	Cisco Python モジュール 125
	IOS CLI コマンドを実行するための Cisco Python モジュール 127

第 6 章	CLI Python モジュール 131
	Python CLI モジュールについて 131
	Python について 131
	Python スクリプトの概要 131
	対話形式の Python プロンプト 131
	Python スクリプト 132
	サポートされる Python のバージョン 133
	Cisco CLI Python モジュールの更新 135
	CLI Python モジュールに関するその他の参考資料 135
	CLI Python モジュールの機能情報 136

第 7 章	EEM Python モジュール 139
	EEM Python モジュールの前提条件 139
	EEM Python モジュールについて 139
	EEM の Python スクリプト 139
	EEM Python パッケージ 140
	Python がサポートする EEM アクション 140
	EEM 変数 141
	EEM CLI ライブラリのコマンド拡張 141
	EEM Python ポリシーの設定方法 142
	Python ポリシーの登録 142
	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 144
	EEM アプレットでの Python スクリプトの追加 146
	EEM Python モジュールに関するその他の参考資料 148
	EEM Python モジュールの機能情報 148

第 III 部 :	モデル駆動型プログラマビリティ	151
第 8 章	NETCONF プロトコル	153
	NETCONF プロトコルの概要	153
	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	153
	NETCONF	154
	NETCONF プロトコルの制約事項	154
	NETCONF RESTCONF IPv6 のサポート	155
	NETCONF グローバル セッションのロック	155
	NETCONF Kill セッション	156
	NETCONF-YANG SSH サーバのサポート	156
	候補コンフィギュレーションのサポート	157
	候補の NETCONF 操作	157
	確認済み候補コンフィギュレーションのコミット	159
	候補サポートの設定	161
	NETCONF プロトコルの設定方法	161
	NETCONF を使用するための権限アクセスの提供	161
	NETCONF-YANG の設定	163
	NETCONF オプションの設定	164
	SNMP の設定	164
	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	166
	NETCONF プロトコルのコンフィギュレーションの確認	167
	NETCONF プロトコルの関連資料	169
	NETCONF プロトコルの機能情報	170
第 9 章	RESTCONF プロトコル	181
	RESTCONF プロトコルの前提条件	181
	RESTCONF プロトコルの制約事項	181
	RESTCONF プロトコルに関する情報	182
	RESTCONF の概要	182

HTTPs メソッド	182
RESTCONF ルート リソース	183
RESTCONF API リソース	184
メソッド	184
RESTCONF YANG パッチのサポート	185
RESTCONF プロトコルの設定方法	189
AAA を使用した NETCONF/RESTCONF の認証	189
RESTCONF の Cisco IOS HTTP サービスの有効化	191
RESTCONF の設定の検証	192
RESTCONF プロトコルの設定例	194
例：RESTCONF プロトコルの設定	194
RESTCONF プロトコルの関連資料	197
RESTCONF プロトコルの機能情報	198

第 10 章

NETCONF および RESTCONF のサービスレベル ACL	203
NETCONF および RESTCONF のサービスレベル ACL に関する情報	203
NETCONF および RESTCONF のサービスレベル ACL の概要	203
NETCONF および RESTCONF のサービスレベル ACL の設定方法	204
NETCONF-YANG セッションの ACL の設定	204
RESTCONF セッションの ACL の設定	205
NETCONF および RESTCONF のサービスレベル ACL の設定例	207
例：NETCONF セッションの ACL の設定	207
例：RESTCONF セッションの ACL の設定	207
NETCONF および RESTCONF のサービスレベル ACL に関するその他の資料	207
NETCONF および RESTCONF のサービスレベル ACL の機能情報	208

第 11 章

gNMI プロトコル	211
gNMI プロトコルの制約事項	211
gNMI プロトコルの概要	212
gNMI について	212
YANG データ ツリーの JSON IETF エンコーディング	212

gNMI GET Request	213
gNMI SetRequest	216
gNMI の名前空間	218
gNMI のワイルドカード	219
gNMI 設定の永続化	222
gNMI ユーザ名とパスワードによる認証	223
gNMI のエラー メッセージ	223
gNMI プロトコルを有効にする方法	223
Linux での OpenSSL を使用した証明書の作成	224
CLI によるデバイスへの証明書のインストール	224
非セキュア モードでの gNMI の有効化	225
セキュア モードでの gNMI の有効化	227
gNMI クライアントの接続	228
gNMI プロトコルの設定例	229
例：非セキュア モードでの gNMI の有効化	229
例：セキュア モードでの gNMI の有効化	230
gNMI プロトコルの関連資料	230
gNMI プロトコルの機能情報	231
<hr/>	
第 12 章	gRPC ネットワーク操作インターフェイス 235
	gRPC ネットワーク操作インターフェイスに関する情報 235
	gNOI プロトコル 235
	証明書管理サービス 236
	Install RPC 236
	Rotate RPC 239
	Revoke RPC 240
	GetCertificate RPC 240
	CanGenerateCSR RPC 241
	相互認証 242
	証明書サービスによるブートストラップ 243
	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 243
	gRPC ネットワーク操作インターフェイスの機能情報 244

第 13 章

モデルベースの AAA 247

- モデルベースの AAA 247
 - モデルベースの AAA の前提条件 247
 - 初期操作 247
 - グループ メンバーシップ 248
 - NACM 権限レベルの依存関係 249
 - NACM の設定の管理と保守 249
 - NACM 設定のリセット 250
 - NACM の設定例 250
- モデルベースの AAA に関するその他の参考資料 253
- モデルベースの AAA に関する機能情報 254

第 14 章

モデル駆動型テレメトリ 257

- モデル駆動型テレメトリ 257
 - モデル駆動型テレメトリの前提条件 257
 - モデル駆動型テレメトリの制約事項 260
 - モデル駆動型テレメトリについて 261
 - モデル駆動型テレメトリの概要 261
 - テレメトリ ロール 261
 - サブスクリプションの概要 261
 - サブスクリプションのモニタリング 276
 - ストリーム 278
 - TLDP 変更時の通知 282
 - トランスポート プロトコル 282
 - テレメトリにおけるハイ アベイラビリティ 283
 - サンプルのモデル駆動型テレメトリ RPC 284
 - 設定済みサブスクリプションの管理 284
 - 応答コードの受信 287
 - NETCONF ダイアルインのサブスクリプションプッシュ更新の受信 287
 - サブスクリプションの詳細の取得 288

モデル駆動型テレメトリに関するその他の参考資料 290

モデル駆動型テレメトリの機能情報 291

第 15 章

In-Service Model Update 303

In-Service Model Update の制約事項 303

In-Service Model Update について 303

In-Service Model Update の概要 303

In-Service Model Update パッケージの互換性 304

更新プログラム パッケージの命名規則 304

更新プログラム パッケージのインストール 305

更新プログラム パッケージの非アクティブ化 305

更新プログラム パッケージのロールバック 306

In-Service Model Update の管理方法 306

更新プログラム パッケージの管理 306

In-Service Model Update の設定例 308

例：更新プログラム パッケージの管理 308

In-Service Model Update の機能情報 312

第 IV 部 :

アプリケーション ホスティング 315

第 16 章

アプリケーション ホスティング 317

アプリケーション ホスティングの制約事項 317

アプリケーション ホスティングに関する情報 318

アプリケーション ホスティングの必要性 318

Cisco IOx の概要 318

アプリケーション ホスティングの概要 319

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 320

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 320

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 321

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 321

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 322

ThousandEyes Enterprise Agent の概要	322
ThousandEyes Enterprise Agent の前提条件	323
ThousandEyes Enterprise Agent に必要なリソース	323
ThousandEyes Enterprise Agent のダウンロード	324
ネイティブ Docker コンテナ : アプリケーションの自動再起動	325
アプリケーションの自動再起動のシナリオ	325
異なるプラットフォームでのアプリケーションの自動再起動	327
サポート対象ネットワークタイプ	327
仮想ネットワーク インターフェイス カード	329
ハイ アベイラビリティ	329
アプリケーション ホスティングの設定方法	329
Cisco IOx の有効化	329
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	330
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	332
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	334
アプリケーションのライフサイクル	335
Docker ランタイムオプションの設定	336
コンテナの静的 IP アドレスの設定	337
管理ポートでのアプリケーション ホスティングの設定	339
アプリケーションの IP アドレスの手動設定	340
アプリケーションのリソース設定の上書き	341
ThousandEyes Enterprise Agent のインストール	342
ThousandEyes Enterprise Agent のアプリケーション ホスティングの設定	343
ThousandEyes Enterprise Agent の AppGigabitEthernet インターフェイスの設定	345
ThousandEyes Enterprise Agent のインストール	346
アプリケーション ホスティング設定の確認	347
アプリケーション ホスティングの設定例	350
例 : Cisco IOx の有効化	350
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	351
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	351
例 : disk0: からアプリケーションをインストール	351

例：アプリケーションの起動	352
例：アプリケーションのライフサイクル	352
例：Docker ランタイムオプションの設定	352
例：コンテナの静的 IP アドレスの設定	353
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	353
例：アプリケーションのリソース設定の上書き	353
例：ThousandEyes Enterprise Agent のインストール	354
ThousandEyes Enterprise Agent の設定例	355
その他の参考資料	358
アプリケーション ホスティングに関する機能情報	359

 第 V 部：

OpenFlow 365

 第 17 章

OpenFlow 367

OpenFlow の前提条件	367
OpenFlow の制約事項	367
OpenFlow について	368
OpenFlow の概要	368
Openflow コントローラ	368
フローの管理	369
OpenFlow パイプライン	369
サポートされる Match フィールドとアクション	369
OpenFlow スケール情報	372
フローの操作	372
OpenFlow テーブル パイプライン	373
OpenFlow Power over Ethernet	373
OpenFlow の設定方法	373
デバイスでの OpenFlow モードの有効化	374
OpenFlow の設定	375
OpenFlow モードでのインターフェイスの設定	377
OpenFlow の確認	378

OpenFlow の設定例 381

- 例：デバイスでの OpenFlow の有効化 381
- 例：OpenFlow の設定 381

その他の参考資料 382

OpenFlow の機能情報 382



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 \(1 ページ\)](#)

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	

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	

機能	リリースとプラットフォーム
	<p>Cisco IOS XE Everest 16.5.1a</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Everest 16.5.1b</p> <ul style="list-style-type: none"> • Cisco 4000 シリーズ サービス統合型ルータ <p>Cisco IOS XE Everest 16.6.2</p> <ul style="list-style-type: none"> • Cisco Catalyst 9400 シリーズ スイッチ <p>Cisco IOS XE Fuji 16.7.1</p> <ul style="list-style-type: none"> • 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) <p>Cisco IOS XE Fuji 16.8.1a</p> <ul style="list-style-type: none"> • Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ <p>Cisco IOS XE Fuji 16.8.2</p> <ul style="list-style-type: none"> • 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) <p>Cisco IOS XE Gibraltar 16.12.1</p> <ul style="list-style-type: none"> • Cisco Catalyst 9200 シリーズ スイッチ <p>(注) この機能は C9200L SKU ではサポートされていません。</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Amsterdam 17.2.1</p> <ul style="list-style-type: none"> • Cisco Cloud Services Router 1000V シリーズ

機能	リリースとプラットフォーム
	<ul style="list-style-type: none"> • 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) <p>Cisco IOS XE Amsterdam 17.3.1</p> <ul style="list-style-type: none"> • 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
ゼロタッチプロビジョニング : HTTP ダウンロード	<p>Cisco IOS XE Fuji 16.8.1</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Fuji 16.8.1a</p> <ul style="list-style-type: none"> • 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	<p>Cisco IOS XE Fuji 16.9.1</p> <ul style="list-style-type: none"> • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ <p>Cisco IOS XE Amsterdam 17.3.2a</p> <ul style="list-style-type: none"> • 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	リリースとプラットフォーム
iPXE	<p>Cisco IOS XE Denali 16.3.2 および Cisco IOS XE Everest 16.5.1a</p> <ul style="list-style-type: none">• Cisco Catalyst 3650 シリーズ スイッチ• Cisco Catalyst 3650 シリーズ スイッチ <p>Cisco IOS XE Everest 16.6.1</p> <ul style="list-style-type: none">• Cisco Catalyst 9300 シリーズ スイッチ• Cisco Catalyst 9500 シリーズ スイッチ <p>Cisco IOS XE Everest 16.6.2</p> <ul style="list-style-type: none">• Cisco Catalyst 9400 シリーズ スイッチ <p>Cisco IOS XE Fuji 16.8.1a</p> <ul style="list-style-type: none">• Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ <p>Cisco IOS XE Fuji 16.9.2</p> <ul style="list-style-type: none">• Cisco Catalyst 9200 シリーズ スイッチ <p>Cisco IOS XE Gibraltar 16.11.1</p> <ul style="list-style-type: none">• Cisco Catalyst 9600 シリーズ スイッチ
シェルとスクリプト化	

機能	リリースとプラットフォーム
ゲスト シェル	

機能	リリースとプラットフォーム
	<p>Cisco IOS XE Everest 16.5.1a</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Everest 16.5.1b</p> <ul style="list-style-type: none"> • Cisco 4000 シリーズ サービス統合型ルータ <p>Cisco IOS XE Everest 16.6.2</p> <ul style="list-style-type: none"> • Cisco Catalyst 9400 シリーズ スイッチ <p>Cisco IOS XE Fuji 16.7.1</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Fuji 16.8.1</p> <ul style="list-style-type: none"> • Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ <p>Cisco IOS XE Fuji 16.9.1</p> <ul style="list-style-type: none"> • Cisco 1000 シリーズ サービス統合型ルータ <p>Cisco IOS XE Gibraltar 16.11.1b</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Gibraltar 16.12.1</p> <ul style="list-style-type: none"> • Cisco Catalyst 9200 シリーズ スイッチ <p>(注) この機能は C9200L SKU ではサポートされていません。</p> <ul style="list-style-type: none"> • Cisco Catalyst 9300L SKU • Cisco Catalyst 9600 シリーズ スイッチ <p>Cisco IOS XE Amsterdam 17.3.1</p>

機能	リリースとプラットフォーム
	<ul style="list-style-type: none"> • 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	<p>Cisco IOS XE Amsterdam 17.1.1</p> <ul style="list-style-type: none"> • 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
Python API	<p>Cisco IOS XE Everest 16.5.1a</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Everest 16.5.1b</p> <ul style="list-style-type: none"> • Cisco 4000 シリーズ サービス統合型ルータ <p>Cisco IOS XE Everest 16.6.2</p> <ul style="list-style-type: none"> • Cisco Catalyst 9400 シリーズ スイッチ <p>Cisco IOS XE Fuji 16.7.1</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Fuji 16.8.1a</p> <ul style="list-style-type: none"> • 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
Python CLI モジュール	<p data-bbox="841 296 1166 321">Cisco IOS XE Everest 16.5.1a</p> <ul data-bbox="873 344 1333 541" style="list-style-type: none"><li data-bbox="873 344 1333 369">• Cisco Catalyst 3650 シリーズ スイッチ<li data-bbox="873 401 1333 426">• Cisco Catalyst 3850 シリーズ スイッチ<li data-bbox="873 457 1333 483">• Cisco Catalyst 9300 シリーズ スイッチ<li data-bbox="873 514 1333 539">• Cisco Catalyst 9500 シリーズ スイッチ <p data-bbox="841 579 1166 604">Cisco IOS XE Everest 16.5.1b</p> <ul data-bbox="873 627 1393 653" style="list-style-type: none"><li data-bbox="873 627 1393 653">• Cisco 4000 シリーズ サービス統合型ルータ <p data-bbox="841 693 1154 718">Cisco IOS XE Everest 16.6.2</p> <ul data-bbox="873 741 1333 766" style="list-style-type: none"><li data-bbox="873 741 1333 766">• Cisco Catalyst 9400 シリーズ スイッチ <p data-bbox="841 806 1114 831">Cisco IOS XE Fuji 16.7.1</p> <ul data-bbox="873 854 1487 1014" style="list-style-type: none"><li data-bbox="873 854 1487 959">• 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)<li data-bbox="873 989 1393 1014">• Cisco Cloud Services Router 1000V シリーズ <p data-bbox="841 1054 1114 1079">Cisco IOS XE Fuji 16.8.1</p> <ul data-bbox="873 1102 1516 1299" style="list-style-type: none"><li data-bbox="873 1102 1516 1169">• 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<li data-bbox="873 1199 1516 1299">• 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) <p data-bbox="841 1339 1130 1365">Cisco IOS XE Fuji 16.8.1a</p> <ul data-bbox="873 1388 1503 1455" style="list-style-type: none"><li data-bbox="873 1388 1503 1455">• 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	<p>Cisco IOS XE Everest 16.5.1a</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Everest 16.5.1b</p> <ul style="list-style-type: none"> • Cisco 4000 シリーズ サービス統合型ルータ <p>Cisco IOS XE Everest 16.6.2</p> <ul style="list-style-type: none"> • Cisco Catalyst 9400 シリーズ スイッチ <p>Cisco IOS XE Fuji 16.7.1</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Fuji 16.8.1a</p> <ul style="list-style-type: none"> • 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	

機能	リリースとプラットフォーム
NETCONF プロトコル	

機能	リリースとプラットフォーム
	<p>Cisco IOS XE Denali 16.3.1</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Everest 16.5.1a</p> <ul style="list-style-type: none"> • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ <p>Cisco IOS XE Everest 16.6.2</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Fuji 16.8.1a</p> <ul style="list-style-type: none"> • 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 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 <p>Cisco IOS XE Fuji 16.9.2</p> <ul style="list-style-type: none"> • Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300L SKU <p>Cisco IOS XE Gibraltar 16.10.1</p> <ul style="list-style-type: none"> • 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

機能	リリースとプラットフォーム
	<p>Cisco IOS XE Gibraltar 16.11.1</p> <ul style="list-style-type: none"> • Cisco Catalyst 9600 シリーズ スイッチ <p>Cisco IOS XE Gibraltar 16.12.1</p> <ul style="list-style-type: none"> • Cisco Catalyst 9800-L ワイヤレスコントローラ <p>Cisco IOS XE Amsterdam 17.3.1</p> <ul style="list-style-type: none"> • 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
NETCONF および RESTCONF IPv6 のサポート	<p>Cisco IOS XE Fuji 16.8.1a</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Gibraltar 16.11.1</p> <ul style="list-style-type: none"> • 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
NETCONF グローバルロックおよびセッションの kill	<p>Cisco IOS XE Fuji 16.8.1a</p> <ul style="list-style-type: none"> • 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

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	<p data-bbox="841 296 1179 323">Cisco IOS XE Gibraltar 16.12.1</p> <ul data-bbox="873 344 1503 1346" style="list-style-type: none"><li data-bbox="873 344 1393 371">• Cisco 1000 シリーズ サービス統合型ルータ<li data-bbox="873 401 1393 428">• Cisco 4000 シリーズ サービス統合型ルータ<li data-bbox="873 457 1503 520">• Cisco ASR 900 シリーズ アグリゲーション サービスルータ<li data-bbox="873 550 1503 613">• Cisco ASR 920 シリーズ アグリゲーション サービスルータ<li data-bbox="873 642 1487 669">• Cisco ASR 1000 アグリゲーション サービスルータ<li data-bbox="873 699 1333 726">• Cisco Catalyst 3650 シリーズ スイッチ<li data-bbox="873 756 1333 783">• Cisco Catalyst 3850 シリーズ スイッチ<li data-bbox="873 812 1333 840">• Cisco Catalyst 9200 シリーズ スイッチ<li data-bbox="873 869 1333 896">• Cisco Catalyst 9300 シリーズ スイッチ<li data-bbox="873 926 1333 953">• Cisco Catalyst 9400 シリーズ スイッチ<li data-bbox="873 982 1333 1010">• Cisco Catalyst 9500 シリーズ スイッチ<li data-bbox="873 1039 1333 1066">• Cisco Catalyst 9600 シリーズ スイッチ<li data-bbox="873 1096 1516 1123">• Cisco Catalyst 9800 シリーズ ワイヤレス コントローラ<li data-bbox="873 1152 1471 1180">• Cisco cBR-8 コンバージドブロードバンドルータ<li data-bbox="873 1209 1393 1236">• Cisco Cloud Services Router 1000V シリーズ<li data-bbox="873 1266 1443 1293">• Cisco Network Convergence System 520 シリーズ<li data-bbox="873 1323 1458 1350">• Cisco Network Convergence System 4200 シリーズ

機能	リリースとプラットフォーム
RESTCONF プロトコル	

機能	リリースとプラットフォーム
	<p>Cisco IOS XE Everest 16.6.1</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Fuji 16.8.1a</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Fuji 16.9.2</p> <ul style="list-style-type: none"> • Cisco Catalyst 9200 シリーズ スイッチ <p>Cisco IOS XE Gibraltar 16.11.1</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Gibraltar 16.12.1</p> <ul style="list-style-type: none"> • Cisco Catalyst 9800-L ワイヤレスコントローラ <p>Cisco IOS XE Amsterdam 17.3.1</p>

機能	リリースとプラットフォーム
	<ul style="list-style-type: none"> • 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
RESTCONF YANG パッチのサポート	<p>Cisco IOS XE Amsterdam 17.1.1</p> <ul style="list-style-type: none"> • 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

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	<p data-bbox="841 296 1179 323">Cisco IOS XE Gibraltar 16.11.1</p> <ul data-bbox="873 346 1521 1234" style="list-style-type: none"><li data-bbox="873 346 1521 407">• Cisco ASR 900 シリーズ アグリゲーション サービス ルータ<li data-bbox="873 436 1521 497">• Cisco ASR 920 シリーズ アグリゲーション サービス ルータ<li data-bbox="873 527 1333 554">• Cisco Catalyst 3650 シリーズ スイッチ<li data-bbox="873 583 1333 611">• Cisco Catalyst 3850 シリーズ スイッチ<li data-bbox="873 640 1333 667">• Cisco Catalyst 9200 シリーズ スイッチ<li data-bbox="873 697 1333 724">• Cisco Catalyst 9300 シリーズ スイッチ<li data-bbox="873 753 1333 781">• Cisco Catalyst 9400 シリーズ スイッチ<li data-bbox="873 810 1333 837">• Cisco Catalyst 9500 シリーズ スイッチ<li data-bbox="873 867 1349 894">• Cisco Catalyst IE3200 高耐久性シリーズ<li data-bbox="873 924 1349 951">• Cisco Catalyst IE3300 高耐久性シリーズ<li data-bbox="873 980 1349 1008">• Cisco Catalyst IE3400 高耐久性シリーズ<li data-bbox="873 1037 1521 1064">• Cisco エンベデッド サービス 3300 シリーズ スイッチ<li data-bbox="873 1094 1446 1121">• Cisco IR1101 耐環境性能 サービス統合型ルータ<li data-bbox="873 1150 1458 1178">• Cisco Network Convergence System 4200 シリーズ<li data-bbox="873 1207 1446 1234">• 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	<p>Cisco IOS XE Fuji 16.8.1a</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Gibraltar 16.10.1</p> <ul style="list-style-type: none"> • Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ <p>Cisco IOS XE Gibraltar 16.11.1</p> <ul style="list-style-type: none"> • Cisco Catalyst 9600 シリーズ スイッチ <p>Cisco IOS XE Gibraltar 16.12.1</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Amsterdam 17.1.1</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Amsterdam 17.2.1r</p> <ul style="list-style-type: none"> • 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
gNMI ユーザ名とパスワードによる認証	<p data-bbox="839 296 1179 325">Cisco IOS XE Gibraltar 16.12.1</p> <ul data-bbox="873 346 1516 877" style="list-style-type: none"><li data-bbox="873 346 1328 375">• Cisco Catalyst 9200 シリーズ スイッチ<li data-bbox="873 401 1328 430">• Cisco Catalyst 9300 シリーズ スイッチ<li data-bbox="873 455 1328 485">• Cisco Catalyst 9400 シリーズ スイッチ<li data-bbox="873 510 1328 539">• Cisco Catalyst 9500 シリーズ スイッチ<li data-bbox="873 564 1328 594">• Cisco Catalyst 9600 シリーズ スイッチ<li data-bbox="873 619 1349 648">• Cisco Catalyst IE3200 高耐久性シリーズ<li data-bbox="873 674 1349 703">• Cisco Catalyst IE3200 高耐久性シリーズ<li data-bbox="873 728 1349 758">• Cisco Catalyst IE3300 高耐久性シリーズ<li data-bbox="873 783 1349 812">• Cisco Catalyst IE3400 高耐久性シリーズ<li data-bbox="873 837 1516 867">• 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	<p data-bbox="839 926 1195 955">Cisco IOS XE Amsterdam 17.3.1</p> <ul data-bbox="873 976 1328 1228" style="list-style-type: none"><li data-bbox="873 976 1328 1005">• Cisco Catalyst 9200 シリーズ スイッチ<li data-bbox="873 1031 1328 1060">• Cisco Catalyst 9300 シリーズ スイッチ<li data-bbox="873 1085 1328 1115">• Cisco Catalyst 9400 シリーズ スイッチ<li data-bbox="873 1140 1328 1169">• Cisco Catalyst 9500 シリーズ スイッチ<li data-bbox="873 1194 1328 1224">• 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	<p data-bbox="800 296 1073 323">Cisco IOS XE Fuji 16.8.1</p> <ul data-bbox="837 344 1474 814" style="list-style-type: none"> <li data-bbox="837 344 1354 371">• Cisco 1000 シリーズ サービス統合型ルータ <li data-bbox="837 401 1354 428">• Cisco 4000 シリーズ サービス統合型ルータ <li data-bbox="837 457 1474 520">• Cisco ASR 1000 シリーズ アグリゲーション サービスルータ <li data-bbox="837 550 1463 613">• Cisco ASR 900 シリーズ アグリゲーション サービスルータ <li data-bbox="837 642 1463 705">• Cisco ASR 920 シリーズ アグリゲーション サービスルータ <li data-bbox="837 735 1354 762">• Cisco Cloud Services Router 1000V シリーズ <li data-bbox="837 791 1305 819">• Cisco Network Convergence System 4200 <p data-bbox="800 848 1089 875">Cisco IOS XE Fuji 16.8.1a</p> <ul data-bbox="837 896 1292 1157" style="list-style-type: none"> <li data-bbox="837 896 1292 924">• Cisco Catalyst 3650 シリーズ スイッチ <li data-bbox="837 953 1292 980">• Cisco Catalyst 3850 シリーズ スイッチ <li data-bbox="837 1010 1292 1037">• Cisco Catalyst 9300 シリーズ スイッチ <li data-bbox="837 1066 1292 1094">• Cisco Catalyst 9400 シリーズ スイッチ <li data-bbox="837 1123 1292 1150">• Cisco Catalyst 9500 シリーズ スイッチ

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

機能	リリースとプラットフォーム
	<p>Cisco IOS XE Everest 16.6.1</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Everest 16.6.2</p> <ul style="list-style-type: none"> • Cisco Catalyst 9400 シリーズ スイッチ <p>Cisco IOS XE Fuji 16.7.1</p> <ul style="list-style-type: none"> • 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) <p>Cisco IOS XE Fuji 16.8.1</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Fuji 16.8.1a</p> <ul style="list-style-type: none"> • Cisco Catalyst 9500 ハイ パフォーマンス シリーズ スイッチ <p>Cisco IOS XE Fuji 16.9.1</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Fuji 16.9.2</p> <ul style="list-style-type: none"> • Cisco Catalyst 9200 および 9200L シリーズ スイッチ • Cisco Catalyst 9300L SKU <p>Cisco IOS XE Gibraltar 16.10.1</p>

機能	リリースとプラットフォーム
	<ul style="list-style-type: none">• 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 <ul style="list-style-type: none">• Cisco Catalyst 9600 シリーズ スイッチ Cisco IOS XE Amsterdam 17.3.1 <ul style="list-style-type: none">• 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

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 デイヤルアウト	<p>Cisco IOS XE Gibraltar 16.10.1</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Gibraltar 16.11.1</p> <ul style="list-style-type: none"> • Cisco Catalyst 9600 シリーズ スイッチ <p>Cisco IOS XE Amsterdam 17.3.1</p> <ul style="list-style-type: none"> • 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

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 デイヤルイン	<p data-bbox="841 296 1179 323">Cisco IOS XE Gibraltar 16.12.1</p> <ul data-bbox="873 344 1503 688" style="list-style-type: none"> <li data-bbox="873 344 1503 371">• Cisco Catalyst 9200 および 9200L シリーズ スイッチ <li data-bbox="873 401 1503 428">• Cisco Catalyst 9300 および 9300L シリーズ スイッチ <li data-bbox="873 457 1333 485">• Cisco Catalyst 9400 シリーズ スイッチ <li data-bbox="873 514 1503 575">• Cisco Catalyst 9500 および 9500 ハイ パフォーマンス シリーズ スイッチ <li data-bbox="873 604 1333 632">• Cisco Catalyst 9600 シリーズ スイッチ <li data-bbox="873 661 1471 688">• Cisco cBR-8 コンバージド ブロードバンド ルータ <p data-bbox="841 726 1195 753">Cisco IOS XE Amsterdam 17.1.1</p> <ul data-bbox="873 774 1503 1047" style="list-style-type: none"> <li data-bbox="873 774 1503 835">• Cisco ASR 900 シリーズ アグリゲーション サービス ルータ <li data-bbox="873 865 1503 926">• Cisco ASR 920 シリーズ アグリゲーション サービス ルータ <li data-bbox="873 955 1446 982">• Cisco Network Convergence System 520 シリーズ <li data-bbox="873 1012 1455 1039">• Cisco Network Convergence System 4200 シリーズ <p data-bbox="841 1077 1195 1104">Cisco IOS XE Amsterdam 17.2.1</p> <ul data-bbox="873 1125 1516 1186" style="list-style-type: none"> <li data-bbox="873 1125 1516 1186">• Cisco ASR 1000 シリーズ アグリゲーション サービス ルータ <p data-bbox="841 1224 1195 1251">Cisco IOS XE Amsterdam 17.3.1</p> <ul data-bbox="873 1272 1516 1333" style="list-style-type: none"> <li data-bbox="873 1272 1516 1333">• 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 ダイアルアウト用の TLS	<p>Cisco IOS XE Amsterdam 17.1.1</p> <ul style="list-style-type: none"> • 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	<p>Cisco IOS XE Amsterdam 17.2.1</p> <ul style="list-style-type: none"> • 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	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	<p>Cisco IOS XE Gibraltar 16.11.1</p> <ul style="list-style-type: none"> • 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 <p>Cisco IOS XE Gibraltar 16.11.1a</p> <ul style="list-style-type: none"> • 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

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	<p>Cisco IOS XE Everest 16.5.1a</p> <ul style="list-style-type: none"> • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ <p>Cisco IOS XE Everest 16.5.1b</p> <ul style="list-style-type: none"> • Cisco 4000 シリーズ サービス統合型ルータ <p>Cisco IOS XE Everest 16.6.1</p> <ul style="list-style-type: none"> • Cisco Catalyst 3650 シリーズ スイッチ • Cisco Catalyst 3850 シリーズ スイッチ <p>Cisco IOS XE Everest 16.6.2</p> <ul style="list-style-type: none"> • Cisco Catalyst 9400 シリーズ スイッチ <p>Cisco IOS XE Fuji 16.7.1</p> <ul style="list-style-type: none"> • Cisco ASR 1000 アグリゲーション サービス ルータ <p>Cisco IOS XE Fuji 16.8.1a</p> <ul style="list-style-type: none"> • 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	<p>Cisco IOS XE Gibraltar 16.12.1</p> <ul style="list-style-type: none"> • Cisco Catalyst 9300 シリーズ スイッチ <p>Cisco IOS XE Amsterdam 17.1.1</p> <ul style="list-style-type: none"> • Cisco Catalyst 9400 シリーズ スイッチ <p>Cisco IOS XE Amsterdam 17.2.1</p> <ul style="list-style-type: none"> • 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

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	<p>Cisco IOS XE Gibraltar 16.12.1</p> <ul style="list-style-type: none"> • Cisco Catalyst 9300 シリーズ スイッチ <p>Cisco IOS XE Amsterdam 17.1.1</p> <ul style="list-style-type: none"> • Cisco Catalyst 9400 シリーズ スイッチ <p>(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。</p>
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	<p>Cisco IOS XE Gibraltar 16.12.1</p> <ul style="list-style-type: none"> • Cisco Catalyst 9300 シリーズ スイッチ <p>Cisco IOS XE Amsterdam 17.1.1</p> <ul style="list-style-type: none"> • Cisco Catalyst 9400 シリーズ スイッチ <p>(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。</p>
アプリケーションホスティング： ThousandEyes の統合	<p>Cisco IOS XE Amsterdam 17.3.3</p> <ul style="list-style-type: none"> • Cisco Catalyst 9300 および 9300L シリーズ スイッチ
ネイティブ Docker コンテナ：ア プリケーションの自動再起動	<p>Cisco IOS XE Amsterdam 17.2.1</p> <ul style="list-style-type: none"> • Cisco Catalyst 9300 シリーズ スイッチ
OpenFlow	
OpenFlow	<p>Cisco IOS XE Fuji 16.9.1</p> <ul style="list-style-type: none"> • 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	<p>Cisco IOS XE Gibraltar 16.12.1</p> <ul style="list-style-type: none"> • Catalyst 9300 シリーズ スイッチ • Catalyst 9400 シリーズ スイッチ



第 1 部

プロビジョニング

- [ゼロ タッチ プロビジョニング \(35 ページ\)](#)
- [iPXE \(71 ページ\)](#)



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 \(35 ページ\)](#)
- [ゼロ タッチ プロビジョニングについて \(35 ページ\)](#)
- [ゼロ タッチ プロビジョニングの構成例 \(37 ページ\)](#)
- [ゼロ タッチ プロビジョニングの機能情報 \(66 ページ\)](#)

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 "xxxxxxxxxxxxxxxx";
  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 コピーを使用する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 コピーを使用している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
 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 Lsmpil8/3, 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:.
OK 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
Tel1/0/37        unassigned      YES unset   down      down
Tel1/0/38        unassigned      YES unset   down      down
Tel1/0/39        unassigned      YES unset   down      down
Tel1/0/40        unassigned      YES unset   down      down
Tel1/0/41        unassigned      YES unset   down      down
Tel1/0/42        unassigned      YES unset   down      down
Tel1/0/43        unassigned      YES unset   down      down
Tel1/0/44        unassigned      YES unset   down      down
Tel1/0/45        unassigned      YES unset   down      down
Tel1/0/46        unassigned      YES unset   down      down
Tel1/0/47        unassigned      YES unset   down      down
Tel1/0/48        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
Tel1/1/1         unassigned      YES unset   down      down
Tel1/1/2         unassigned      YES unset   down      down
Tel1/1/3         unassigned      YES unset   down      down
Tel1/1/4         unassigned      YES unset   down      down
Tel1/1/5         unassigned      YES unset   down      down
Tel1/1/6         unassigned      YES unset   down      down
Tel1/1/7         unassigned      YES unset   down      down
Tel1/1/8         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
```



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

```
Switch/Stack Mac Address : ecl.d.8b0a.6800 - Local Mac Address
Mac persistency wait time: Indefinite
```

```

Switch#   Role           Priority   Current
-----   ---           -
*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           Type           Technology-package
Current                     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:.
OK 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
Tel1/0/37         unassigned      YES unset   down      down
Tel1/0/38         unassigned      YES unset   down      down
Tel1/0/39         unassigned      YES unset   down      down
Tel1/0/40         unassigned      YES unset   down      down
Tel1/0/41         unassigned      YES unset   down      down
Tel1/0/42         unassigned      YES unset   down      down
Tel1/0/43         unassigned      YES unset   down      down
Tel1/0/44         unassigned      YES unset   down      down
Tel1/0/45         unassigned      YES unset   down      down
Tel1/0/46         unassigned      YES unset   down      down
Tel1/0/47         unassigned      YES unset   down      down
Tel1/0/48         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
Tel1/1/1          unassigned      YES unset   down      down
Tel1/1/2          unassigned      YES unset   down      down
Tel1/1/3          unassigned      YES unset   down      down
Tel1/1/4          unassigned      YES unset   down      down
Tel1/1/5          unassigned      YES unset   down      down
Tel1/1/6          unassigned      YES unset   down      down
Tel1/1/7          unassigned      YES unset   down      down
Tel1/1/8          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

```

```

Switch#      Role      Priority    Current
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      Type      Technology-package
Current                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
Tel/0/37          unassigned      YES unset  down      down
Tel/0/38          unassigned      YES unset  down      down
Tel/0/39          unassigned      YES unset  down      down
Tel/0/40          unassigned      YES unset  down      down
Tel/0/41          unassigned      YES unset  down      down
Tel/0/42          unassigned      YES unset  down      down
Tel/0/43          unassigned      YES unset  down      down
Tel/0/44          unassigned      YES unset  down      down
Tel/0/45          unassigned      YES unset  down      down
Tel/0/46          unassigned      YES unset  down      down
Tel/0/47          unassigned      YES unset  down      down
Tel/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
Tel/1/1           unassigned      YES unset  down      down
Tel/1/2           unassigned      YES unset  down      down
Tel/1/3           unassigned      YES unset  down      down
Tel/1/4           unassigned      YES unset  down      down
Tel/1/5           unassigned      YES unset  down      down
Tel/1/6           unassigned      YES unset  down      down
Tel/1/7           unassigned      YES unset  down      down
Tel/1/8           unassigned      YES unset  down      down
Fol/1/1          unassigned      YES unset  down      down
Fol/1/2          unassigned      YES unset  down      down
TwentyFiveGigE1/1/1 unassigned      YES unset  down      down
TwentyFiveGigE1/1/2 unassigned      YES unset  down      down
Apl/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  ecd.8b0a.6800 V01       17.02.01

Switch/Stack Mac Address : ecd.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                                           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
Apl/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 トレー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
		<p>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。</p> <p>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。</p> <ul style="list-style-type: none"> • 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 <p>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。</p> <ul style="list-style-type: none"> • 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。 <p>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。</p> <ul style="list-style-type: none"> • 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) <p>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。</p> <ul style="list-style-type: none"> • 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
		<p>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。</p> <ul style="list-style-type: none"> • Cisco Catalyst 9200 シリーズ スイッチ <p>(注) この機能は C9200L SKUではサポートされていません。</p> <ul style="list-style-type: none"> • 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 <p>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。</p> <ul style="list-style-type: none"> • 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) <p>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。</p> <ul style="list-style-type: none"> • 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
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	<p>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。</p> <p>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。</p> <ul style="list-style-type: none"> • 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 <p>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。</p>
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	<p>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。</p> <ul style="list-style-type: none"> • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ <p>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。</p> <ul style="list-style-type: none"> • 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



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 \(71 ページ\)](#)
- [iPXE の設定方法 \(81 ページ\)](#)
- [iPXE の設定例 \(83 ページ\)](#)
- [iPXE のトラブルシューティングのヒント \(85 ページ\)](#)
- [iPXE に関する追加情報 \(87 ページ\)](#)
- [iPXE の機能情報 \(87 ページ\)](#)

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。

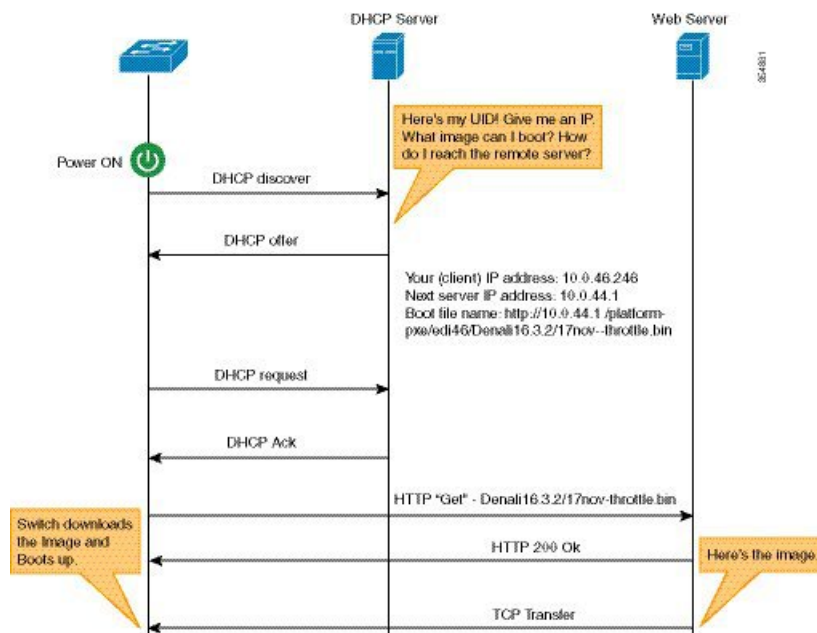


(注) このマニュアルでは、手動ブートという用語も使われています。手動ブートは、ROMmon のリロードを行うかどうかを決定するフラグです。デバイスが ROMmon モードの場合は、手動で **boot** コマンドを実行する必要があります。

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スペ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、ユーザーによって **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ます、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

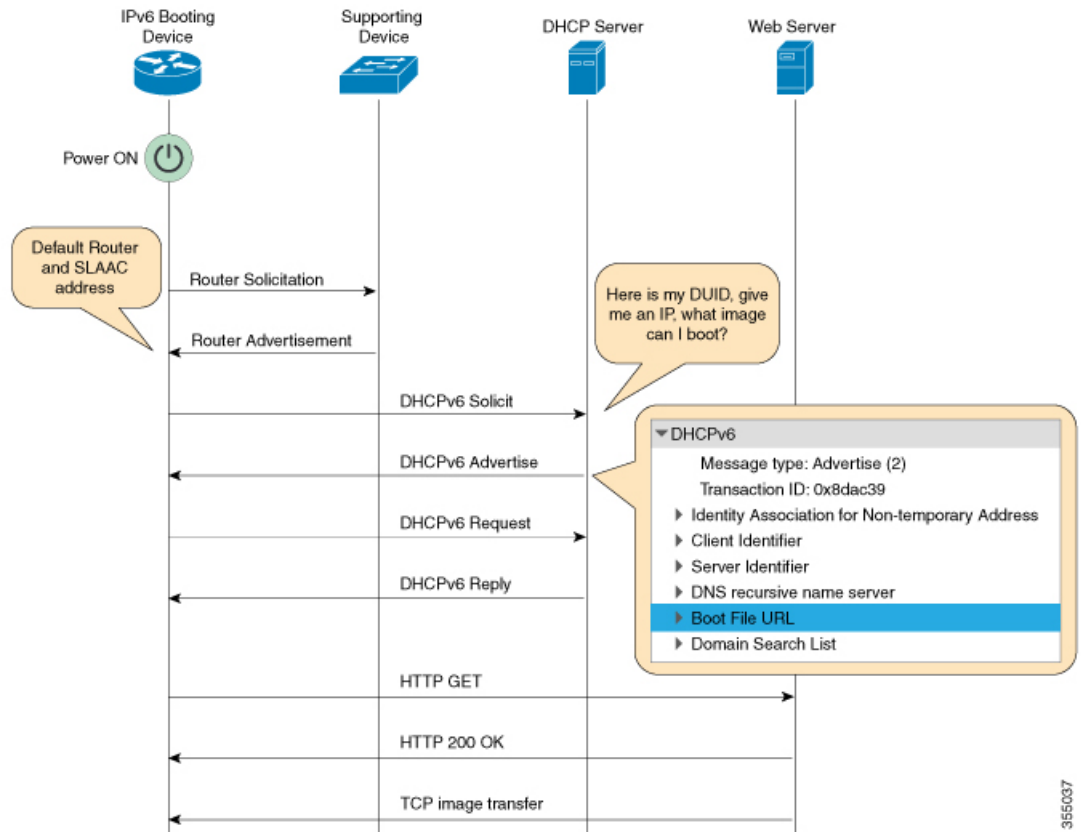
Switch#	Role	Mac Address	Priority	H/W Version	Current State
1	Member	046c.9d1e.1a00	1		Ready
2	Standby	046c.9d01.7d80	1		Ready
*3	Active	f8b7.e24e.9a00	1	P2B	Ready

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 アドバタイズメント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 のブレイクを有効にします。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 のみを変更できます。

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	<ul style="list-style-type: none"> • boot ipxe forever [<i>switch number</i>] • boot ipxe timeout <i>seconds</i> [<i>switch number</i>] 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 { <i>switch switch-number</i> all } { flash: ftp: http: usbflash0 tftp: }	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。
	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	

	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	<ul style="list-style-type: none"> • 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` + のシャシー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 進形式でのシャシーシリアル番号の ASCII コードが続きます。このサンプルのスイッチのシャシーシリアル番号は、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:43:31:38:33:31:58:31:58:31:58:31:41:53;
  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_NUMROMmon 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、別のキーストロッ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	『 <i>Dynamic Host Configuration Protocol for IPv6 (DHCPv6)</i> 』
RFC 3986	『 <i>Uniform Resource Identifier (URI): Generic Syntax</i> 』

シスコのテクニカル サポート

説明	リンク
<p>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。</p> <p>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。</p> <p>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。</p>	<p>http://www.cisco.com/support</p>

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。 <ul style="list-style-type: none"> • 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。 <ul style="list-style-type: none"> • 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。 <ul style="list-style-type: none"> • 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	<p>IPXE は IPv6 プロトコルをサポートしています。</p> <p>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。</p> <ul style="list-style-type: none">• 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

- [ゲスト シェル \(93 ページ\)](#)
- [Python API \(125 ページ\)](#)
- [CLI Python モジュール \(131 ページ\)](#)
- [EEM Python モジュール \(139 ページ\)](#)



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（93 ページ）](#)
- [ゲスト シェルについて（93 ページ）](#)
- [ゲスト シェルを有効にする方法（105 ページ）](#)
- [ゲスト シェルの設定例（114 ページ）](#)
- [ゲスト シェルに関するその他の参考資料（119 ページ）](#)
- [ゲスト シェルの機能情報（120 ページ）](#)

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。

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、ユーザーは、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	<ul style="list-style-type: none"> 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。 <p>(注) CentOS は Python 3.6 のみをサポートします。</p>
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	<p>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。</p> <p>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。</p> <p>(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。</p>
事前にインストールされたカスタムの Python ライブラリ	<ul style="list-style-type: none"> • 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	対応
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に、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 のネットワーク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せん。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

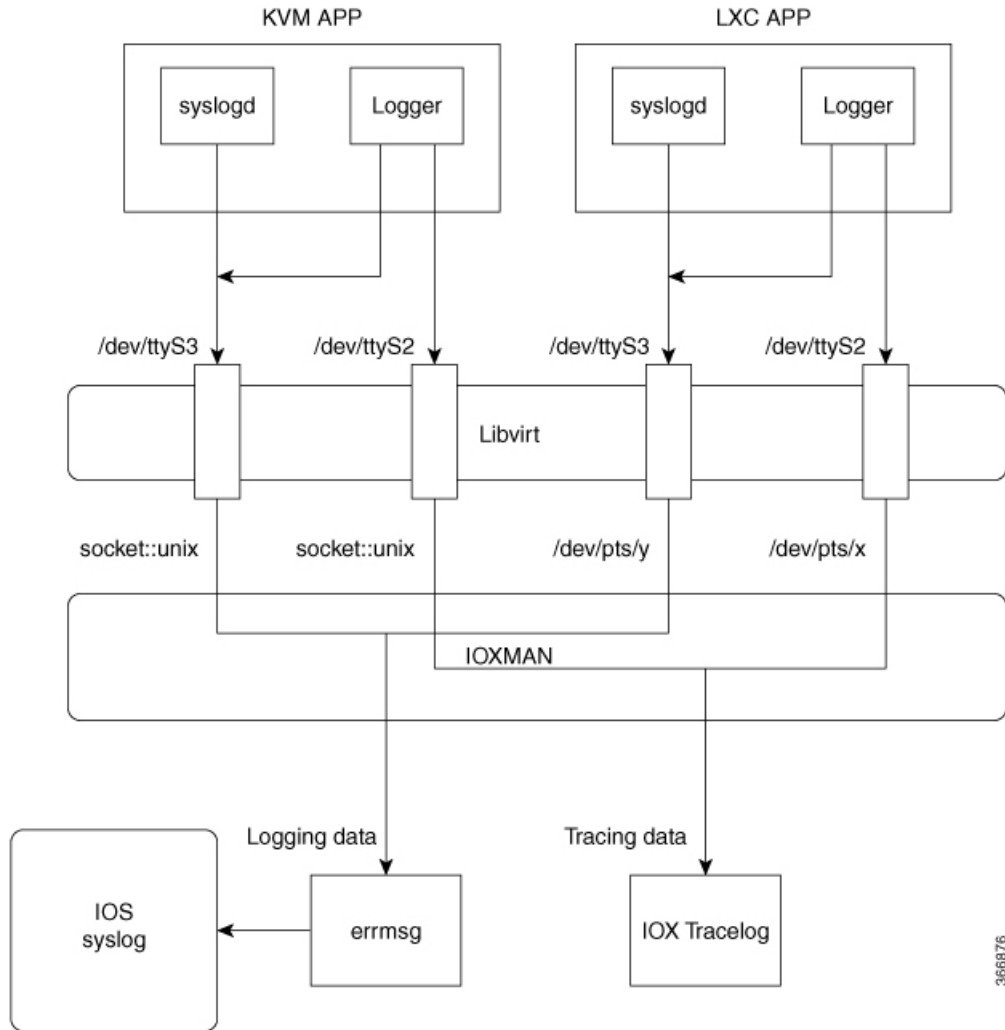
IOx のトレースとロギングの機能を使用すると、ホスト デバイスでゲストアプリケーションを個別に実行できます。これにより、ホストへのデータのロギングとトレースをレポートするのに役立ちます。トレースデータは IOx トレースログに保存され、ロギングデータはホストデバイスの Cisco IOS syslog に保存されます。

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。

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
Libvirt 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) <ul style="list-style-type: none"> • guestshell enable コマンドは、ネットワークに管理 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（106 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

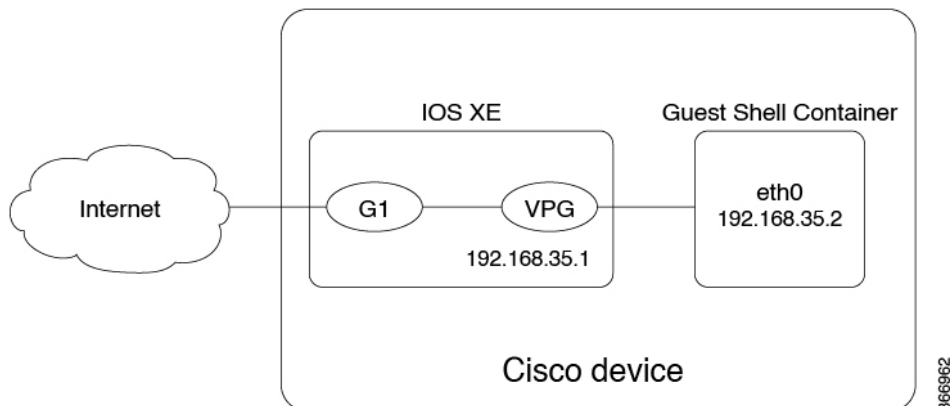
Device(config-if)# interface VirtualPortGroup0
Device(config-if)# ip address 192.168.35.1 255.255.255.0
Device(config-if)# ip nat inside
Device(config-if)# exit

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せん。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 <i>interface-number</i> 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 <i>name</i> 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 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。

	コマンドまたはアクション	目的
ステップ 8	vlan <i>vlan-ID</i> guest-interface <i>guest-interface-number</i> 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 <i>ip-address netmask netmask</i> 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 <i>ip-address guest-interface network-interface</i> 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# <i>ip-address</i> 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。

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ます。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
```

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

```

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ク設定

ゲストシェルのネットワークでは、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-host 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
	<p>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。</p> <p>http://www.cisco.com/go/mibs</p>

シスコのテクニカル サポート

説明	リンク
<p>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。</p> <p>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。</p> <p>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。</p>	<p>http://www.cisco.com/support</p>

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
<p>ゲストシェル</p>	<p>Cisco IOS XE Everest 16.5.1a Cisco IOS XE Everest 16.5.1b</p>	<p>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。</p> <p>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。</p> <ul style="list-style-type: none"> • 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 <p>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。</p> <ul style="list-style-type: none"> • Cisco 4000 シリーズサービスタ統合型ルータ
	<p>Cisco IOS XE Everest 16.6.2</p>	<p>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。</p>

機能名	リリース	機能情報
	Cisco IOS XE Fuji 16.7.1	<p>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。</p> <ul style="list-style-type: none"> • 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 <p>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。</p>
	Cisco IOS XE Fuji 16.8.1	<p>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。</p>
	Cisco IOS XE Fuji 16.9.1	<p>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。</p>
	Cisco IOS XE Gibraltar 16.11.1b	<p>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。</p> <ul style="list-style-type: none"> • 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
		<p>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。</p> <ul style="list-style-type: none"> • Cisco Catalyst 9200 シリーズ スイッチ <p>(注) この機能は C9200L SKU ではサポートされていません。</p> <ul style="list-style-type: none"> • Cisco Catalyst 9300L SKU • Cisco Catalyst 9600 シリーズ スイッチ
	Cisco IOS XE Amsterdam 17.3.1	<p>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。</p> <ul style="list-style-type: none"> • 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
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。



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 \(125 ページ\)](#)

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(cli)
```

```
Help on function cli in module cli:
```

```
cli(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.
```

```
cli("show version")
```

```
cli("show version ; show ip interface brief")
```

```
cli("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.cli('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.cli('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.cli(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.cli(command)** 関数の出力例を示します。

```

>>> cli
>>> cli.cli('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.cli('show clock')
*18:13:35.313 UTC Mon Mar 13 2017
>>> output=cli.cli('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 \(131 ページ\)](#)
- [CLI Python モジュールに関するその他の参考資料 \(135 ページ\)](#)
- [CLI Python モジュールの機能情報 \(136 ページ\)](#)

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
:
:
:
Tel1/1/4                  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

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	<ul style="list-style-type: none"> • 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	<p>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。</p> <p>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。</p> <p>(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。</p> <p>(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。</p>

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
<p>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。</p> <p>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。</p> <p>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。</p>	<p>http://www.cisco.com/support</p>

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	<p>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。</p> <p>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。</p> <ul style="list-style-type: none"> • 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 <p>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。</p> <ul style="list-style-type: none"> • 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	<p>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。</p> <ul style="list-style-type: none"> • 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 \(139 ページ\)](#)
- [EEM Python モジュールについて \(139 ページ\)](#)
- [EEM Python ポリシーの設定方法 \(142 ページ\)](#)
- [EEM Python モジュールに関するその他の参考資料 \(148 ページ\)](#)
- [EEM Python モジュールの機能情報 \(148 ページ\)](#)

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せん。
- 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ンネルハンドラをリリースします。
- eem_cli_exec : 指定されたチャン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、チャン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 をチャンネルハンドラに接続します。チャン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 のチャン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 のチャンネルハンドラから、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 のチャン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 チャン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 チャン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	<code>enable</code> 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	<code>configure terminal</code> 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： <pre>Device(config)# event manager directory user policy flash:/user_library</pre>	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、 <code>eem_script.py</code> ポリシーが <code>flash:/user_library</code> フォルダまたはパスで使用できます。
ステップ 4	event manager policy policy-filename 例： <pre>Device(config)# event manager policy eem_script.py</pre>	EEM ポリシーを EEM に登録します。 <ul style="list-style-type: none"> • 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 <code>.py</code> 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： <pre>Device(config)# exit</pre>	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： <pre>Device# show event manager policy registered</pre>	保留 EEM ポリシーを表示します。
ステップ 7	show event manager history events 例： <pre>Device# show event manager history events</pre>	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
Tel1/1/1 unassigned YES unset down down
Tel1/1/2 unassigned YES unset down down
Tel1/1/3 unassigned YES unset down down
Tel1/1/4 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 <i>applet-name</i> 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 [<i>tag event-tag</i>] syslog pattern <i>regular-expression</i> 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。 <ul style="list-style-type: none">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
<p>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。</p> <p>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。</p> <p>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。</p>	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 トレー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。 <ul style="list-style-type: none"> • 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。 <ul style="list-style-type: none"> • 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

- [NETCONF プロトコル \(153 ページ\)](#)
- [RESTCONF プロトコル \(181 ページ\)](#)
- [NETCONF および RESTCONF のサービスレベル ACL \(203 ページ\)](#)
- [gNMI プロトコル \(211 ページ\)](#)
- [gRPC ネットワーク操作インターフェイス \(235 ページ\)](#)
- [モデルベースの AAA \(247 ページ\)](#)
- [モデル駆動型テレメトリ \(257 ページ\)](#)
- [In-Service Model Update \(303 ページ\)](#)



第 8 章

NETCONF プロトコル

- [NETCONF プロトコルの概要 \(153 ページ\)](#)
- [NETCONF プロトコルの設定方法 \(161 ページ\)](#)
- [NETCONF プロトコルのコンフィギュレーションの確認 \(167 ページ\)](#)
- [NETCONF プロトコルの関連資料 \(169 ページ\)](#)
- [NETCONF プロトコルの機能情報 \(170 ページ\)](#)

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 は非常に独自のであり、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 はシンプルなり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>
```

```

        </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) リバースト、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 ~ 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、アカウントिंग (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ー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。

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 を持つユーザー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。 <ul style="list-style-type: none"> • 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、アカウントिंग (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。 <ul style="list-style-type: none"> • 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。

	コマンドまたはアクション	目的
		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： <pre>Device(config)# aaa authorization exec default local</pre>	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。 <ul style="list-style-type: none"> 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： <pre>Device(config)# end</pre>	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： <pre>Device> enable</pre>	特権 EXEC モードを有効にします。 <ul style="list-style-type: none"> 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： <pre>Device# configure terminal</pre>	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	netconf-yang 例： Device (config)# netconf-yang	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。 (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 <i>username</i> 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 <code>.ssh/id_rsa.pub</code> ファイルから）公開キー値を取得できます。

	コマンドまたはアクション	目的
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。

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 statistics**
5. **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 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5 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

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	<i>YANG : Network Configuration Protocol (NETCONF)</i> 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 (<i>NETCONF</i>)
RFC 6536	ネットワーク設定プロトコル (<i>NETCONF</i>) アクセス制御モデル
RFC 8040	<i>RESTCONF</i> プロトコル

シスコのテクニカル サポート

説明	リンク
<p>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。</p> <p>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。</p> <p>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。</p>	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	<p>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。</p> <p>次のコマンドが導入されました： netconf-yang</p> <p>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。</p> <ul style="list-style-type: none"> • 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	<p>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。</p> <ul style="list-style-type: none"> • 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	<p>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。</p> <ul style="list-style-type: none"> • 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
		<p>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。</p> <ul style="list-style-type: none"> • 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	<p>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。</p> <ul style="list-style-type: none"> • 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	<p>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。</p> <ul style="list-style-type: none"> • 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	<p>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。</p>
	Cisco IOS XE Gibraltar 16.12.1	<p>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。</p>

機能名	リリース	機能情報
	Cisco IOS XE Amsterdam 17.3.1	<p>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。</p> <ul style="list-style-type: none">• 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	<p>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す。</p> <ul style="list-style-type: none"> • 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	<p>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。</p>

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	<p>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。</p> <ul style="list-style-type: none"> • 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	<p>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。</p> <p>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。</p> <ul style="list-style-type: none"> • 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 <p>次のコマンドが導入されました： netconf-yang feature candidate-datastore</p>

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	<p>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。</p> <p>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。</p> <ul style="list-style-type: none"> • 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
		<p>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。</p> <ul style="list-style-type: none"> • 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



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 \(181 ページ\)](#)
- [RESTCONF プロトコルの制約事項 \(181 ページ\)](#)
- [RESTCONF プロトコルに関する情報 \(182 ページ\)](#)
- [RESTCONF プロトコルの設定方法 \(189 ページ\)](#)
- [RESTCONF プロトコルの設定例 \(194 ページ\)](#)
- [RESTCONF プロトコルの関連資料 \(197 ページ\)](#)
- [RESTCONF プロトコルの機能情報 \(198 ページ\)](#)

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 のルートを決めます。
- 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`

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:DMIdmil! --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せん。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:DMIdml! --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:DMIdml! --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:DMIdmil! --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、およびアカウントिंग (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)。
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。

	コマンドまたはアクション	目的
ステップ 4	aaa group server radius <i>server-name</i> 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。 • <i>server-name</i> 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 <i>ip-address key key-name</i> 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 <i>vrf-name</i> 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 <i>group-name</i> 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 <i>list-name</i> 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 <i>group-name</i> 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： Device(config)# aaa session-id common	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。
ステップ 12	line console <i>number</i> 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 <i>authentication-list</i> 例：	ログインに対する AAA 認証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-line)# login authentication NOAUTH	
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： Device(config)# ip http secure-server	セキュア HTTP (HTTPS) サーバをイネーブルにします。
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 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
<p>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。</p> <p>お使いの製品のセキュリティ情報や技術情報入手するために、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。</p> <p>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。</p>	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	<p>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。</p> <p>この機能が次のプラットフォームで追加されました。</p> <ul style="list-style-type: none"> • 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 <p>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</p>
	Cisco IOS XE Fuji 16.8.1a	<p>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。</p> <ul style="list-style-type: none"> • 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	<p>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。</p> <ul style="list-style-type: none"> • 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
		<p>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。</p> <ul style="list-style-type: none"> • 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	<p>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。</p> <ul style="list-style-type: none"> • 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
RESTCONF YANG パッチのサポート	Cisco IOS XE Amsterdam 17.1.1	<p>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。</p> <p>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。</p> <ul style="list-style-type: none"> • 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](#) (203 ページ)
- [NETCONF および RESTCONF のサービスレベル ACL の設定方法](#) (204 ページ)
- [NETCONF および RESTCONF のサービスレベル ACL の設定例](#) (207 ページ)
- [NETCONF および RESTCONF のサービスレベル ACL に関するその他の資料](#) (207 ページ)
- [NETCONF および RESTCONF のサービスレベル ACL の機能情報](#) (208 ページ)

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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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 { <i>host-address</i> <i>host-name</i> any } [<i>wildcard</i>] 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 {{ <i>ipv4</i> <i>ipv6</i> } access-list name <i>access-list-name</i> } port <i>port-number</i> } 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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
<p>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。</p> <p>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。</p> <p>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。</p>	<p>http://www.cisco.com/support</p>

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	<p>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。</p> <p>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</p> <ul style="list-style-type: none"> • 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 \(211 ページ\)](#)
- [gNMI プロトコルの概要 \(212 ページ\)](#)
- [gNMI プロトコルを有効にする方法 \(223 ページ\)](#)
- [gNMI プロトコルの設定例 \(229 ページ\)](#)
- [gNMI プロトコルの関連資料 \(230 ページ\)](#)
- [gNMI プロトコルの機能情報 \(231 ページ\)](#)

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 ASCII 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: "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
<pre> +++++++ 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\"}" } } </pre>	<pre> +++++++ Received set response: ++++++ response { path { elem { name: "interfaces" } elem { name: "interface" key { key: "name" value: "Loopback111" } } elem { name: "config" } } op: UPDATE } timestamp: 1521699342123890045 </pre>

表 17: リーフ値での SetRequest の例

SetRequest	SetResponse
<pre> +++++++ 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\"" } } </pre>	<pre> +++++++ 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 </pre>

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、SetRequestRPC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.gNMISubject 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
# Encrypt 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	gNxi 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 <i>port-number</i> 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	gNxI 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.gNMISub(stub=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
<p>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。</p> <p>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。</p> <p>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。</p>	<p>http://www.cisco.com/support</p>

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	<p>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。</p> <p>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。</p> <ul style="list-style-type: none"> • 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	<p>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。</p> <ul style="list-style-type: none"> • 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
		<p>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。</p> <ul style="list-style-type: none"> • 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	<p>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。</p>
gNMI ユーザ名とパスワードによる認証	Cisco IOS XE Gibraltar 16.12.1	<p>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。</p>
gNMI 設定の永続化	Cisco IOS XE Amsterdam 17.3.1	<p>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。</p> <ul style="list-style-type: none"> • 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
gNOI 証明書の管理	Cisco IOS XE Amsterdam 17.3.1	<p>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。</p> <ul style="list-style-type: none"> • 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
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	<p>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。</p> <p>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。</p> <ul style="list-style-type: none"> • 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 \(235 ページ\)](#)
- [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 \(243 ページ\)](#)
- [gRPC ネットワーク操作インターフェイスの機能情報 \(244 ページ\)](#)

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せん。

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/xr/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

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 (`InstallCertificateResponse`) 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、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。

- 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。
- 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 証明書はトラスプールに存在しており、トラス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。

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

シスコのテクニカル サポート

説明	リンク
<p>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。</p> <p>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。</p> <p>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。</p>	<p>http://www.cisco.com/support</p>

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이스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	<p>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。</p> <p>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。</p> <ul style="list-style-type: none">• 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	<p>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。</p> <p>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。</p> <ul style="list-style-type: none"> • 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 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 \(247 ページ\)](#)
- [モデルベースの AAA に関するその他の参考資料 \(253 ページ\)](#)
- [モデルベースの AAA に関する機能情報 \(254 ページ\)](#)

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 NACM 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 \(156 ページ\)](#) 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
<code><module-name>ietf-netconf</module-name></code>	RPC を含むモジュールの名前
<code><rpc-name>edit-config</rpc-name></code>	RPC の名前
<code><access-operations>exec</access-operations></code>	RPC の実行権限
<code><action>deny</action></code>	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
<code><name>hide-enable-passwords</name></code>	固有のルールリスト名
<code>< group > permission </group ></code>	ルールリストが適用されるグループ
<code><name>deny-enable-passwords</name></code>	固有のルール名
<code><path xmlns:ios="http://cisco.com/ns/yang/Cisco-IOS-XE-native">/ios:native/enable</path></code>	許可または拒否されるデータ ノードへのパス
<code><access-operations>*</access-operations></code>	CRUDx の動作タイプ
<code><action>deny</action></code>	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	<i>YANG : Network Configuration Protocol (NETCONF)</i> 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 (<i>NETCONF</i>)
RFC 6536	ネットワーク設定プロトコル (<i>NETCONF</i>) 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
<p>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。</p> <p>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。</p> <p>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。</p>	<p>http://www.cisco.com/support</p>

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	<p>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。</p> <ul style="list-style-type: none"> • 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	<p>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。</p> <ul style="list-style-type: none"> • 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 \(257 ページ\)](#)

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
<code>confd</code>	コンフィギュレーションデーモン
<code>nesd</code>	ネットワーク要素シンクロナイザデーモン
<code>syncfd</code>	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）
<code>ncsshd</code>	NETCONF セキュアシェル（SSH）デーモン
<code>dmiauthd</code>	DMI 認証デーモン。
<code>nginx</code>	NGINX Web サーバ。RESTCONF の Web サーバとして機能します。
<code>ndbmand</code>	NETCONF データベースマネージャ
<code>pubd</code>	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
<code>gnmib</code>	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 brevity):
{
  "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含まれません。たとえば、サブスクリイバでは、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。サブスクリ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

- デバイスと受信者間の 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。

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

プシジョンの1つが試行されます。定期的にとり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、変更時にとり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

トランスポートプロトコル	NETCONF		gRPC		gNMI	
Encodings	XML	非対応	非対応	キー値 Google Protocol Buffers (kvGPB)	JSON_IETF	なし

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。 <ul style="list-style-type: none"> サブスクリプションが多すぎる。 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、サブスクリプションサービス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 ダイア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リ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、サブスクリ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リ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リ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リ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リバ 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、ダイア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、サブスクリ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、サブスクリライブのみが発行でき、そのサブスクリライブ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>
      </mdt-subscriptions>
    </mdt-oper-data>
  </data>
</rpc-reply>
```

```

        </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 ダイアルインサブスクリプション

次に、gNMI ダイア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、先に受信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>
```

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、サブスクリ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

サブスクリプ先の *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
```

- 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。ユニオン演算子は NETCONF トランスポートでのみ機能し、gRPC では機能しません。

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、サブスクリプ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、サブスクリプ対象情報の更新は 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リプ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リプすると、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。

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。

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。

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。次のガイドラインが XPath 式に適用されます。

- 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。
- ユニオン演算子 (|) はサポートされていません。

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ー (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際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、設定された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。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.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 ダイア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 ダイア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）。

	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	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。
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 <i>id</i> 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 <i>path</i> 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。
ステップ 6	update-policy {on-change periodic <i>period</i>} 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。

	コマンドまたはアクション	目的
ステップ 8	receiver ip address <i>ip-address</i> receiver-port protocol protocol profile name 例： Device(config-mdt-sub)# 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-sub)# 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 ダイアラ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、自己終了 `datastore-contents-xml` タグを持つこととなります。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-IDdetail** 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": ""
        }
      ]
    }
  ]
}

```

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
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 <i>draft-ietf-netconf-subscribed-notifications-03</i>	https://tools.ietf.org/id/draft-ietf-netconf-subscribed-notifications-03.txt
イベント通知の <i>NETCONF</i> サポート	draft-ietf-netconf-netconf-event-notifications-01
<i>RFC 5277</i>	NETCONF イベント通知
<i>RFC 6241</i>	ネットワーク設定プロトコル (NETCONF)

標準/RFC	タイトル
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リプション	draft-ietf-netconf-yang-push-04
YANG データストア プッシュ更新のサブスクリプション 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
<p>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。</p> <p>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。</p> <p>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。</p>	<p>http://www.cisco.com/support</p>

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 28: 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	<p>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リバに、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。</p> <ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Cisco Catalyst 9400 シリーズ スイッチ
	Cisco IOS XE Fuji 16.7.1	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<p>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。</p> <p>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。</p> <ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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
		<p>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。</p> <p>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。</p> <ul style="list-style-type: none"> • 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
		<ul style="list-style-type: none"> • Cisco Network Convergence System 4200 シリーズ
	Cisco IOS XE Gibraltar 16.11.1	<ul style="list-style-type: none"> • 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	<p>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。</p> <ul style="list-style-type: none"> • 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	<p>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。</p> <p>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。</p> <ul style="list-style-type: none"> • 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
		<p>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。</p> <ul style="list-style-type: none"> • 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 <ul style="list-style-type: none">• 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](#) (303 ページ)
- [In-Service Model Update について](#) (303 ページ)
- [In-Service Model Update の管理方法](#) (306 ページ)
- [In-Service Model Update の設定例](#) (308 ページ)
- [In-Service Model Update の機能情報](#) (312 ページ)

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ま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

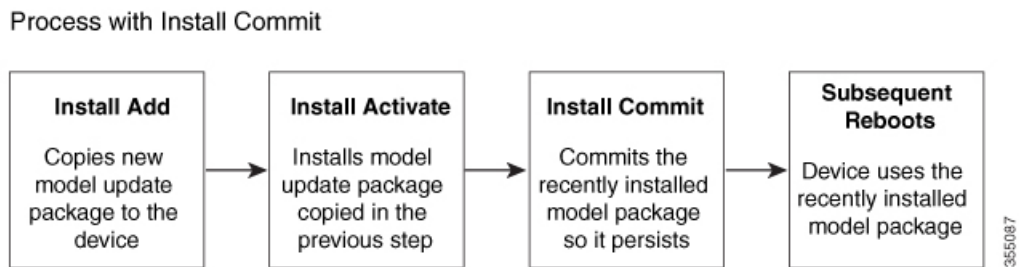
デバイスに In-Service Model Update パッケージをインストールするには、**install add**、**install activate**、および **install commit** コマンドを特権 EXEC モードで使用します。

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 5: 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 29: 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	<p>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。</p> <p>(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。</p>

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	<p>enable</p> <p>例 :</p> <pre>Device> enable</pre>	<p>特権 EXEC モードを有効にします。</p> <ul style="list-style-type: none"> パスワードを入力します (要求された場合)。
ステップ 2	<p>install add file tftp: filename</p> <p>例 :</p> <pre>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</pre>	<p>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。</p> <ul style="list-style-type: none"> 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	<p>install activate file bootflash: filename</p> <p>例 :</p> <pre>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</pre>	<p>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。</p> <ul style="list-style-type: none"> 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	<p>install commit</p> <p>例 :</p> <pre>Device# install commit</pre>	<p>リロードが繰り返されても持続する変更を行います。</p> <ul style="list-style-type: none"> NETCONF プロセスは再開されません。
ステップ 5	<p>install deactivate file bootflash: filename</p> <p>例 :</p> <pre>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</pre>	<p>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。</p>
ステップ 6	<p>install commit</p> <p>例 :</p> <pre>Device# install commit</pre>	<p>リロードが繰り返されても持続する変更を行います。</p> <ul style="list-style-type: none"> NETCONF プロセスは再開されません。
ステップ 7	<p>install rollback to {base committed id commit-ID}</p> <p>例 :</p> <pre>Device# install rollback to base</pre>	<p>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。</p> <ul style="list-style-type: none"> <i>commit-id</i> 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
		<ul style="list-style-type: none"> 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 : <pre>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</pre>	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。 <ul style="list-style-type: none"> 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 : <pre>Device# show install summary</pre>	アクティブパッケージに関する情報を表示します。 <ul style="list-style-type: none"> 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: ned: 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: ned:
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 30 : 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	<p>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。</p> <p>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。</p> <ul style="list-style-type: none"> • Cisco Catalyst 9300 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ <p>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。</p> <ul style="list-style-type: none"> • 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) <p>コマンド install (Programmability)、show install (Programmability) が導入または更新されました。</p>
	Cisco IOS XE Everest 16.6.1	<p>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。</p> <ul style="list-style-type: none"> • 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	<p>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。</p> <ul style="list-style-type: none"> • 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

- [アプリケーションホスティング \(317ページ\)](#)



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 \(317 ページ\)](#)
- [アプリケーションホスティングに関する情報 \(318 ページ\)](#)
- [アプリケーションホスティングの設定方法 \(329 ページ\)](#)
- [アプリケーションホスティング設定の確認 \(347 ページ\)](#)
- [アプリケーションホスティングの設定例 \(350 ページ\)](#)
- [その他の参考資料 \(358 ページ\)](#)
- [アプリケーションホスティングに関する機能情報 \(359 ページ\)](#)

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 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、スタック内の各スイッチの AppGigabitEthernet インターフェイスに適用する必要があります。設定がすべてのスイッチに適用されない場合、AppGigabitEthernet インターフェイスはスイッチオーバー後に機能しません。

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は、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 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。



(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。

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 AppGigabitEthernet 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 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。

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ます。Enterprise Agent はブラウザテスト (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。

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 Amsterdam 17.3.3 では、Cisco Catalyst 9300 および 9300L シリーズスイッチで ThousandEyes Enterprise Agent がサポートされます。

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 31 : ThousandEyes Enterprise Agent に必要なリソース

アプリケーションメディア	アプリケーションモード	最大リソース	サポートされるリリース
SSD (注) Cisco IOS XE Amsterdam 17.3.3 では、120G SSD のみがサポートされています。	組み込み型	<ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • Cisco Catalyst 9300 および 9300L シリーズ スイッチ

アプリケーションメディア	アプリケーションモード	最大リソース	サポートされるリリース
フラッシュ	組み込み型	<ul style="list-style-type: none"> • 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 <ul style="list-style-type: none"> • Cisco Catalyst 9300 および 9300L シリーズ スイッチ

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 32: 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
<ul style="list-style-type: none"> • 「Install Enterprise Agents on Cisco Switches with Docker」 ページ [英語] からファイルをダウンロードします。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。 	<ul style="list-style-type: none"> • 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。

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 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 33: 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。
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。

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
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。	IOx sigterm ハンドラで、SSD のクリーンなアンマウントを処理します。 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システムフラッシュで起動します。	IOx sigterm ハンドラで、SSD のクリーンなアンマウントを処理します。 SSO の後に、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すると、情報の一括同期を実行します。
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。

異なるプラットフォームでのアプリケーションの自動再起動

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_RESTARTABILITY: 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 9300 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るネットワークのタイプを示します。

表 34: 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	<ul style="list-style-type: none"> • 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
前面パネルのトランクポート	<ul style="list-style-type: none"> • 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
前面パネル VLAN ポート	<ul style="list-style-type: none"> • 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 ネットワークアドレス変換 (NAT)	<ul style="list-style-type: none"> • 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 <p>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。</p>
Cisco IOx NAT	サポート対象外



(注) AppGigabitEthernet ポートは、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 スイッチではサポートされません。

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。

ハイ アベイラビリティ

IOxは15分ごとにアプリケーションデータを同期します。データ同期の前にSSOが実行されると、最後の15分間のデータが失われる可能性があります。永続データのディスクサイズは100MBに制限されています。永続データのディスクサイズが100MBを超えるアプリケーションは、定期的に同期されない可能性があります。アプリケーションは、同期が発生する定期的な間隔を制御できません。

ThousandEyes Enterprise Agent のインストール

Enterprise Agent のインストール後、SSOを30分間待機します。インフラストラクチャの同期とアプリケーションの同期の問題により、アプリケーションのインストールとSSOの間に30分の待機時間がなければ、Enterprise Agentは新しくアクティブで再起動します。その後、Enterprise Agentはホスト名に番号が付けられた状態でThousandEyesポータルに表示されます。

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。 <ul style="list-style-type: none">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	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。 <ul style="list-style-type: none">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。 <ul style="list-style-type: none">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。 <ul style="list-style-type: none">スタック可能スイッチの場合、<i>number</i> 引数は <i>switch-number/0/1</i> 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。

	コマンドまたはアクション	目的
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。
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。 <ul style="list-style-type: none">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。 <ul style="list-style-type: none">スタック可能スイッチの場合、<i>number</i> 引数は <i>switch-number/0/1</i> 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。

	コマンドまたはアクション	目的
ステップ 9	guest-interface <i>guest-interface-number</i> 例： Device(config-config-app-hosting-trunk)# guest-interface 2	AppGigabitEthernet インターフェイストラッ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。
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。

	コマンドまたはアクション	目的
ステップ 3	app-hosting appid <i>application-name</i> 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	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 <i>application-name</i> package <i>package-path</i> 例：	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。

	コマンドまたはアクション	目的
	Device# app-hosting install appid iox_app package usbflash1:my_iox_app.tar	<ul style="list-style-type: none"> 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。
ステップ 3	app-hosting activate appid <i>application-name</i> 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。 <ul style="list-style-type: none"> 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 <i>application-name</i> 例： Device# app-hosting start appid iox_app	アプリケーションを起動します。 <ul style="list-style-type: none"> 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 <i>application-name</i> 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 <i>application-name</i> 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 <i>application-name</i> 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。 <ul style="list-style-type: none"> 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 <i>name</i> 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 <i>interface-number</i> 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 <i>ip-address netmask netmask</i> 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 <i>ip-address guest-interface network-interface</i> 例：	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
	Device(config-app-hosting)# app-default-gateway 172.19.0.23 guest-interface 0	
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。

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • <i>Mgmt-vrf</i> 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。
ステップ 5	ip address <i>ip-address mask</i> 例： <pre>Device(config-if)# ip address 198.51.100.1 255.255.255.254</pre>	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： <pre>Device(config-if)# exit</pre>	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 <i>appid name</i> 例： <pre>Device(config)# app-hosting appid iox_app</pre>	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 <i>network-interface</i> 例： <pre>Device(config-app-hosting)# app-vnic management guest-interface 1</pre>	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。 <ul style="list-style-type: none"> • 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 <i>network-interface</i> 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 <i>guest-interface 1</i> を使用しています。
ステップ 9	end 例： <pre>Device(config-app-hosting-mgmt-gateway)# end</pre>	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：	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
	Device# configure terminal	
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 <i>application-name</i> 例： Device(config)# app-hosting appid appid lkeys	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 <i>vlan-ID</i> guest-interface <i>guest-interface-number</i> 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。 <ul style="list-style-type: none"> • 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 <i>number</i> 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、 <i>number</i> 引数は <i>switch-number/0/1</i> です。
ステップ 4	switchport trunk allowed vlan <i>vlan-ID</i> 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 lkeys https://downloads.thousandeyes.com/ enterprise-agent/thousandeyes-enterprise-agent-3.0.cat9k.tar または Device# app-hosting install appid lkeys 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 lkeys	(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
-----
lkeys                                 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 running-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 lkeys
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 lkeys https://downloads.thousandeyes.com/
enterprise-agent/thousandeyes-enterprise-agent-3.0.cat9k.tar
OR
Device# app-hosting install appid lkeys package flash:/apps/[greenfield-app-tar]
Device# app-hosting start appid lkeys
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           : lkeys
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 lkeys
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 lkeys
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
<p>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。</p> <p>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。</p> <p>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。</p>	<p>http://www.cisco.com/support</p>

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 35: 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	<p>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。</p> <ul style="list-style-type: none"> 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	<p>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。</p> <p>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。</p> <ul style="list-style-type: none"> • 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
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	<p>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。</p> <ul style="list-style-type: none"> • 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：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。 <ul style="list-style-type: none"> 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：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。 <ul style="list-style-type: none"> 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。 <p>(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。</p>

機能名	リリース	機能情報
ThousandEyes BrowserBot	Cisco IOS XE Bengaluru 17.6.1	<p>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립ト化テスト用の BrowserBot を提供します。</p> <p>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。</p> <ul style="list-style-type: none"> • 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
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	<p>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。</p> <ul style="list-style-type: none"> • 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

- [OpenFlow \(367 ページ\)](#)



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](#) (367 ページ)
- [OpenFlow の制約事項](#) (367 ページ)
- [OpenFlow について](#) (368 ページ)
- [OpenFlow の設定方法](#) (373 ページ)
- [OpenFlow の確認](#) (378 ページ)
- [OpenFlow の設定例](#) (381 ページ)
- [その他の参考資料](#) (382 ページ)
- [OpenFlow の機能情報](#) (382 ページ)

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)、スパン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、セキュアなチャンネルを介して、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。

Cisco Catalyst 9000 シリーズ スイッチは最大 9 個のフロー テーブルをサポートしています。

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入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 36: 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。

フローは、パケットの次のヘッダーフィールドも変更できます。

- イーサネットの宛先 MAC アドレス
- イーサネットの送信元 MAC アドレス
- VLAN ID

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。

OpenFlow スケール情報

表 37: 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
256/512 ビット ACL TCAM エントリ	18K/9K	54K/27K	54K/27K
統計情報/フレキシブル カウンタ (バイト+フ レーム)	192K	192K	302K

システムが起動すると、デフォルトでは、フローごとに2つのテーブルエントリがあります。テーブル機能メッセージでは、フローは単一のエントリを使用します。ただし、IPv6_SRC_AND_IPv6_DEST のキーにはフローごとに2つのエントリが必要です。

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（一部またはすべて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ります）。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。

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	<ul style="list-style-type: none"> • delete flash:vlan.dat • delete flash:stby-vlan.dat 	<ul style="list-style-type: none"> • VLAN 情報を保存する vlan.dat ファイルを削除します。

	コマンドまたはアクション	目的
	例 : Device# delete flash:vlan.dat Device# delete flash:stby-vlan.dat	<ul style="list-style-type: none"> • 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。
ステップ 8	enable 例 : Device> enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none"> • 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： 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	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witch の Transport Layer Security (TLS) トラストポイントを設定します。

	コマンドまたはアクション	目的
ステップ 9	end 例： Device(config-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 bpdupfilter 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 <i>type number</i> 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。

	コマンドまたはアクション	目的
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： Device(config-if)# switchport nonnegotiate	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。
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 bpdupfilter enable 例： Device(config-if)# spanning-tree bpdupfilter 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        Tel1/1/1        PORT_UP       LINK_DOWN   10GB-FD
30        Tel1/1/2        PORT_UP       LINK_DOWN   10GB-FD
31        Tel1/1/3        PORT_UP       LINK_DOWN   10GB-FD
32        Tel1/1/4        PORT_UP       LINK_DOWN   10GB-FD
33        Tel1/1/5        PORT_UP       LINK_DOWN   10GB-FD
34        Tel1/1/6        PORT_UP       LINK_DOWN   10GB-FD
35        Tel1/1/7        PORT_UP       LINK_DOWN   10GB-FD
36        Tel1/1/8        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        Twel1/1/1       PORT_UP       LINK_DOWN   10GB-FD
40        Twel1/1/2       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/
Faucet OpenFlow コントローラ	<ul style="list-style-type: none"> • https://faucet.nz/ • https://docs.faucet.nz/en/latest/
PoE	<ul style="list-style-type: none"> • 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
<p>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。</p> <p>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。</p> <p>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。</p>	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、特に断りが無い限り、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 38: 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	<p>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。</p> <p>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。</p> <ul style="list-style-type: none"> • 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	<p>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導入されました。</p>
OpenFlow Power over Ethernet	Cisco IOS XE Gibraltar 16.12.1	<p>PoE は OpenFlow ポートでサポートされます。</p> <p>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。</p> <ul style="list-style-type: none"> • Catalyst 9300 シリーズ スイッチ • 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。