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、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。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。システムイメージとともにバンドルされており、**guestshell enable** コマンド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。

次の表は、ゲスト シェルのさまざまな機能とサポート対象のプラットフォームに関する情報を提供します。

表 *1: Cisco* ゲスト シェルの機能

	ゲスト シェル Lite（限定的な LXC コンテナ）	**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	Cisco IOS XE
サポートされるプラットフォーム	• Cisco Catalyst 3650 シリーズ スイッチ（全モデル） • Cisco Catalyst 3850 シリーズ スイッチ（全モデル）	• Cisco ISR 4000 シリーズ サービス統合型ルータ（最低 8 GB の RAM を有するモデル）
ゲスト シェル環境	Montavista CGE7	CentOS 7
Python 2.7	サポート対象（Python V2.7.11）	サポート対象（Python V2.7.5）
カスタムの Python ライブラリ	• Cisco 組込イベント マネージャ • Cisco IOS XE CLI • Ncclient	• Cisco 組込イベント マネージャ • Cisco IOS XE CLI
サポートされる rootfs	Busybox、SSH、および Python PIP のインストール	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	サポート対象外
RPM のインストール	サポート対象外	サポートあり
アーキテクチャ	MIPS	x86

ゲスト シェルとゲスト シェル Lite

ゲストシェル コンテナを使用すると、ユーザは、システム上で自分のスクリプトやアプリケーションを実行できるようになります。Intel x86 プラットフォーム上のゲスト シェル コンテナは、CentOS 7.0 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 7.0 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。

Catalyst 3650 や Catalyst 3850 シリーズ スイッチなどの MIPS プラットフォーム上のゲスト シェル Lite コンテナには、Carrier Grade Edition（CGE）7.0 の rootfs があります。ゲスト シェル Lite では、スクリプトのインストールまたは実行のみ可能です。これらのデバイスでは、Yum のインストールはサポートされていません。

ゲスト シェルのセキュリティ

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

ゲスト シェルのハードウェア要件

この項では、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 *2 : Catalyst* スイッチでのゲスト シェルのサポート

プラットフォーム	デフォルトの **DRAM**	ゲスト シェルのサポート
WS-3650-xxx（すべて）	4 GB	サポート対象
WS-3850-xxx（すべて）	4 GB	サポート対象
C9300-xx-x（すべて）	8 GB	サポート対象
C9500-24Q-x（すべて）	16 GB	サポート対象

Catalyst 3850 シリーズ スイッチの最小システム要件は、4 GB の DRAM です。

表 *3 : ISR 4000* シリーズ サービス統合型ルータでのゲスト シェルのサポート

プラットフォーム	デフォルトの **DRAM**	ゲスト シェルのサポート
ISR 4221	4GB	未サポート
ISR 4321	4 GB	未サポート
	8 GB	サポート対象
ISR 4331	8 GB	サポート対象
	16 GB	サポート対象
ISR 4351	8 GB	サポート対象
	16 GB	サポート対象

プラットフォーム	デフォルトの **DRAM**	ゲスト シェルのサポート
ISR 4431	8 GB	サポート対象
	16 GB	サポート対象
ISR 4451	8 GB	サポート対象
	16 GB	サポート対象

ISR 4000 シリーズ サービス統合型ルータの最小システム要件は、8 GB の DRAM です。

（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

Catalyst 3650 および Catalyst 3850 シリーズ スイッチでは、ゲスト シェルは、フラッシュのファイル システムにのみインストールできます。Catalyst 3850 シリーズ スイッチのブートフラッシュでは、ゲスト シェルを正常にインストールするには 75 MB の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）のサービス セット識別子（SSID）（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 SSID）空き容量が必要です。

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、ISR 4000 シリーズ 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 Catalyst 3850 シリーズ スイッチでは、ゲスト シェルが使用できるストレージ 容量は 18 MB です。Cisco 4000 シリーズ サービス統合型ルータでは、ゲスト シェルが使用できるストレージ容量は 800 MB です。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 *4:* 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	256/256 MB

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、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、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 IOS 端末にログインしたときと同じ特権で実行されます。

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

管理ポートを介してのゲスト シェルへのアクセス

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

（注） 管理ポートがないプラットフォームの場合、VirtualPortGroup を 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。

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、gs_script 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、gs_script ディレクトリの内容のみが、すべてのスタック メンバー間で同期されます。ハイ アベイラビリティでの切り替えの際にデータを保持するには、このディレクトリにデータを格納します。

ハイ アベイラビリティでの切り替えの際には、新しいアクティブ デバイスは、それぞれのゲスト シェル インストールを作成します。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、切り替え時に維持されます。

IOx の概要

IOx は Cisco が開発したエンド 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、システムの開発および使用に役立つアプリケーションの 1 つです。

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化します。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。

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 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）。
- 一元的なアプリケーション ライフ サイクル管理。
- クラウド ベースの開発。

例：ゲスト シェルのネットワーキング設定

ゲスト シェルのネットワーキングでは、次の設定が必要です。

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

ゲスト シェルを有効にする方法

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Libirtd サービスが実行している必要があります。

手順

	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： `Device# show app-hosting list`	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。

次のタスク

次に、ISR 4000 シリーズ ルータでの **show iox-service** コマンドの出力例を示します。

```
Device# show iox-service

Virtual Service Global State and Virtualization Limits:
```

```
Infrastructure version : 1.7
Total virtual services installed : 0
Total virtual services activated : 0

Machine types supported   : KVM, LXC
Machine types disabled    : none

Maximum VCPUs per virtual service : 6
Resource virtualization limits:
Name                         Quota     Committed     Available
------------------------------------------------------------------
system CPU (%)                  75             0            75
memory (MB)                  10240             0         10240
bootflash (MB)                1000             0          1000
harddisk (MB)                20000             0         18109
volume-group (MB)           190768             0        170288


IOx Infrastructure Summary:
---------------------------
IOx service (CAF)    : Running
IOx service (HA)     : Not Running
IOx service (IOxman) : Running
Libvirtd             : Running
```

次に示すのは、Catalyst 3850 シリーズ スイッチでの **show iox-service** コマンドの短縮された出力例です。

```
Device# show iox-service

IOx Infrastructure Summary:
---------------------------
IOx service (CAF)    : Running
IOx service (HA)     : Running
IOx service (IOxman) : Running
Libvirtd             : Running
```

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。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	• **guestshell enable** 例： `Device# guestshell enable`	ゲスト シェル サービスの有効化。 （注） • **guestshell enable** コマンドは、ネットワーキングに管理 Virtual Routing and Forwarding（VRF）インスタンスを使用します。 • フロント パネル ネットワー キングに VirtualPortGroups（VPG）を使用している場合は、まず VPG を構成する必要があります。 • ゲスト IP アドレスとゲートウェイ IP アドレスは同じサブネット内にある必要があります。
ステップ 3	**guestshell run** *linux-executable* 例： `Device# guestshell run python`	ゲスト シェルで Linux プログラムを実行します。
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。

	コマンドまたはアクション	目的
ステップ 6	**guestshell destroy** 例： `Device# guestshell destroy`	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。

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。

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） **guestshell run** コマンドは、Linux 実行可能ファイルの実行に相当する IOS であり、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/sample_script.py parameter1 parameter2
```

ゲスト シェルの設定例

例：ゲスト シェルの管理

次の例では、Catalyst 3850 シリーズ スイッチ上でゲスト シェルを有効にする方法を示しています。

```
Device> enable
Device# guestshell enable

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

Device# guestshell run python

Python 2.7.11 (default, Feb 21 2017, 03:39:40)
[GCC 5.3.0] 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
```

VirtualPortGroup 設定の例

ゲスト シェル ネットワーキングに VirtualPortGroup インターフェイスを使用する場合、VirtualPortGroup インターフェイスには設定済みの静的 IP アドレスが必要です。フロント ポート インターフェイスはインターネットに接続されている必要があり、ネット

ワーク アドレス変換（NAT）は VirtualPortGroup とフロント パネル ポートの間で設定されている必要があります。

次に示すのは、VirtualPortGroup の設定例です。

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

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 3.10.101.cge-rt110 #1 SMP Sat Feb 11 00:33:02
PST 2017 mips64 GNU/Linux
```

Catalyst 3650 および Catalyst 3850 シリーズ スイッチには、BusyBox が提供する定義された一連の Linux 実行可能ファイルがあり、Cisco 4000 シリーズ サービス統合型ルータには、CentOS Linux リリース 7.1.1503 が提供するコマンドがあります。

次の例は、Catalyst 3850 シリーズ スイッチ上での **dohost** コマンドの使用を示しています。

```
[guestshell@guestshell ~]$ dohost "show version"

Cisco IOS Software [Everest], Catalyst L3 Switch Software [CAT3K_CAA-UNIVERSALK9-M),
Experimental Version 16.5.2017200014[v165_throttle-BLD-
 BLD_V165_THROTTLE_LATEST_20170531_192849 132]
```

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

例：ゲスト シェルのネットワーキング設定

ゲスト シェルのネットワーキングでは、次の設定が必要です。

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

ゲスト シェルの DNS 設定の例

ゲスト シェルのサンプル DNS 構成は次のとおりです。

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

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

例：プロキシ環境変数の設定

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

次の例は、プロキシ変数を設定する方法を示しています。

```
[guestshell@guestshell ~]$cat /bootflash/proxy_vars.sh
export http_proxy=http://proxy.example.com:80/
export https_proxy=http://proxy.example.com:80/
export ftp_proxy=http://proxy.example.com:80/
export no_proxy=example.com
export HTTP_PROXY=http://proxy.example.com:80/
export HTTPS_PROXY=http://proxy.example.com:80/
export FTP_PROXY=http://proxy.example.com:80/
guestshell ~] source /bootflash/proxy_vars.sh
```

例：プロキシ設定用の Yum および PIP の構成

次の例は、プロキシ環境変数の設定に Yum を使用する方法を示しています。

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

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

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

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

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

ゲスト シェルに関するその他の参考資料

関連資料

関連項目	マニュアル タイトル
	『Programmability Command Reference, Cisco IOS XE Everest 16.6.1』
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

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

シスコのテクニカル サポート

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

ゲスト シェルの機能情報

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

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

表 *5:* ゲスト シェルの機能情報

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