



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。

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 \(1 ページ\)](#)
- [アプリケーションホスティングに関する情報 \(2 ページ\)](#)
- [アプリケーションホスティングの設定方法 \(6 ページ\)](#)
- [アプリケーションホスティングの設定例 \(19 ページ\)](#)
- [アプリケーションホスティングに関する機能情報 \(21 ページ\)](#)

アプリケーションホスティングの制約事項

Catalyst 9300 シリーズスイッチ、**Catalyst 9400** シリーズスイッチ、および **Catalyst 9500** シリーズスイッチの制約事項

- ネットワーク アドレス変換 (NAT) はサポートされていません。
- アプリケーションホスティングは、Virtual Routing and Forwarding (VRF) に対応していません。
- アプリケーションホスティングでは、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 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、背面パネルのシスコ認定 USB フラッシュが使用可能な場合に、アプリケーションホスティングが実行されます。
- Cisco IOS XE Fuji 16.9.1 では、前面パネルの USB スティックは Cisco Catalyst 9400 シリーズスイッチではサポートされていません。アプリケーションホスティングにはハードディスク ドライブ (HDD) のみがサポートされています。

アプリケーションホスティングに関する情報

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、再利用ができポータブルでスケーラブルなアプリケーションを開発する必要性が高まってきました。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。

Cisco のデバイスは、Linux ツール チェーンを使用して開発されたサードパーティ製の市販アプリケーションをサポートしています。ユーザは、シスコが提供するソフトウェア開発キットと相互にコンパイルされたカスタム アプリケーションを実行できます。アプリケーションホスティングは、「Kernel Virtual Machine (KVM)」と「コンテナ」という2つの形態で提供されます。

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。

シスコ 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 ログिंग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。
- VirtualPortGroup および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。

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「仮想化環境」と呼ばれます。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。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せん。

デバイス上の管理インターフェイスにより、アプリケーション ホスティング ネットワークが IOS 管理インターフェイスに接続します。アプリケーションのレイヤ 3 インターフェイスは、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/VM 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。

IOXMAN

IOXMAN は、シリアルデバイスをエミュレートする Libvirt を除く、ゲスト アプリケーションのログングまたはトレース サービスを提供するトレース インフラストラクチャを確立するプロセスです。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、ログング データを 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。

Catalyst 9000 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 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 9500 シリーズ ハイ パフォーマンス スイッチに固有の、アプリケーションホスティングの特性について説明します。

これらのスイッチは、Linux コンテナでホストされるサードパーティ製アプリケーションに前面パネルからアクセスするための VirtualPortGroup インターフェイスをサポートしています。

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、『インターフェイスおよびハードウェア コンフィギュレーション ガイド』の「USB 3.0 SSD の設定」の章を参照してください。

Cisco Catalyst 9400 シリーズ スイッチは、リムーバブルのスーパーバイザに装着できる M2 Serial Advanced Technology Attachment (SATA) ドライブをサポートしています。前面パネルの USB もサポートされています。ただし、Cisco Catalyst 9400 シリーズ スイッチの前面パネルの USB ではアプリケーションホスティングはサポートされていません。Cisco IOS XE Gibraltar 16.10.1 以降のリリースでは、Cisco Catalyst 9400 シリーズ スイッチは USB 3.0 を使用したアプリケーションホスティングをサポートしています。

Cisco Catalyst 9500 シリーズ ハイ パフォーマンス スイッチは、Cisco Catalyst 9300 シリーズ スイッチおよび Cisco Catalyst 9500 シリーズ スイッチと同様に、背面パネルの USB 3.0 をサポートしています。

VirtualPortGroup

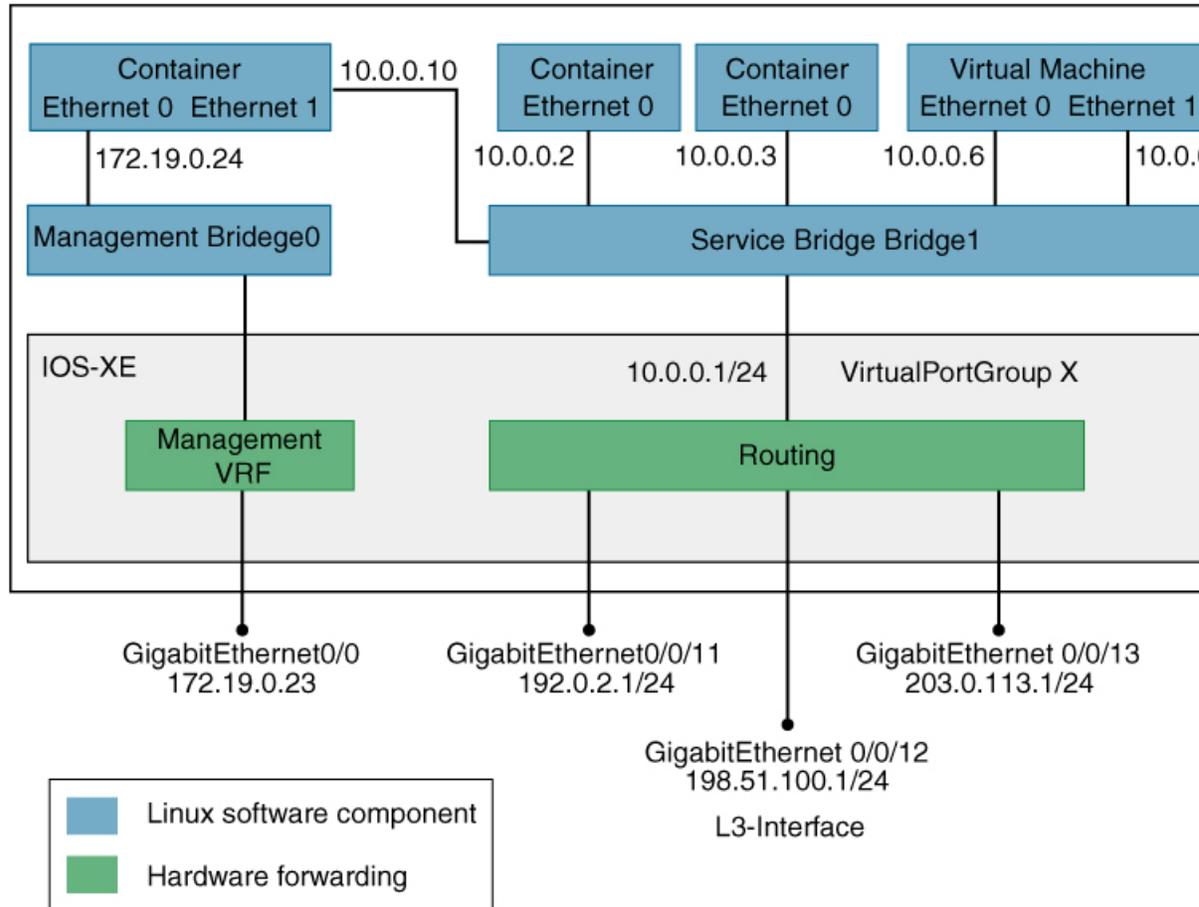
VirtualPortGroup は、Linux ブリッジ IP アドレスにマッピングする Cisco IOS 上のソフトウェア構成要素です。そのため、VirtualPortGroup は、Linux コンテナのスイッチ仮想インターフェイス (SVI) を表します。各ブリッジには、複数のインターフェイスを含めることができ、それぞれ異なるコンテナへマッピングされます。各コンテナには、複数のインターフェイスを含めることもできます。

VirtualPortGroup インターフェイスは、**interface virtualportgroup** コマンドを使用して設定します。これらのインターフェイスが作成されると、IP アドレスとその他のリソースが割り当てられます。

最大 32 個の VirtualPortGroup インターフェイスを設定できます。これらの VirtualPortGroup インターフェイスはそれぞれ 2 つの転送エントリを持ちます。

VirtualPortGroup 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 IOS ルーティングドメインに接続します。アプリケーションのレイヤ 3 インターフェイスは、IOS からルーティングされたトラフィックを受信します。VirtualPortGroup インターフェイスは、SVC ブリッジを介してコンテナ/VM インターフェイスに接続します。

図 1: アプリケーションホスティングのネットワーキング



vNIC

コンテナのライフサイクル管理には、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vNIC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。vpgXと呼ばれるもう1つのインターフェイスは、ホストシステムの一部です。

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。IOxは、コンテナ内のvNICごとに、ゲートウェイ (VirtualPortGroupインターフェイス)、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/VM内のvNICは、標準のイーサネットインターフェイスと見なされます。

アプリケーションホスティングの設定方法

IOx の有効化

IOx Local Manager へのアクセスを有効にするには、次の作業を実行します。Local Manager を使用することで、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. **ip http server**
5. **ip http secure-server**
6. **username name privilege level password {0 | 7 | user-password} encrypted-password**
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	iox 例： Device(config)# iox	IOx をイネーブルにします。
ステップ 4	ip http server 例： Device(config)# ip http server	IP または IPv6 システム上の HTTP サーバを有効化します。
ステップ 5	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。

	コマンドまたはアクション	目的
ステップ 6	username name privilege level password {0 7 user-password} encrypted-password 例： <pre>Device(config)# username cisco privilege 15 password 0 ciscoI</pre>	ユーザ名ベースの認証システムとユーザの権限レベルを確立します。 <ul style="list-style-type: none"> ユーザ名の特権レベルは 15 に設定する必要があります。
ステップ 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。

レイヤ3 データ ポートへの VirtualPortGroup の設定

複数のレイヤ3 データポートを1つ以上の VirtualPortGroup またはコンテナにルーティングできます。VirtualPortGroups とレイヤ3 のデータポートは、異なるサブネット上にある必要があります。

レイヤ3 データポートで外部ルーティングを許可するには、**ip routing** コマンドを有効にします。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip routing**
4. **interface type number**
5. **no switchport**
6. **ip address ip-address mask**
7. **exit**
8. **interface type number**
9. **ip address ip-address mask**
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： <pre>Device> enable</pre>	特権 EXEC モードをイネーブルにします。 <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	ip routing 例： Device(config)# ip routing	IP ルーティングをイネーブルにします。 • レイヤ3データポートで外部ルーティングを許可するには、 ip routing コマンドを有効にする必要があります。
ステップ 4	interface type number 例： Device(config)# interface gigabitethernet 1/0/1	インターフェイスを設定し、インターフェイス コンフィギュレーションモードを開始します。
ステップ 5	no switchport 例： Device(config-if)# no switchport	インターフェイスをレイヤ3モードにし、スイッチポートではなくルータ インターフェイスのように動作させます。
ステップ 6	ip address ip-address mask 例： Device(config-if)# ip address 10.1.1.1 255.255.255.254	インターフェイスに IP アドレスを設定します。
ステップ 7	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 8	interface type number 例： Device(config)# interface virtualportgroup 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。
ステップ 9	ip address ip-address mask 例： Device(config-if)# ip address 192.0.2.1 255.255.255.1	インターフェイスに IP アドレスを設定します。
ステップ 10	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。

アプリケーションのインストールとアンインストール

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 モードをイネーブルにします。 <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	app-hosting install appid <i>application-name</i> package <i>package-path</i> 例： Device# app-hosting install appid lxc_app package flash:my_iox_app.tar	指定された場所からアプリケーションをインストールします。 <ul style="list-style-type: none"> • アプリケーションは、flash、bootflash、usbflash0、usbflash1、harddisk などの任意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 lxc_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 lxc_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 lxc_app	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 lxc_app	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 lxc_app	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。

アプリケーションの IP アドレスの手動設定

次の方法で、Kernel-based Virtual Machine (KVM) または Linux コンテナ (LXC) の IP アドレスを設定できます。

- KVM または LXC に直接ログインし、**ifconfig** Linux コマンドを設定します。
- KVM または LXC で Dynamic Host Configuration Protocol (DHCP) を有効にし、IOS 設定で DHCP サーバリレーを設定します。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **vrf forwarding** *vrf-name*
5. **ip address** *ip-address mask*
6. **exit**
7. **interface** *type number*
8. **ip address** *ip-address mask*
9. **exit**
10. **app-hosting appid** *name*
11. **app-vnic gateway virtualportgroup** *number* **guest-interface** *network-interface*
12. **exit**
13. **app-vnic management** *guest-interface* *network-interface*
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	interface <i>type number</i> 例： Device(config)# interface gigabitethernet 0/0	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。
ステップ 4	vrf forwarding <i>vrf-name</i> 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
ステップ 5	ip address <i>ip-address mask</i> 例 : Device(config-if)# ip address 198.51.100.1 255.255.255.254	インターフェイスに IP アドレスを設定します。
ステップ 6	exit 例 : Device(config-if)# exit	インターフェイス コンフィギュレーション モードを終了し、グローバルコンフィギュレーションモードに戻ります。
ステップ 7	interface <i>type number</i> 例 : Device(config)# interface VirtualPortGroup 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。
ステップ 8	ip address <i>ip-address mask</i> 例 : Device(config-if)# ip address 192.0.2.1 255.255.255.1	インターフェイスに IP アドレスを設定します。
ステップ 9	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 10	app-hosting <i>appid name</i> 例 : Device(config)# app-hosting appid lxc_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 11	app-vnic gateway <i>virtualportgroup number</i> guest-interface <i>network-interface</i> 例 : Device(config-app-hosting)# app-vnic gateway1 virtualportgroup 0 guest-interface 1	ゲストインターフェイスを VirtualPortGroup に接続し、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。 <ul style="list-style-type: none"> • gateway キーワードは、定義されたポートマッピングを表す一意の識別子です。任意のゲートウェイ番号を指定できます。この例では gateway1 を使用しています。 • numbervirtualportgroup のキーワード引数ペアは、コンテナへの接続に使用される IOS VirtualPortGroup インターフェイス番号を指定します。この例では virtualportgroup 0 を使用しています。 • guest-interface <i>network-interface</i> のキーワード引数ペアは、指定された VirtualPortGroup 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、

	コマンドまたはアクション	目的
		コンテナのイーサネット1インターフェイスに対して <code>guest-interface 1</code> を使用しています。
ステップ 12	exit 例： <pre>Device(config-app-hosting-gateway0)# exit</pre>	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 13	app-vnic management guest-interface network-interface 例： <pre>Device(config-app-hosting)# 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 IOS 管理 GigabitEthernet0/0 インターフェイスを指定します。 • network-interface guest-interface のキーワード引数ペアは、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 <code>guest-interface 1</code> を使用しています。
ステップ 14	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。

LXCでの静的IPアドレスの設定

- Kernel-based Virtual Machine (KVM) は、アプリケーションホスティングの静的IP設定をサポートしていません。
- 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。

LXCのIPアドレスは、IOSコマンドを使用して設定できます。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface type number**
4. **vrf forwarding vrf-name**
5. **ip address ip-address mask**

6. **exit**
7. **interface** *type number*
8. **ip address** *ip-address mask*
9. **exit**
10. **app-hosting appid** *name*
11. **app-vnic gateway virtualportgroup** *number guest-interface network-interface*
12. **guest-ipaddress** *ip-address netmask netmask*
13. **exit**
14. **name-server** *ip-address*
15. **app-vnic management guest-interface** *interface-number*
16. **guest-ipaddress** *ip-address netmask netmask*
17. **exit**
18. **app-default-gateway** *ip-address guest-interface network-interface*
19. **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	interface <i>type number</i> 例： Device(config)# interface gigabitethernet 0/0	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。
ステップ 4	vrf forwarding <i>vrf-name</i> 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。
ステップ 5	ip address <i>ip-address mask</i> 例： Device(config-if)# ip address 198.51.100.1 255.255.255.254	インターフェイスに IP アドレスを設定します。
ステップ 6	exit 例： Device(config-if)# exit	インターフェイス コンフィギュレーション モードを終了し、グローバルコンフィギュレーションモードに戻ります。
ステップ 7	interface <i>type number</i> 例：	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
	<code>Device(config)# interface VirtualPortGroup 0</code>	
ステップ 8	ip address <i>ip-address mask</i> 例： <code>Device(config-if)# ip address 192.0.2.1 255.255.255.1</code>	インターフェイスに IP アドレスを設定します。
ステップ 9	exit 例： <code>Device(config-if)# exit</code>	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 10	app-hosting <i>appid name</i> 例： <code>Device(config)# app-hosting appid lxc_app</code>	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 11	app-vnic gateway <i>virtualportgroup number guest-interface network-interface</i> 例： <code>Device(config-app-hosting)# app-vnic gateway1 virtualportgroup 0 guest-interface 1</code>	仮想ネットワーク インターフェイス ゲートウェイおよびゲスト インターフェイスの詳細を設定し、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 12	guest-ipaddress <i>ip-address netmask netmask</i> 例： <code>Device(config-app-hosting-gateway1)# guest-ipaddress 10.0.0.3 netmask 255.255.255.0</code>	ゲストの IP アドレスとマスクを設定します。
ステップ 13	exit 例： <code>Device(config-app-hosting-gateway1)# exit</code>	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 14	name-server <i>ip-address</i> 例： <code>Device(config-app-hosting)# name-server0 10.2.2.2</code>	DNS サーバを設定します。
ステップ 15	app-vnic management <i>guest-interface interface-number</i> 例： <code>Device(config-app-hosting)# app-vnic management guest-interface 0</code>	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 16	guest-ipaddress <i>ip-address netmask netmask</i> 例： <code>Device(config-app-hosting-mgmt-gateway)# guest-ipaddress 172.19.0.24 netmask 255.255.255.0</code>	管理ゲストインターフェイスの詳細を設定します。

	コマンドまたはアクション	目的
ステップ 17	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 18	app-default-gateway ip-address guest-interface network-interface 例： Device (config-app-hosting-mgmt-gateway) # 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 19	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リソース設定の上書き

リソースの変更は、**app-hosting activate** コマンドが設定された後にのみ有効になります。

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。

	コマンドまたはアクション	目的
ステップ 3	app-hosting appid name 例： Device(config)# app-hosting appid lxc_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。

アプリケーションホスティングを有効にするための LVM の作成



(注) このタスクは Cisco Catalyst 9300 シリーズスイッチにのみ適用されます。

ドライブでアプリケーションホスティングを有効にするには、ローカルボリュームマネージャ (LVM) を作成する必要があります。

手順の概要

1. **enable**
2. 次のいずれかのコマンドを使用します。
 - **request platform hardware filesystem ssdflash-slot: initialize**
 - **format ssdflash:LVM**
3. **request platform hardware filesystem ssdflash- slot: sanitize**
4. **show usbflash1: fileys**
5. **show file systems**

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	次のいずれかのコマンドを使用します。 • request platform hardware filesystem ssdflash-slot: initialize • format ssdflash:LVM	ドライブが LVM モードに入り、フォーマットされます。
ステップ 3	request platform hardware filesystem ssdflash- slot: sanitize	ドライブが raw デバイスの状態に戻ります。すべてのデータがドライブ上で消去されます。
ステップ 4	show usbflash1: fileys	LVM ファイル システムの情報を表示します。
ステップ 5	show file systems	ファイル システムの全般情報を表示します。

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の確認

手順の概要

1. **enable**
2. **show iox-service**
3. **show app-hosting detail**
4. **show app-hosting list**

手順の詳細

ステップ 1 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）。

例：

```
Device> enable
```

ステップ2 show iox-service

すべての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)      : Running
IOx Service (HA)      : Running
IOx Service (IOxman)  : Running
LibvirtD               : Running
```

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        : Lubuntu 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 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
```

アプリケーションホスティングの設定例

例：IOxの有効化

```
Device> enable
Device# configure terminal
Device(config)# iox
Device(config)# ip http server
Device(config)# ip http secure-server
Device(config)# username cisco privilege 15 password 0 ciscoI
Device(config)# end
```

例：レイヤ3データポートへのVirtualPortGroupの設定

```
Device> enable
Device# configure terminal
Device(config)# ip routing
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# no switchport
Device(config-if)# ip address 10.1.1.1 255.255.255.254
Device(config-if)# exit
Device(config)# interface virtualportgroup 0
Device(config-if)# ip address 192.0.2.1 255.255.255.1
Device(config-if)# end
```

例: アプリケーションのインストールとアンインストール

```
Device> enable
Device# app-hosting install appid lxc_app package flash:my_iox_app.tar.tar
Device# app-hosting activate appid lxc_app
Device# app-hosting start appid lxc_app
Device# app-hosting stop appid lxc_app
Device# app-hosting deactivate appid lxc_app
Device# app-hosting uninstall appid lxc_app
```

例：アプリケーションのIPアドレスの手動設定

```
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)# interface virtualportgroup 0
Device(config-if)# ip address 192.0.2.1 255.255.255.1
Device(config-if)# exit
Device(config)# app-hosting appid lxc_app
Device(config-app-hosting)# app-vnic gateway1 virtualportgroup 0 guest-interface 1
Device(config-app-hosting-gateway0)# exit
Device(config-app-hosting)# vnic management guest-interface 1
Device(config-app-hosting-mgmt-gateway)# end
```

例：LXCでの静的IPアドレスの設定

```
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)# interface virtualportgroup 0
Device(config-if)# ip address 192.0.2.1 255.255.255.1
Device(config-if)# exit
Device(config)# app-hosting appid lxc_app
Device(config-app-hosting)# app-vnic gateway1 virtualportgroup 0 guest-interface 1
Device(config-app-hosting-gateway1)# guest-ipaddress 10.0.0.3 netmask 255.255.255.0
Device(config-app-hosting-gateway1)# exit
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-mgmt-gateway)# app-default-gateway 172.19.0.23 guest-interface 0
Device(config-app-hosting)# end
```

例：アプリケーションのリソース設定の上書き

```
Device# configure terminal
Device(config)# app-hosting appid lxc_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
```

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 1: 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 Fuji 16.9.1	<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> <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 9400 シリーズ スイッチ • Cisco Catalyst 9500 シリーズ スイッチ

