



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 ホスティングの注意事項と制限事項 (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 (3 ページ)
- アプリケーション データのコピー (13 ページ)
- アプリケーション データの削除 (14 ページ)
- アプリケーション ホスティング設定の確認 (14 ページ)
- アプリケーション ホスティングの設定例 (17 ページ)
- その他の参考資料 (18 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 (19 ページ)

アプリケーション ホスティングの注意事項と制限事項

この項では、アプリケーション ホスティング機能の注意事項と制限事項について示します。

- コンテナごとに 1 つのインターフェイスのみがサポートされます。
- Cisco NX-OS リリース 10.3(3)F 以降、アプリケーション ホスティング機能は Cisco Nexus 3600 N3K-C36180YC-R、N3K-C3636C-R、および N3K-C36480LD-R2 PID でのみサポートされます。

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、再利用可能でポータブル、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トされるアプリケーションは、自動化、設定管理のモニタリング、既存のツールチェーンとの統合など、さまざまな目的で使用できます。



(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。

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、NX-OS 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。
- 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。
- 特別なアプリケーションブリッジインターフェイスを介する、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）をセットアップする。

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 NX-OS アプリケーションホスティング機能は、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 NX-OS 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 または .tar.gz ファイルとしてパッケージ化されています。これらのアプリケーションに固有の構成も、.tar または .tar.gz ファイルの一部としてパッケージ化されています。

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。

アプリケーションホスティング機能の有効化

このタスクを実行して、シスコのアプリケーションホスティング機能を有効にします。この機能は、ホストシステム上のアプリケーションの管理、管轄、モニター、トラブルシューティングのためのユーザーインターフェイス コマンドと API インターフェイスを有効にし、関連する様々な活動を実行できるようにします。

手順の概要

1. **configure terminal**
2. **feature app-hosting**
3. **end**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： <pre>switch# configure terminal</pre>	グローバル コンフィギュレーション モードを開始します。
ステップ 2	feature app-hosting 例： <pre>switch(config)# feature app-hosting</pre>	シスコ アプリケーション ホスティング 機能を有効にします。
ステップ 3	end 例： <pre>switch(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。

■ アプリケーションホスティング ブリッジ接続の設定

アプリケーションホスティング ブリッジ接続の設定

アプリケーションコンテナへのレイヤ3接続には、独自のエンドポイント IPv4 アドレスが必要です。NX-OS では、アプリケーションホスティングブリッジと呼ばれる仮想ブリッジメカニズムが、Cisco Nexus スイッチ内のアプリケーションコンテナをホストします。

ブリッジは、アプリケーションコンテナへのゲートウェイとして機能し、アタッチされたVRF ルーティングコンテキストにトラフィックをルーティングするのに役立ちます。ブリッジは、VRF コンテキストごとに、スイッチインターフェイスを介してアプリケーションのサブネット トラフィックを転送します。

スイッチインターフェイスを介するネットワーク接続を使用したアプリケーションコンテナのホスティングには、少なくとも2つの割り当て可能なアドレスを持つ専用のエンドポイント IP サブネットが必要です。1つのIPアドレスはアプリケーションコンテナのゲストインターフェイス用で、もう1つのIPアドレスはアプリケーションコンテナのゲートウェイ用です。

内部的には、アプリケーションコンテナのゲストインターフェイスは、アプリケーションホスティング仮想ブリッジからは独立した、仮想ネットワークインターフェイスカード (vNIC) です。

手順の概要

1. **configure terminal**
2. **app-hosting bridge bridge-index**
3. **ip address ip-address/mask**
4. **vrf member name**
5. **exit**
6. **app-hosting appid name**
7. **app-vnic gateway bridge bridge-index guest-interface guest-interface-number**
8. **guest-ipaddress ip-address/mask**
9. **exit**
10. **app-default-gateway ip-address guest-interface guest-interface**
11. **end**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ1	configure terminal 例： switch# 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2	app-hosting bridge bridge-index 例： switch(config)# app-hosting bridge 1	アプリケーションホスティングブリッジを設定し、アプリケーションホスティングブリッジ構成モードを開始します。

	コマンドまたはアクション	目的
		• <1-8> ブリッジインデックス
ステップ 3	ip address ip-address/mask 例： <pre>switch(config-app-hosting-bridge)# ip address 172.25.44.1/30</pre>	アプリケーションコンテナへのゲートウェイとして機能するアプリケーションブリッジIPv4アドレスを設定します。 (注) IPがインターフェイスまたは仮想IPのいずれかによって使用されている場合、サブネットは拒否されます。
ステップ 4	vrf member name 例： <pre>switch(config-app-hosting-bridge)# vrf member overlay-VRF</pre>	VRFコンテキストを設定します。構成していなければ、VRFのデフォルトに属しています。
ステップ 5	exit 例： <pre>switch(config-app-hosting-bridge)# exit</pre>	アプリケーションブリッジ構成モードを終了して、グローバル構成モードに戻ります。
ステップ 6	app-hosting appid name 例： <pre>switch(config)# app-hosting appid te_app</pre>	アプリケーションを構成し、アプリケーションホスティング構成モードを開始します。
ステップ 7	app-vnic gateway bridge bridge-index guest-interface guest-interface-number 例： <pre>switch(config-app-hosting)# app-vnic gateway bridge 1 guest-interface 0</pre>	アプリケーションのゲストVNIC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vnicインターフェイスモードを開始します。 (注) Cisco NX-OS リリース 10.3(3)F 以降では、1つのVNICのみを構成できます。
ステップ 8	guest-ipaddress ip-address/mask 例： <pre>switch(config-app-hosting-app-vnic)# guest-ipaddress 172.25.44.2/30</pre>	ブリッジ1サブネットから使用可能なIPv4アドレスの1つを設定します。
ステップ 9	exit 例： <pre>switch(config-app-hosting-app-vnic)# exit</pre>	アプリケーションvnicインターフェイス構成モードを終了して、アプリケーションホスティング構成モードに戻ります。
ステップ 10	app-default-gateway ip-address guest-interface guest-interface 例：	ブリッジ1サブネットから使用可能なIPv4アドレスを構成します。 ステップ3で説明されたようにゲートウェイアドレスを構成します。

■ アプリケーションのライフサイクル

	コマンドまたはアクション	目的
	switch(config-app-hosting-appid)# app-default-gateway 172.25.44.1 guest-interface 0	
ステップ 11	end 例： switch(config-app-hosting)# end	アプリケーション ホスティング構成モードを終了し、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 コマンドは、アプリケーションをアップグレードする方法を示しています。



(注) アプリケーションのインストール後に構成の変更を行った場合、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。アプリケーションの起動後に変更を加えるには、変更を行う前にアプリケーションを停止して非アクティブ化し、アプリケーションを再度アクティブ化して起動します。

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

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： switch# enable	特権 EXEC モードを有効にします。 • 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	app-hosting install appid application-name package package-path 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
	switch# app-hosting install appid te_app package bootflash:my_te_app.tar	• ローカルの保管場所（つまり、ブートフラッシュ）からアプリケーションをインストールすることができます。
ステップ3	app-hosting activate appid application-name 例： switch# app-hosting activate appid te_app	アプリケーションをアクティビ化します。 • このコマンドは、すべてのアプリケーションリソース要求を検証します。すべてのリソースが使用可能な場合、コマンドはアプリケーションをアクティビにします。それ以外の場合、アクティビ化は失敗します。
ステップ4	app-hosting start appid application-name 例： switch# app-hosting start appid te_app	アプリケーションを起動します。 • アプリケーションの起動スクリプトをアクティビにします。
ステップ5	app-hosting stop appid application-name 例： switch# app-hosting stop appid te_app	(任意) アプリケーションを停止します。
ステップ6	app-hosting deactivate appid application-name 例： switch# app-hosting deactivate appid te_app	(任意) アプリケーションに割り当てられているすべてのリソースを無効にします。
ステップ7	app-hosting uninstall appid application-name 例： switch# app-hosting uninstall appid te_app	(任意) アプリケーションをアンインストールします。 • 保存されているすべてのパッケージとイメージをアンインストールします。また、アプリケーションに加えられたすべての変更とアップグレードを削除します。

アプリケーションのアップグレード

次の EXEC コマンドは、アプリケーションをアップグレードする方法を示しています。

手順の概要

1. **enable**
2. **switch# app-hosting upgrade appid application-name package package-path**

Docker ランタイムオプションの設定

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： switch# enable	特権 EXEC モードを有効にします。 • 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	switch# app-hosting upgrade appid application-name package package-path 例： switch# app-hosting upgrade appid tea package bootflash:thousandeys-enterprise-agent-4.1.0.cisco.tar	既存のアプリケーションを新しいバージョンにアップグレードします。その間、このコマンドはアプリケーションを停止し、アップグレードし、新しいアプリケーションイメージでアップグレード前の状態に戻します。 (注) <ul style="list-style-type: none">• STOPPED 状態のアプリケーションをアップグレードした場合、アップグレードが正常に完了すると、新しいapp-hosting 状態は ACTIVATED に変わります。• ローカルの保管場所（つまり、ブートフラッシュ）からアプリケーションをアップグレードすることができます。

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。



(注) 実行時オプションを変更するには、アプリケーションが非アクティブ状態になっている必要があります。

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

手順

	コマンドまたはアクション	目的
ステップ1	enable 例： switch# enable	特権 EXEC モードを有効にします。 • 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： switch# 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： switch(config)# app-hosting appid te_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： switch(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： switch(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： switch(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。

管理インターフェイスでのアプリケーションホスティングの構成

NX-OS を使用すると、アプリケーションコンテナは Cisco NX-OS 管理インターフェイスを介してネットワーク接続を共有できます。仮想 NAT ブリッジを内部的に設定し、ゲスト vNIC インターフェイスにプライベート IP アドレスを割り当てることができます。ゲストインターフェイスのプライベート IP アドレスは、Apphosting フレームワークによって自動的に割り当てられます。

手順の概要

1. **enable**
2. **configure terminal**
3. **app-hosting appid name**
4. **app-vnic management guest-interface network-interface**

5. end

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ1	enable 例： switch# enable	特権 EXEC モードを有効にします。 • 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： switch# configure terminal	グローバル構成モードを開始します。
ステップ3	app-hosting appid name 例： switch(config)# app-hosting appid te_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-vnic management guest-interface network-interface 例： switch(config-app-hosting)# app-vnic management guest-interface 0	ゲストインターフェイスを管理ポートに接続し、アプリケーションホスティング管理ゲートウェイコンフィギュレーションモードを開始します。 • management キーワードは、プライベートIPNATモードでコンテナに接続するCisco NX-OSインターフェイス <i>mgmt0</i> を指定します。 • guest-interface <i>network-interface</i> のキーワード引数ペアは、Cisco NX-OS 管理インターフェイス <i>mgmt0</i> 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0インターフェイスに対して <i>guest-interface 0</i> を使用しています。
ステップ5	end 例： switch(config-app-hosting-mgmt-gateway)# end	アプリケーションホスティング管理ゲートウェイコンフィギュレーションモードを終了し、特権EXECモードに戻ります。

アプリケーションの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。

これらのコマンドを使用して、リソースと app-hosting appid 構成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. **end**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ1	enable 例： switch# enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none"> • 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： switch# configure terminal	グローバル構成モードを開始します。
ステップ3	app-hosting appid name 例： switch(config)# app-hosting appid te_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： switch(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。 <ul style="list-style-type: none"> • 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：	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。

■ 高度なアプリケーションホスティング機能

	コマンドまたはアクション	目的
	switch(config-app-resource-profile-custom)# cpu 7400	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： switch(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	end 例： switch(config-app-resource-profile-custom)# end	カスタム アプリケーション リソース プロファイル構成モードを終了し、特権 EXEC モードに戻ります。

高度なアプリケーションホスティング機能

デフォルトでは、アプリケーションホスティング機能は、シスコがサポートする署名付きアプリケーションパッケージのみを許可します。シスコ以外の署名付きアプリケーション Docker イメージをインストールするには、署名検証機能をオフにする必要があります。これはグローバル設定であり、インストールされているすべてのアプリケーションに影響します。**app-hosting signed-verification [disable | enable]** コマンドは、署名検証を無効にし、シスコ以外の Docker アプリケーションのインストールを支援します。

アプリケーションホスティング機能が設定されている場合、ポートフラッシュから 2 GB のファイルスペースがアプリケーションストレージスペースとして予約されます。特定のアプリケーションにさらに多くの領域が必要な場合は、パーティションサイズを増やすことができます。または、**app-hosting bootflash backend storage limit size** グローバル構成コマンドを使用して、アプリケーションのスペース要件に基づき、スペースを減らすこともできます。アプリケーションは再起動します。

手順の概要

1. **enable**
2. **configure terminal**
3. **app-hosting signed-verification [disable | enable]**
4. **app-hosting bootflash backend storage limit size**
5. **end**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ1	enable 例： switch# enable	特権 EXEC モードを有効にします。 • 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： switch# 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 signed-verification [disable enable] 例： switch(config)# app-hosting signed-verification disable	パッケージ検証を無効にして、シスコ以外のアプリケーションを許可します。 • 署名による検証はデフォルトで有効になっていません。
ステップ4	app-hosting bootflash backend storage limit size 例： switch(config)# app-hosting bootflash backend storage limit 600	インストールするすべてのアプリケーションを考慮して、必要なアプリケーションストレージサイズを設定します。 • デフォルトでは 2048 MB が使用されます。 • サイズは MB 単位で指定します。ブートフラッシュの使用可能な空き領域よりも小さくする必要があります。
ステップ5	end 例： switch(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構成モードを終了し、特権 EXEC モードに戻ります。

アプリケーションデータのコピー

アプリケーションの永続データマウントからアプリケーションデータを削除するには、特権 EXEC モードで**app-hosting data appid <appid> copy**コマンドを使用します。

app-hosting data appid tea copy bootflash:src dest

値は次のとおりです。

`src` はブートフラッシュからの送信元ファイルで、`dest` は接続先ファイルパスです。

アプリケーションデータの削除

アプリケーションの永続データマウントからアプリケーションデータを削除するには、特権 EXEC モードで **app-hosting data appid <appid> delete** コマンドを使用します。

```
app-hosting data appid tea delete file
```

値は次のとおりです。

file は、アプリケーションの永続データマウントから削除されるファイルです。

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。任意の順序でこれらのコマンドを使用できます。

手順の概要

1. **show app-hosting infra**
2. **show app-hosting list**
3. **show app-hosting bridge**
4. **show app-hosting detail**
5. **show app-hosting resource**
6. **show app-hosting app-hosting utilization appid <app-name>**
7. **show-tech app-hosting**

手順の詳細

手順

ステップ1 **show app-hosting infra**

アプリケーションホスティングインフラの概要を表示します。

(注)

さらに操作を実行する前に、CAF を動作状態に移行します。

例：

```
switch(config)# show app-hosting infra
App signature verification: disabled
Docker partition size: 0 MB
Inband packet rate limit: 0 PPS
Services
-----
CAF 1.16.0.0 : Running
HA : Running
App Manager : Running
Libvirtd 4.7.0 : Running
```

```
Dockerd 18.09.0-ce : Running  
Linux kernel 5.10.126 : Running
```

ステップ2 show app-hosting list

動作しているアプリのリストを表示します。

例：

```
switch(config)# show app-hosting list  
App id          Statenginx_1         started
```

ステップ3 show app-hosting bridge

アプリケーションホスティング ブリッジのリストを表示します。

例：

```
switch(config)# show app-hosting bridge  
Bridge ID    VRF      IP Address      IPv6 Address1           blue     172.10.23.45/24  ::/0
```

ステップ4 show app-hosting detail

プリケーションホスティングについての詳細情報を表示します。

例：

```
switch(config)# show app-hosting detail  
App id : nginx_1  
Owner : appmgr  
State : started  
Application  
Type : docker  
Name : nginx  
Version : latest  
Description :  
Author :  
Path : /bootflash/nginx.tar.gz  
URL Path :  
Activated profile name : default  
  
Resource reservation  
Memory : 64 MB  
Disk : 10 MB  
CPU : 200 units  
  
Platform resource profiles  
Profile Name CPU(unit) Memory(MB) Disk(MB)  
-----  
  
Attached devices  
Name          Type          Alias  
-----  
iox_trace     serial/trace  serial3  
iox_syslog    serial/syslog serial2  
iox_console_aux serial/aux   serial1  
iox_console_shell serial/shell serial0  
  
Network interfaces  
-----  
eth0:
```

■ アプリケーションホスティング設定の確認

```

MAC address : 5254.9999.0000
IPv4 address : 192.168.10.130
IPv6 address : fe80::5054:99ff:fe99:0/64
Network name : iox-nat_docker0
Tx Packets : 9
Tx Bytes : 726
Tx Errors : 0
Rx Packets : 0
Rx Bytes : 0
Rx Errors : 0

Docker
-----
Run-time information
Command :
Entry-point : /docker-entrypoint.sh nginx -g 'daemon off;'
Run options in use : --publish=40080:80
Package run options :
Application health information
Status : 0
Last probe error :
Last probe output :

```

ステップ5 show app-hosting resource

アプリケーションホスティングのリソースに関する情報を表示します。

例 :

```

switch(config)# show app-hosting resource
CPU:
Total: 7400 units
Available: 7200 units
VCPU:
Application Hosting
Additional References
Application Hosting
46
Count: 1
Memory:
Total: 3840 (MB)
Available: 3776 (MB)
Storage space:
Total: 110745 (MB)
Available: 93273 (MB)
vice

```

ステップ6 show app-hosting app-hosting utilization appid <app-name>

アプリケーションの使用率を表示します。

例 :

```

switch(config)# show app-hosting utilization appid nginx_1
Application: nginx_1
CPU Utilization:
CPU Allocation: 200 units
CPU Used: 0 %
Memory Utilization:
Memory Allocation: 64 MB
Memory Used: 7000 KB
Disk Utilization:
Disk Allocation: 10 MB
Disk Used: 0 MB

```

ステップ7 show-tech app-hosting

すべてのアプリケーションホスティング ログと、関連する依存コンポーネント ログを表示します。

この show-tech コマンドは、次の show コマンドの詳細を収集します。

例：

```
show system internal app-hosting
show system internal app-hosting event-history debug
show system internal app-hosting event-history error
show system internal app-hosting event-history msgs
show app-hosting list
show app-hosting detail
show app-hosting utilization
show app-hosting infra
show app-hosting resource
show app-hosting bridge
show routing appmgr vrf all
show routing ipv6 appmgr vrf all
```

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：AppHosting 機能の有効化

次の例は、Cisco Apphosting 機能を有効にする方法を示しています。

```
switch# configure terminal
switch(config)# feature app-hosting
switch(config)# end
```

例：アプリケーションホスティング ブリッジ接続の構成

この例は、アプリケーションホスティング ブリッジ接続を構成する方法を示しています。

```
switch(config)# configure terminal
switch(config)# app-hosting bridge 1
switch(config-app-hosting-bridge)# ip address 172.25.44.1/30
switch(config-app-hosting-bridge)# vrf member overlay-VRF
switch(config-app-hosting-bridge)# exit
switch(config)# app-hosting appid te_app
switch(config-app-hosting)# app-vnic bridge 1 guest-interface 0
switch(config-app-hosting-app-vnic)# guest-ipaddress 172.25.44.2/30
switch(config-app-hosting-app-vnic)# exit
switch(config-app-hosting-appid)# app-default-gateway 172.25.44.1 guest-interface 0
switch(config-app-hosting)# end
```

■ 例：Docker ランタイムオプションの設定

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。

```
switch> enable
switch# configure terminal
switch(config)# app-hosting appid te_app
switch(config-app-hosting)# app-resource docker
switch(config-app-hosting-docker)# run-opts 1 "-v ${APP_DATA}:/data"
switch(config-app-hosting-docker)# end
```

例：管理インターフェイスでのアプリケーションホスティングの構成

次に、管理インターフェイスでアプリケーションホスティングを構成する例を示します。

```
switch> enable
switch# configure terminal
switch(config)# app-hosting appid te_app
switch(config-app-hosting)# app-vnic management guest-interface 0
switch(config-app-hosting)# 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。

```
switch> enable
switch# configure terminal
switch(config)# app-hosting appid te_app
switch(config-app-hosting)# app-resource profile custom
switch(config-app-resource-profile-custom)# cpu 7400
switch(config-app-resource-profile-custom)# memory 2048
switch(config-app-resource-profile-custom)# 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
アプリケーション ホスティングの構成	Cisco Nexus 3000 and 9000 Series NX-API REST SDK User Guide and API Reference (Cisco Nexus 3000 および 9000 シリーズ NX-API REST SDK ユーザ ガイドと API リファレンス)

テクニカル サポート

説明	リンク
<p>Cisco のサポート Web サイトでは、Cisco 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

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。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 にアクセスしてください。Cisco.com のアカウントは必要ありません。

■ アプリケーションホスティングに関する機能情報

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
Cisco アプリケーション ホスティング フレームワーク (CAF)	Cisco NX-OS リリース 10.3(3)F	<p>ホスト対象のアプリケーションは Software as a Service (SaaS) ソリューションであり、ユーザー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、Cisco 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>アプリケーション ホスティング機能は、以下の PID で Cisco Nexus 3600 シリーズでサポートされています。</p> <ul style="list-style-type: none"> • N3K-C36180YC-R • N3K-C3636C-R • N3K-C36480LD-R2

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。