



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 \(1 ページ\)](#)
- [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 \(22 ページ\)](#)
- [アプリケーションホスティング設定の確認 \(40 ページ\)](#)
- [アプリケーションホスティングの設定例 \(43 ページ\)](#)
- [その他の参考資料 \(51 ページ\)](#)
- [アプリケーションホスティングに関する機能情報 \(52 ページ\)](#)

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 インターフェイスはスタック内のアクティブスイッチでのみ動作します。したがって、AppGigabitEthernet インターフェイスの設定は、スタック内のすべてのスイッチの AppGigabitEthernet インターフェイスに適用する必要があります。この設定が適用されていないスイッチがある場合、スイッチオーバー後にそのスイッチでは AppGigabitEthernet インターフェイスが機能しません。

Cisco Catalyst 9000 シリーズスイッチは、アプリケーションが SSD でホストされている場合、複数のアプリケーションをサポートします。アプリケーションは、次の条件を満たす必要があります。

- シスコの署名がある。
- 次のスイッチングインフラストラクチャ要件を満たしている。
 - AppGigabitEthernet ポート上のネットワーク設定で、アプリケーション間の競合が発生しない。
 - アプリケーションを実行するのに十分なリソースがある。

1つのアプリケーションが使用可能なすべてのアプリケーションホスティングリソースを消費する場合、複数のアプリケーションを導入することはできません。たとえば、1つのアプリケーションがすべてのコンピューティングリソースとランタイムリソースを消費している場合、他のアプリケーションはデバイスにインストールできなくなります。

前面パネルトランクおよび VLAN ポートのアプリケーションホスティング

アプリケーションホスティングでは前面パネル VLAN ポートおよびトランクポートがサポートされています。レイヤ2トラフィックは、これらのポートを介して、Cisco IOS デーモンの外部で動作するソフトウェアコンポーネントに配信されます。

アプリケーションホスティングの場合、前面パネルポートをトランクインターフェイスまたは VLAN 固有のインターフェイスとして設定できます。トランクインターフェイスとして使用する場合、前面パネルポートはレイヤ2トランクポートとして機能するように拡張され、ポートで受信したすべてのトラフィックがアプリケーションで使用可能になります。ポートを VLAN インターフェイスとして使用する場合、アプリケーションは特定の VLAN ネットワークに接続されます。



- (注) 背面パネルの USB または M2 SATA ドライブをアプリケーション ホスティングに使用する場
合、ストレージメディアは *ext4* ファイルシステムとしてフォーマットする必要があります。

Cisco Catalyst 9300 シリーズ スイッチ のアプリケーション ホスティング

ここでは、Cisco Catalyst 9300 シリーズ スイッチでのアプリケーション ホスティングについて説明します。

アプリケーション ホスティングの場合、Cisco Catalyst 9300 シリーズ スイッチは管理インターフェイスと前面パネルポートをサポートします。

USB 3.0 SSD は Cisco Catalyst 9300 シリーズ スイッチで有効になっています。USB 3.0 SSD は、アプリケーションをホストするための追加の 120 GB ストレージを提供します。詳細については、『*Interfaces and Hardware Configuration Guide*』の「Configuring USB 3.0 SSD」の章を参照してください。

次の 2 種類のネットワークング アプリケーションがサポートされています。

- コントロールプレーン：管理インターフェイスにアクセスするアプリケーション。
- データプレーン：前面パネルのポートにアクセスするアプリケーション。

Cisco Catalyst 9300X シリーズ スイッチの前面パネル アプリケーション ホスティング

Cisco IOS XE Bengaluru 17.6.1 では、前面パネルのアプリケーション ホスティングが Cisco Catalyst 9300X シリーズ スイッチで有効になっています。

アプリケーションは、ホスティングに専用の前面パネルポートを使用できます。**app-vnic AppGigabitEthernet port** コマンドを使用して、アプリケーションホスティングに使用するポートを指定します。両方の前面パネルポートを同じレイヤ 2 アプリケーションに接続できます。

これらのスイッチは、アクセスモードとトランクモードの両方でアプリケーションホスティングをサポートします。アプリケーションホスティングは、両方のモードで同時に有効にできません。



- (注) **app-vnic** コマンドで行われた設定は、アクティベーション中に拒否される可能性があります。

表 1: アクセスモードとトランクモードでのアプリケーションホスティングの設定シナリオ例

シナリオ	サポート対象/非サポート対象
単一のアプリケーションとアクセスモードの2つの前面パネルポート。	サポート。 重複する VLAN はありません。
単一のアプリケーションとトランクモードの2つの前面パネル。	サポート。 重複する VLAN はありません。
単一のアプリケーションとトランクモードおよびアクセスモードの2つの前面パネルポート。	サポート。 重複する VLAN はありません。
単一のアプリケーションと、デフォルトのアプリケーションゲートウェイが設定されたトランクモードの2つの前面パネルポート。	サポート。 同一のアプリケーションと2つのインターフェイスが異なるサブネットに設定されていますが、デフォルトゲートウェイは、外部接続を持つ1つの VLAN に接続されます。
単一のアプリケーションと、VLAN が重複しているトランクモードとアクセスモードの2つの前面パネル。	有効な設定ではありません。 VLAN が両方のポートで重複しています。
アクセスモードでの単一アプリケーションと、同一 VLAN に設定された2つの前面パネルポート。	有効な設定ではありません。
トランクモードでの単一アプリケーションと、重複する VLAN 範囲で設定された2つの前面パネルポート。	有効な設定ではありません。 トラフィックが分離されていおらず、VLAN 範囲が重複しています。
トランクモードでの単一アプリケーションと、重複する VLAN 範囲で設定された2つの前面パネルポート。	有効な設定ではありません。 この設定はアクティベーション中に拒否されます。 両方の前面パネルポートがトランクモードであるため、任意の VLAN を使用できます。ただし、両方のポートに同じ VLAN が設定されているため、VLAN は両方のポートで重複します。 (注) 同じシナリオがアクセスモードにも適用されます。

シナリオ	サポート対象/非サポート対象
トランクモードとアクセスモードでの単一のアプリケーション、および VLAN が重複している前面パネルポート。	有効な設定ではありません。 トランクモードとアクセスモードで同じ VLAN が設定されています。この設定により、VLAN は両方のポートで重複しています。
トランクモードでの複数のアプリケーション。	有効な設定ではありません。 トラフィックが分離されません。
トランクモードとアクセスモードでの 2 つのアプリケーション。	有効な設定ではありません。 VLAN が重複しています。

Cisco Catalyst 9300X シリーズ スイッチのハイアベイラビリティ

Cisco Catalyst 9300X シリーズ スイッチで使用可能な混合モードスタックでは、アクティブデバイスとスタンバイデバイスが、アプリケーションホスティング用に 1+1 冗長性を使用します。混合モードのサポートとは、異なるモデルバリエーションと異なるネットワークモジュールがスタックで使用される場合です。

Cisco Catalyst 9300X シリーズ スイッチと Cisco Catalyst 9300 シリーズ スイッチをスタックすると、Cisco Catalyst 9300X シリーズ スイッチの 2 つの前面パネルポートのいずれか 1 つが動的に無効になります。AppGigabitEthernet 1/0/1 インターフェイスのみが有効として表示されます。

このセクションでは、いくつかのハイアベイラビリティのシナリオについて説明します。

スタックモード	機能	使用されるポート	動作
Cisco Catalyst 9300X シリーズ スイッチ (アクティブ) + Cisco Catalyst 9300X シリーズ スイッチ (スタンバイ)	2	1	サポート対象
Cisco Catalyst 9300X シリーズ スイッチ (アクティブ) + Cisco Catalyst 9300X シリーズ スイッチ (スタンバイ)	1	1	サポート対象

スタックモード	機能	使用されるポート	動作
Cisco Catalyst 9300X シリーズスイッチ (アクティブ) + Cisco Catalyst 9300 シリーズスイッチ (スタンバイ)	1	1 ポート 1 を使用した場合のみ。	サポート。 この設定は、 app-vnic Appgigabitethernet port 1 trunk コマンドまたは app-vnic AppgigabitEthernet trunk コマンドを使用してポート 1 が設定されている場合にサポートされます。 ポート番号が指定されていない場合は、スイッチオーバーが発生したときにデフォルトのポート 1 が使用されます。
Cisco Catalyst 9300X シリーズスイッチ (アクティブ) + Cisco Catalyst 9300 シリーズスイッチ (スタンバイ)	1	2	未サポート このシナリオでは、スイッチオーバーが発生すると、新しいアクティブには前面パネルポートが 2 つないため、アプリケーションの設定が失敗します。 スイッチオーバー後、Cisco Catalyst 9300 シリーズスイッチではアプリケーションが再起動されません。これは、前面パネルのポートが 1 つだけ設定されており、この設定が失敗するためです。使用可能な前面パネルポートを使用してアプリケーションを再設定する必要があります。

スタックモード	機能	使用されるポート	動作
Cisco Catalyst 9300X シリーズ スイッチ (アクティブ) + Cisco Catalyst 9300 シリーズ スイッチ (スタンバイ)	2	2	未サポート (注) たとえば、app1 と app2 の 2 つのアプリケーションが実行されており、各アプリケーションがそれぞれ異なる前面パネルポート (port1 と port2 など) を使用しているとします。 スイッチオーバー後、前面パネル port1 上の app1 が実行状態の Cisco Catalyst 9300 シリーズ スイッチで開始されます。ただし app2 は、前面パネルに port2 がないため、Cisco Catalyst 9300 シリーズ スイッチでは開始されません。
Catalyst 9300 シリーズ スイッチ (アクティブ) + Catalyst 9300X シリーズ スイッチ (スタンバイ)	1 つ以上	1	サポート。 (注) スイッチオーバー後、アプリケーションは Catalyst 9300X シリーズ スイッチで前面パネルのポートを使用して再開始されます。

Cisco Catalyst 9400 シリーズ スイッチでのアプリケーションホスティング

ここでは、Cisco Catalyst 9400 シリーズ スイッチでのアプリケーションホスティングについて説明します。

アプリケーションホスティングの場合、Cisco Catalyst 9400 シリーズスイッチは管理インターフェイスと前面パネルポートをサポートします。アプリケーションは、C9400-SSD-240GB、C9400-SSD-480GB、および C9400-SSD-960GB ソリッドステートドライブ (SSD) でホストできます。

これらのスイッチは、アプリケーションホスティングに M2 SATA モジュールを使用します。詳細については、『*Interfaces and Hardware Configuration Guide*』の「M2 SATA Module」の章を参照してください。

Cisco Catalyst 9400 シリーズスイッチでは、アプリケーションはアクティブなスーパーバイザでのみホストできます。スイッチオーバー後、新しくアクティブになったスーパーバイザの 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 9410 シリーズスイッチでのアプリケーションホスティング

Cisco IOS XE Bengaluru 17.5.1 では、アプリケーションホスティングが Cisco Catalyst 9410 シリーズスイッチでサポートされています。アプリケーションホスティング用に AppGigabitEthernet インターフェイスを有効にするには、インターフェイス コンフィギュレーションモードで **enable** コマンドを設定します。



(注) **enable** コマンドは、Cisco Catalyst 9410 シリーズスイッチでのみ使用できます。

スロット 4 の 48 ポートラインカードをアプリケーションホスティングに使用する場合、そのポートはデフォルトのシャットダウンモードである必要があります。スロット 4 の 48 ポートラインカードがアクティブな場合、アプリケーションホスティングは拒否されます。ラインカードポートが無効な場合、スロット 4 の 48 ポートラインカードが非アクティブとしてマークされます。

スロット 4 に 48 ポートラインカードが装着されている場合、ポート 4/0/48 はアップ状態になりません。ラインカード 4 が空の場合、または 24 ポートラインカードの場合、無効になるポートはありません。

ポート (4/0/48) を有効にするには、**no iox** コマンドを使用してアプリケーションホスティングを無効にします。ポートが有効または無効の場合、コンソールにシステムメッセージは表示されません。

インサービスソフトウェアアップグレード (ISSU) の実行中は、AppGigabitEthernet インターフェイスを有効にする必要があるため、ラインカードポートは自動的に無効になりません。ソフトウェアのダウングレードの前に、AppGigabitEthernet インターフェイスを無効にして、前面パネルポートを無効にする必要があります。

ホットスワップ (OIR)

表 2: 活性挿抜 (OIR) のシナリオ

OIR のシナリオ	アクション
スロット 4 のラインカードが空で、AppGigabitEthernet インターフェイスが有効になっている。	無効なポートはありません。
スロット 4 のラインカードが 48 ポートラインカードで、AppGigabitEthernet インターフェイスが有効になっている。	スロット 4 のポート 48 は無効です。ポートが無効になった後は、ポートに設定は適用されません。ポート 48 は非アクティブとしてマークされます。
スロット 4 のラインカードは 24 ポートラインカードである。	スロット 4 のポートは無効になりません。
スロット 4 のラインカードが 48 ポートのラインカードで、それが 24 ポートのラインカードに置き換えられ、AppGigabitEthernet インターフェイスが有効化された。	スロット 4 のポートは無効になりません。
スロット 4 のラインカードが 24 ポートのラインカードであり、48 ポートのラインカードに置き換えられ、AppGigabitEthernet インターフェイスが有効になっている。	スロット 4 のポート 48 は無効です。
OIR の操作中にスタンバイスーパーバイザが新しいアクティブになり、新しいアクティブの前面パネルポートがアプリケーションホスティングに使用される。	スロット 4 のポート 48 の状態は変化しません。スタンバイスーパーバイザの OIR は、アクティブスーパーバイザの前面パネルポートには影響しません。

Cisco StackWise Virtual

ここでは、デュアルスーパーバイザのアップリンクポートを StackWise Virtual リンクとして使用する場合のシナリオについて説明します。

- アプリケーションホスティングが有効で、ラインカード 4 のポート 48 がアップ状態ではない場合、アクティブシャーシとスタンバイシャーシの両方で無効になります。
- アクティブまたはスタンバイシャーシのラインカード 4 のポート 48 でリンクがアップ状態である場合、**enable** コマンドは拒否されます。
- ラインカード 4 のポート 48 をデュアルアクティブ検出 (DAD) リンクとして使用する場合は、DAD リンクを削除し、これを別のポートで設定します。
- ラインカード 4 のポート 48 を StackWise Virtual リンクとして使用し、前面パネルポートを有効にする必要がある場合は、ポート 48 の StackWise Virtual リンクを削除し、別のポ

トを StackWise Virtual リンクとして使用します。ラインカード4のポート48は、StackWise Virtual または DAD リンクとして使用できません。

Cisco Catalyst 9500 シリーズスイッチでのアプリケーションホスティング

Cisco Catalyst 9500-High Performance シリーズスイッチは、M2 SATA モジュール、SSD-240G、SSD-480G、および SSD-960 (C9k-F1-SSD-240GB) のみをサポートします。前面パネルの USB はサポートされていません。

詳細については、『Cisco IOS XE Amsterdam 17.2.x (Catalyst 9500 スイッチ) インターフェイスおよびハードウェアコンポーネントコンフィギュレーションガイド』の「M2 SATA モジュール」を参照してください。

Cisco IOS XE Cupertino 17.7.1 では、Cisco Catalyst 9500X シリーズスイッチは、AppGigabitEthernet インターフェイスでのアプリケーションホスティングをサポートしています。アプリケーションホスティングは、次の M2 SATA モジュールでサポートされています。SSD-240G、SSD-480G、および SSD-960 (C9k-F1-SSD-240GB)。

Cisco Catalyst 9600 シリーズスイッチでのアプリケーションホスティング

Cisco Catalyst 9600 シリーズスイッチは、アプリケーションホスティングのために M2 SATA モジュールのみをサポートします。前面パネルの USB はサポートされていません。次の M2 SATA モジュール (SSD-240G、SSD-480G、および SSD-960 (C9k-F2-SSD-240GB)) がサポートされています。

詳細については、『Cisco IOS XE Amsterdam 17.2.x (Catalyst 9600 スイッチ) インターフェイスおよびハードウェアコンポーネントコンフィギュレーションガイド』の「M2 SATA モジュール」を参照してください。

内部フラッシュから SSD へのアプリケーションの自動転送および自動インストール

IOx が有効である場合、使用可能な最適なメディアが選択され、そのメディアを使用して IOx サービスが開始されます。IOx は、その起動時にアプリケーションを実行するメディアも選択します。

IOx が再起動して別のメディアが選択された場合は、すべてのアプリケーション (Docker アプリケーションのみサポートされます) を新しいメディアに移行し、コンテナを変更前と同じ状態に復元する必要があります。アプリケーションに関連付けられているすべての永続データとボリュームも移行する必要があります。

再起動中、IOx は次の優先順位でメディアを選択します。

1. ハードディスク
2. フラッシュ

フラッシュはゲストシェルのみをサポートします。他のアプリケーションは許可されません。

ユースケース

このセクションでは、アプリケーションの自動転送および自動インストール中のいくつかのユースケースについて説明します。

表 3: アプリケーションの自動転送および自動インストールのユースケース

使用例	結果
IOx がフラッシュで実行している間に SSD が接続される。	IOx の実行中に SSD が接続される場合、実行中のアプリケーションまたは IOx に影響はありません。IOx が SSD に移行されるのは、IOx を無効にして再起動し、CLI を介して有効にした場合、またはシステムの再起動の場合のみです。
IOx データが新しいメディアにコピーされている間に、システムがリブートします。	あるメディアから別のメディアへの IOx データの移行中にシステムが再起動した場合は、システムの再起動時に移行プロセスが継続されます。古いメディアのデータは、コピー操作が完了したときにのみ削除されます。

ThousandEyes Enterprise Agent の概要

ThousandEyes Enterprise Agent は、エンタープライズ ネットワーク監視ツールであり、ビジネスに影響を与えるネットワークとサービス全体のエンドツーエンドのビューを提供します。内部、外部、キャリア、およびインターネットネットワーク全体のネットワークトラフィックパスをリアルタイムでモニタして、ネットワークパフォーマンスデータを提供します。Enterprise Agent は、WAN やインターネットの接続状態を詳細に把握するために、ブランチサイトやデータセンターにインストールするのが最も一般的です。

以前の Cisco IOS XE リリースでは、ThousandEyes は SSD 上のサードパーティ製カーネルベース仮想マシン (KVM) アプライアンスとしてサポートされていました。

Cisco IOS XE Amsterdam 17.3.3 では、ThousandEyes Enterprise Agent の新しいバージョンであるバージョン 3.0 が導入されました。これは、アプリケーションホスティング機能を使用してシスコデバイスで実行される組み込み型の Docker ベースアプリケーションです。Enterprise Agent は SSD とブートフラッシュの両方で使用でき、ブラウザテスト (ページロードとトランザクション) を除くすべてのテストをサポートします。ブラウザテストは、Cisco IOS XE Bengaluru 17.6.1 以降のリリースの Enterprise Agent バージョン 4.0 で使用できます。

ThousandEyes Enterprise Agent は次の機能を提供します。

- ネットワークとアプリケーションのパフォーマンスベンチマーク。
- 詳細なホップバイホップメトリック。
- ブランチまたはキャンパスからデータセンターまたはクラウドへのエンドツーエンドのパスの可視化。
- 機能停止の検出と解決。
- ユーザーエクスペリエンス分析。
- トラフィックフローパターンの可視化。

Cisco IOS XE Bengaluru 17.6.1 で利用可能な ThousandEyes Enterprise Agent バージョン 4.0 は、ThousandEyes Agent バージョン 3.0 では利用できない次の追加機能をサポートしています。

- BrowserBot のサポート（背面パネル SSD が使用可能な場合）。
- DNAC アプリケーションのアイコンおよび説明
- Docker ヘルスモニタリング。
- ThousandEyes Enterprise Agent をアップグレードするための **app-hosting upgrade URL** コマンド。

ThousandEyes Enterprise Agent の前提条件

- ThousandEyes サイトで入手可能な ThousandEyes Enterprise Agent イメージは、HTTPS ダウンロード用に www.cisco.com で使用される認証局（CA）と同じ認証局によって署名される必要があります。ユーザー名とパスワードは使用されません。
- Enterprise Agent をインストールするには、インターネット接続またはプロキシサーバーが必要です。詳細については、<https://docs.thousandeyes.com/product-documentation/enterprise-agents> にある ThousandEyes のドキュメントを参照してください。
- Enterprise Agent アプリケーションは、ユーザーのライセンス権限が検証された後にのみ使用できます。
- Docker ベースのアプリケーションのみがサポートされます。
- 1:1 スタックモードは、ThousandEyes ステートフル スイッチオーバー（SSO）をサポートするための必須条件です。

1:1 モードとは、スタック内の特定のデバイスにアクティブラ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 4: ThousandEyes Enterprise Agent に必要なリソース

アプリケーションメディア	最大リソース	サポートされるリリース
SSD (注) 120G SSD のみがサポートされます。	<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 Cisco IOS XE Bengaluru 17.5.1 <ul style="list-style-type: none"> • Cisco Catalyst 9400 シリーズ スイッチ Cisco IOS XE Bengaluru 17.6.1 <ul style="list-style-type: none"> • Cisco Catalyst 9300X シリーズ スイッチ
フラッシュ	<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せん。 	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 Cisco IOS XE Bengaluru 17.5.1 <ul style="list-style-type: none"> • Cisco Catalyst 9400 シリーズ スイッチ Cisco IOS XE Bengaluru 17.6.1 <ul style="list-style-type: none"> • Cisco Catalyst 9300X シリーズ スイッチ

Cisco IOS XE Bengaluru 17.6.1では、アドオンモードはCisco Catalyst 9300、9300L、および9300Xシリーズスイッチ、およびCisco Catalyst 9400シリーズスイッチでサポートされています。

ThousandEyes Enterprise Agent のダウンロード

ThousandEyes Enterprise Agentには、ブラウンフィールドとグリーンフィールドの2つのタイプがあります。既存のデバイスの場合は、ThousandEyes Webサイトからブラウンフィールドバージョンをダウンロードできます。一方、新しいデバイスは、グリーンフィールドアプリケーションがブートフラッシュにロードされた状態で出荷されます。

次の表に、エージェントで使用可能なダウンロードオプションを示します。

表 5: ThousandEyes Enterprise Agent のダウンロードオプション

ブラウフィールド	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。

ThousandEyes BrowserBot

ThousandEyes Enterprise Agent バージョン 4.0 には、トランザクション スクリプト テスト用の BrowserBot が用意されています。BrowserBot は、ページロードおよびトランザクションテストを管理する Enterprise Agent のコンポーネントです。BrowserBot を使用すると、ThousandEyes クラウドポータルでの Web ブラウザのアクションを模倣するカスタマイズされた JavaScript テストを有効にできます。ホストオペレーティングシステムを誤った JavaScript 操作から保護するために、ThousandEyes Agent は JavaScript を実行するサンドボックスコンテナを作成します。

制限がないディスクがアプリケーションで使用される場合、ThousandEyes Agent は初期化中に BrowserBot パッケージを動的にインストールします。これにより、ポータル トランザクション スクリプト テストを設定できます。



(注) BrowserBot のサポートは、ThousandEyes Agent バージョン 3.0 では適用されません。

BrowserBot は、大量のハードウェアリソースを消費します。2 GB のシステムメモリと 2 つの VCPU 負荷が、すべての IOx アプリケーションに割り当てられる最大 IOx システムメモリと CPU 負荷です。ブートフラッシュで複数のアプリケーションを同時に実行できるようにするには、エージェントをアクティブ化する前に、デフォルトの `package.yaml` BrowserBot のリソースを削減します。`app-resource profile custom` コマンドを使用して、デフォルトの `package.yaml` 設定を上書きします。

- CPU : 1850 CPU ユニット (1/4 VCPU)
- メモリ : 500MB

トランザクションのスクリプト化の詳細については、次のリンクを参照してください。

- <https://docs.thousandeyes.com/product-documentation/tests/transaction-scripting-guide>
- <https://docs.thousandeyes.com/product-documentation/tests/transaction-scripting-reference>

トランザクションのスクリプト化の例については、<https://github.com/thousandeyes/transaction-scripting-examples> を参照してください。

ThousandEyes Agent のアップグレードとダウングレード

ThousandEyes Agent のアップグレード

Cisco IOS XE Amsterdam 17.3.3 および Bengaluru 17.5.1 で使用可能な ThousandEyes Enterprise Agent 3.0 は、Cisco IOS XE Bengaluru 17.6.1 で使用可能な Agent 3.0 または Agent 4.0 にアップグレードできます。Agent 3.0 は、アップグレード後に操作によって復元されます。

Agent 4.0 は Cisco IOS XE Bengaluru 17.6.1 で使用でき、エージェントは、自動アップグレードにより起動時に最新の Agent 4.0 バイナリに更新されます。現在、Agent 4.0 に対するアップグレードはありません。

アプリケーションのアップグレードは、次の方法で実行できます。

- ThousandEyes エージェントの自動アップグレード : アプリケーションの起動時に自動的に実行されます。実行中のコンテナ内のエージェントバイナリはアップグレードされますが、アプリケーションパッケージはアップグレードされません。
- `app-hosting upgrade` コマンドを使用する。
- DNAC アプリのアップグレード。

ThousandEyes Agent のダウングレード

Cisco IOS XE Amsterdam 17.3.3、Cisco IOS XE Bengaluru 17.5.1、および Cisco IOS XE 17.6.1 で使用可能な Agent 3.0 はダウングレードできません。

Cisco IOS XE Bengaluru 17.6.1 で使用可能な Agent 4.0 は、Cisco IOS XE Bengaluru 17.6.1 で使用可能な Agent 3.0 にダウングレードできます。他のダウングレードはできません。

ダウングレードするときに、アプリケーションが以前のリリースと同じ状態にならない場合は、アプリケーションを非アクティブ化またはアンインストールしてから、インストールまたは再起動します。

ネイティブ Docker コンテナ：アプリケーションの自動再起動

アプリケーションの自動再起動機能を使用すると、プラットフォームに導入されたアプリケーションは、システムのスイッチオーバーまたは再起動時に最後に設定された動作状態を維持できます。基盤となるホスティングフレームワークは、スイッチオーバー中も保持されます。この機能はデフォルトで有効であり、ユーザが無効にすることはできません。

アプリケーションの永続データは同期されません。Cisco Application Framework (CAF) が認識しているセキュアデータストレージと永続データのみが同期されます。

スイッチオーバーまたはシステムの再起動時に IOx を同じ状態で再起動するには、アクティブデバイスとスタンバイデバイスにある IOx メディアが同期している必要があります。

Cisco Catalyst 9300 シリーズスイッチは、アプリケーションホスティングで SSD のみをサポートします。新しい SSD を挿入したら、他の SSD と同じ同期状態にする必要があります。アプリケーションの自動再起動同期を機能させるには、スタンバイデバイスに IOx と互換性のある SSD が必要です。

show iox-service コマンドの出力は同期の状態を表示します。

アプリケーションの自動再起動機能は、Cisco Catalyst 9300 シリーズスイッチでのみサポートされます。

アプリケーションの自動再起動のシナリオ

ここでは、さまざまなアプリケーションの自動再起動のシナリオについて説明します。

表 6: アプリケーションの自動再起動のシナリオ

シナリオ	アクティブデバイスの単一メディア	アクティブデバイスとスタンバイデバイスのメディア
システムブートアップ	システムブートアップ時に IOx とアプリケーションを起動します。USB SSD はローカルデバイスであるため、すぐに表示されます。この時点では同期は行われません。	システムのブートアップ時に IOx とアプリケーションを起動します。既存の情報をスタンバイデバイスに一括同期します。

シナリオ	アクティブデバイスの単一メディア	アクティブデバイスとスタンバイデバイスのメディア
スイッチオーバー	新しいアクティブデバイスでメディアが見つかりません。IOxは、以前にインストールされたアプリケーションがなく、最小限の機能を持つシステムフラッシュで起動します。	システムスイッチオーバー (SSO) 後に、新しいアクティブデバイスでIOxとアプリケーションを以前の状態で起動します。新しいスタンバイデバイスがブートアップした後に、情報の一括同期を実行します。
ブートアップまたはスイッチオーバー：USB SSD がメンバーデバイスに存在します。	メンバーデバイスに存在するSSDの同期はありません。メンバーSSDはIOxおよびアプリケーションのホストには使用されません。	メンバーデバイスに存在するSSDの同期はありません。メンバーSSDはIOxおよびアプリケーションのホストには使用されません。
デバイスの削除：アクティブデバイスからローカルUSB SSDが削除されます。	ローカルUSB SSDが削除されると、IOxがグレースフル終了を処理します。 SSDがアクティブデバイスに差し戻されたら、ユーザがトリガーするIOxの再起動が必要です。	IOxはグレースフル終了を処理します。IOxはローカルディスク上でのみ動作するため、スタンバイSSDはIOxの起動に使用されません。 SSDがアクティブデバイスに差し戻されたら、ユーザがトリガーするIOxの再起動が必要です。
デバイスの削除：USB SSD がスタンバイデバイスから削除されます。	該当なし	IOx同期操作が失敗します。IOxはSSO対応ではなくなりました。
デバイスの削除：リモートUSB SSD がリモートメンバーデバイスから削除されます。	IOxはメンバーSSDを使用しないため、影響はありません。	IOxはメンバーSSDを使用しないため、影響はありません。
デバイスのダウン：IOxが実行されているアクティブなデバイスがダウンします。	新しいアクティブデバイスでメディアが見つかりません。IOxは、以前にインストールされたアプリケーションがなく、最小限の機能を持つ状態でシステムフラッシュで起動します。	新しいアクティブデバイスで、SSO前の状態でIOxとアプリケーションを起動します。新しいスタンバイデバイスがブートアップすると、情報の一括同期を実行します。

シナリオ	アクティブデバイスの単一メディア	アクティブデバイスとスタンバイデバイスのメディア
指定されたアクティブ/スタンバイデバイスの変更 (スタック環境 1:1)	変更はリブート後に反映されます。リブート後、新しいアクティブデバイスから IOx が起動します。	変更はリブート後に反映されます。リブート後、新しいアクティブデバイスから IOx が起動します。

Cisco Catalyst 9300 シリーズ スイッチでのアプリケーション自動再起動

ここでは、マルチメンバースタックの Cisco Catalyst 9300 シリーズ スイッチでアプリケーションの自動再起動がどのように機能するかについて説明します。

Cisco Catalyst 9300 シリーズ スイッチでは、アプリケーションの自動再起動は、スタック内の特定のデバイスにアクティブロールとスタンバイロールを割り当てる、1+1 スイッチ冗長モードまたは StackWise Virtual モードでサポートされます。

スイッチスタックが N+1 モードの場合、アプリケーションの自動再起動はサポートされません。デバイスが N+1 モードの場合、次のログメッセージがコンソールに表示されます。

```
Feb 5 20:29:17.022: %IOX-3-IOX_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 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 7: 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
前面パネルポート (トランクおよび VLAN)	<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 17.5.1 の Catalyst 9600 シリーズ スイッチ • Cisco IOS XE Bengaluru 17.6.1 の Catalyst 9300X シリーズ スイッチ <p>(注) Catalyst 9300X シリーズ スイッチは、複数の AppGigabitEthernet ポートをサポートします。</p>
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

仮想ネットワーク インターフェイス カード

アプリケーションコンテナのライフサイクルを管理するには、内部論理インターフェイスごとに1つのコンテナをサポートするレイヤ3ルーティングモデルが使用されます。これは、各アプリケーションに対して仮想イーサネットペアが作成されることを意味します。このペアのうち仮想ネットワークインターフェイスカードと呼ばれるインターフェイスは、アプリケーションコンテナの一部です。

NICは、コンテナ内の標準イーサネットインターフェイスで、プラットフォームデータプレーンに接続してパケットを送受信します。Cisco IOx は、コンテナ内の各vNICについて、IPアドレスおよび一意のMACアドレス割り当てを行います。

コンテナ内のvNICは、標準のイーサネットインターフェイスと見なされます。

アプリケーションホスティングの設定方法

ここでは、アプリケーションホスティングの設定を構成するさまざまな作業について説明します。

Cisco IOx の有効化

このタスクを実行して Cisco IOx へのアクセスを有効にすることで、CLI ベースのユーザインターフェイスでホストシステム上のアプリケーションの管理、制御、モニタ、トラブルシューティング、および関連するさまざまなアクティビティを実行できます。

手順の概要

1. **enable**
2. **configure terminal**
3. **iox**
4. **username name privilege level password {0 | 7 | user-password} encrypted-password**
5. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	iox 例： Device(config)# iox	Cisco IOx をイネーブルにします。
ステップ 4	username name privilege level password {0 7 user-password} encrypted-password 例： Device(config)# username cisco privilege 15 password 0 ciscoI	ユーザー名ベースの認証システムとユーザーの権限レベルを確立します。 • ユーザー名の特権レベルは 15 に設定する必要があります。
ステップ 5	end 例： Device(config)# end	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

前面パネル VLAN ポートのアプリケーションホスティングの設定



(注) このタスクは、Cisco IOS XE Amsterdam 17.1.1 以降のリリースに適用されます。

アプリケーションホスティング トランク コンフィギュレーション モードでは、許可されるすべての AppGigabitEthernet VLAN ポートがコンテナに接続されます。ネイティブおよび VLAN タグ付きフレームは、コンテナ ゲスト インターフェイスによって送受信されます。AppGigabitEthernet トランクポートにマッピングできるコンテナ ゲスト インターフェイスは 1 つだけです。

トランクポートと VLAN アクセスポートの両方の同時設定がサポートされます。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface AppGigabitEthernet number**
4. **switchport trunk allowed vlan vlan-ID**
5. **switchport mode trunk**
6. **exit**
7. **app-hosting appid name**
8. **app-vnic AppGigabitEthernet trunk**
9. **vlan vlan-ID guest-interface guest-interface-number**
10. **guest-ipaddress ip-address netmask netmask**
11. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 <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。
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。 <ul style="list-style-type: none">複数の VLAN からゲストインターフェイスへのマッピングがサポートされます。

	コマンドまたはアクション	目的
ステップ 10	guest-ipaddress ip-address netmask netmask 例： Device (config-config-app-hosting-vlan-access-ip) # guest-ipaddress 192.168.0.2 netmask 255.255.255.0	(オプション) 静的 IP を設定します。
ステップ 11	end 例： Device (config-config-app-hosting-vlan-access-ip) # end	アプリケーションホスティング VLAN アクセス IP コンフィギュレーションモードを終了し、特権 EXEC モードに戻ります。

前面パネルトランクポートのアプリケーションホスティングの設定

アプリケーションホスティング トランク コンフィギュレーションモードでは、許可されるすべての AppGigabitEthernet VLAN ポートがコンテナに接続されます。ネイティブおよび VLAN タグ付きフレームは、コンテナ ゲスト インターフェイスによって送受信されます。

AppGigabitEthernet トランクポートにマッピングできるコンテナ ゲスト インターフェイスは 1 つだけです。

Cisco IOS XE Gibraltar 16.2.1 では、アプリケーション ID は、アプリケーションホスティング トランク コンフィギュレーションモードまたはアプリケーションホスティング VLAN アクセス コンフィギュレーションモードで設定できますが、両方のモードで設定することはできません。

Cisco IOS XE Amsterdam 17.1.1 以降のリリースでは、トランクポートと VLAN アクセスポートの両方の同時設定がサポートされています。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface AppGigabitEthernet number**
4. **switchport trunk allowed vlan vlan-ID**
5. **switchport mode trunk**
6. **exit**
7. **app-hosting appid name**
8. **app-vnic AppGigabitEthernet trunk**
9. **guest-interface guest-interface-number**
10. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 <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。
ステップ 9	guest-interface guest-interface-number 例： Device(config-config-app-hosting-trunk)# guest-interface 2	AppGigabitEthernet インターフェイス トランクに接続されているアプリケーションのインターフェイスを設定します。

	コマンドまたはアクション	目的
ステップ 10	end 例： Deviceconfig-config-app-hosting-trunk) # end	アプリケーションホスティングトランク コンフィギュレーションモードを終了し、特権 EXEC モードに戻ります。

コンフィギュレーションモードでのアプリケーションの起動

アプリケーションホスティングコンフィギュレーションモードの **start** コマンドは、**app-hosting activate appid** および **app-hosting start appid** コマンドと同等です。

アプリケーションホスティングコンフィギュレーションモードの **no start** コマンドは、**app-hosting stop appid** および **app-hosting deactivate appid** コマンドと同等です。



- (注) アプリケーションをインストールする前に **start** コマンドを設定してから **install** コマンドを設定すると、Cisco IOx は自動的に内部 **activate** アクションと **start** アクションを実行します。これにより、**install** コマンドを設定することでアプリケーションを自動的に起動できます。

手順の概要

1. **enable**
2. **configure terminal**
3. **app-hosting appid application-name**
4. **start**
5. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# configure terminal	グローバルコンフィギュレーションモードを開始します。
ステップ 3	app-hosting appid application-name 例： Device(config)# app-hosting appid iox_app	アプリケーションを設定し、アプリケーションホスティングコンフィギュレーションモードを開始します。
ステップ 4	start 例：	(任意) アプリケーションを起動して実行します。

	コマンドまたはアクション	目的
	Device(config-app-hosting)# start	<ul style="list-style-type: none"> 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。 <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 application-name package package-path 例： Device# app-hosting install appid iox_app package usbflash1:my_iox_app.tar	指定した場所からアプリケーションをインストールします。 <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 application-name 例：	アプリケーションをアクティブ化します。

	コマンドまたはアクション	目的
	Device# <code>app-hosting activate appid iox_app</code>	<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# <code>app-hosting start appid iox_app</code>	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# <code>app-hosting stop appid iox_app</code>	(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# <code>app-hosting deactivate appid iox_app</code>	(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# <code>app-hosting uninstall appid iox_app</code>	(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 name 例： Device(config)# app-hosting appid iox_app	アプリケーションを設定し、アプリケーションホスティング コンフィギュレーション モードを開始します。
ステップ 4	name-server# ip-address 例： Device(config-app-hosting)# name-server0 10.2.2.2	ドメインネームシステム (DNS) サーバを設定します。
ステップ 5	app-vnic management guest-interface interface-number 例： Device(config-app-hosting)# app-vnic management guest-interface 0	仮想ネットワーク インターフェイスおよびゲスト インターフェイスの管理ゲートウェイを設定し、アプリケーションホスティングゲートウェイ コンフィギュレーション モードを開始します。
ステップ 6	guest-ipaddress ip-address netmask netmask 例： Device(config-app-hosting-mgmt-gateway)# guest-ipaddress 172.19.0.24 netmask 255.255.255.0	管理ゲストインターフェイスの詳細を設定します。
ステップ 7	exit 例： Device(config-app-hosting-mgmt-gateway)# exit	アプリケーション ホスティング管理ゲートウェイ コンフィギュレーション モードを終了し、アプリケーション ホスティング コンフィギュレーション モードに戻ります。
ステップ 8	app-default-gateway ip-address guest-interface network-interface 例： Device(config-app-hosting)# app-default-gateway 172.19.0.23 guest-interface 0	デフォルトの管理ゲートウェイを設定します。

	コマンドまたはアクション	目的
ステップ 9	end 例： Device(config-app-hosting)# end	アプリケーションホスティング コンフィギュレーションモードを終了し、特権 EXEC モードに戻ります。

管理ポートでのアプリケーションホスティングの設定

手順の概要

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet0/0**
4. **vrf forwarding vrf-name**
5. **ip address ip-address mask**
6. **exit**
7. **app-hosting appid name**
8. **app-vnic management guest-interface network-interface**
9. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	interface gigabitethernet0/0 例： Device(config)# interface gigabitethernet0/0	インターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。 • Cisco Catalyst 9000 シリーズ スイッチでは、管理インターフェイスは GigabitEthernet0/0 です。
ステップ 4	vrf forwarding vrf-name 例： Device(config-if)# vrf forwarding Mgmt-vrf	インターフェイスまたはサブインターフェイスに Virtual Routing and Forwarding (VRF) インスタンスまたは仮想ネットワークを関連付けます。 • <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。

	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	app-hosting appid name 例： Device(config)# app-hosting appid iox_app	アプリケーションを設定し、アプリケーションホスティング コンフィギュレーション モードを開始します。
ステップ 8	app-vnic management guest-interface network-interface 例： Device(config-app-hosting)# app-vnic management guest-interface 1	ゲストインターフェイスを管理ポートに接続し、アプリケーションホスティング管理ゲートウェイ コンフィギュレーション モードを開始します。 <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 network-interface のキーワード引数ペアは、Cisco IOS 管理インターフェイスに接続されているコンテナの内部イーサネット インターフェイス番号を指定します。この例では、コンテナのイーサネット 1 インターフェイスに対して <i>guest-interface 1</i> を使用しています。
ステップ 9	end 例： Device(config-app-hosting-mgmt-gateway)# end	アプリケーションホスティング管理ゲートウェイ コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

アプリケーションの IP アドレスの手動設定

次の方法を使用して、コンテナの IP アドレスを設定できます。

- コンテナにログインし、**ifconfig** Linux コマンドを設定します。

1. 次のコマンドを使用して、アプリケーションにログインします。

```
app-hosting connect appid APPID {session | console}
```

2. アプリケーションの Linux サポートに基づいて、標準の Linux インターフェイス コンフィギュレーション コマンドを使用します。

```
- ifconfig dev IFADDR/subnet-mask-length
```

または

```
- ip address {add|change|replace} IFADDR dev IFNAME [ LIFETIME ] [ CONFFLAG-LIST ]
```

- コンテナで Dynamic Host Configuration Protocol (DHCP) を有効にし、Cisco IOS の設定で DHCP サーバとリレーエージェントを設定します。
- Cisco IOx は、アプリケーション DHCP インターフェイスに使用されるアプリケーションコンテナ内で実行する DHCP クライアントを提供します。

アプリケーションのリソース設定の上書き

リソースの変更を有効にするには、最初に **app-hosting stop** および **app-hosting deactivate** コマンドを使用してアプリケーションを停止して非アクティブ化し、次に **app-hosting activate** および **app-hosting start** コマンドを使用してアプリケーションを再起動する必要があります。

アプリケーションホスティングコンフィギュレーションモードで **start** コマンドを使用している場合は、**no start** および **start** コマンドを設定します。

これらのコマンドを使用して、リソースと **app-hosting appid iox_app** 設定の両方をリセットできます。

手順の概要

1. **enable**
2. **configure terminal**
3. **app-hosting appid name**
4. **app-resource profile name**
5. **cpu unit**
6. **memory memory**
7. **vcpu number**
8. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# configure terminal	グローバルコンフィギュレーションモードを開始します。

	コマンドまたはアクション	目的
ステップ 3	app-hosting appid name 例： Device(config)# app-hosting appid iox_app	アプリケーションホスティングをイネーブルにし、アプリケーションホスティング コンフィギュレーションモードを開始します。
ステップ 4	app-resource profile name 例： Device(config-app-hosting)# app-resource profile custom	カスタムアプリケーションリソースプロファイルを設定し、カスタムアプリケーションリソースプロファイル コンフィギュレーションモードを開始します。 <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： 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。 <ul style="list-style-type: none"> • 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 1keys	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 <i>ip-address netmask netmask</i> 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 <i>ip-address guest-interface network-interface</i> 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# <i>ip-address</i> 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 <i>options</i> 例： Device (config-app-hosting-docker) # run-opts 1 "-e TEAGENT_ACCOUNT_TOKEN=[account-token]"	Docker ランタイムオプションを指定します。
ステップ 13	prepend-pkg-opts 例： Device (config-app-hosting-docker) # prepend-pkg-opts	パッケージオプションを Docker ランタイムオプションとマージします。 • 重複する変数は上書きされます。
ステップ 14	end 例： Device (config-app-hosting-docker) # end	アプリケーションホスティング Docker コンフィギュレーションモードを終了し、特権 EXEC モードに戻ります。

ThousandEyes Enterprise Agent の AppGigabitEthernet インターフェイスの設定

手順の概要

1. **enable**
2. **configure terminal**
3. **interface appgigabitethernet *number***
4. **switchport trunk allowed vlan *vlan-ID***
5. **switchport mode trunk**
6. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを開始します。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	interface appgigabitethernet <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
Libvirt                    : 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
```

例：コンテナの静的 IP アドレスの設定

```
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 serial10
serial/aux      iox_console_aux   serial11
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 8: 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 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。
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	<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 USB ポート間のデータパス接続を導入します。</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。

機能名	リリース	機能情報
アプリケーションホスティング：ThousandEyesの統合	Cisco IOS XE Amsterdam 17.3.3 Cisco IOS XE Bengaluru 17.5.1 Cisco IOS XE Bengaluru 17.6.1	ThousandEyes は、クラウド対応のエンタープライズ ネットワーク監視ツールであり、ネットワークとサービス全体のエンドツーエンドのビューを提供します。 <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。 (注) ThousandEyes 統合機能は、Cisco IOS XE Bengaluru 17.4.x リリースではサポートされていません。
ThousandEyes BrowserBot	Cisco IOS XE Bengaluru 17.6.1	ThousandEyes アドオンエージェントモードがサポートされています。アドオンモードは、トランザクションのスク립ト化テスト用の BrowserBot を提供します。 <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

機能名	リリース	機能情報
ネイティブ Docker コンテナ：アプリケーションの自動再起動	Cisco IOS XE Amsterdam 17.2.1 Cisco IOS XE Bengaluru 17.5.1	<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。

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。