



# ThousandEyes Enterprise Agent

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。ここでは、Enterprise Agent をダウンロードしてインストールする方法について説明します。

- [ThousandEyes Enterprise Agent の前提条件](#) (1 ページ)
- [ThousandEyes Enterprise Agent について](#) (2 ページ)
- [ThousandEyes Enterprise Agent のインストール方法](#) (6 ページ)
- [ThousandEyes Enterprise Agent の設定例](#) (10 ページ)
- [その他の参考資料](#) (15 ページ)
- [ThousandEyes Enterprise Agent の機能情報](#) (15 ページ)

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

1:1 モードとは、スタック内の特定のデバイスにアクティブロールとスタンバイロールが割り当てられる場合です。これは、スタック内の任意のスイッチをアクティブまたはスタンバイにすることができる従来の N+1 ロール選択アルゴリズムより優先されます。

# ThousandEyes Enterprise Agent について

## 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 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 1: 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"> <li>• CPU : 2 vCPU</li> <li>• メモリ : 2G RAM</li> <li>• ストレージ : SSD 上の制限なし</li> </ul>	Cisco IOS XE Amsterdam 17.3.3 <ul style="list-style-type: none"> <li>• Cisco Catalyst 9300 および 9300L シリーズ スイッチ</li> </ul> Cisco IOS XE Bengaluru 17.5.1 <ul style="list-style-type: none"> <li>• Cisco Catalyst 9400 シリーズ スイッチ</li> </ul> Cisco IOS XE Bengaluru 17.6.1 <ul style="list-style-type: none"> <li>• Cisco Catalyst 9300X シリーズ スイッチ</li> </ul>
フラッシュ	<ul style="list-style-type: none"> <li>• CPU : 2 vCPU</li> <li>• メモリ : 2G RAM</li> <li>• ストレージ : フラッシュファイルシステムの 4G パーティションのうち、アプリケーションによる永続的なロギング用に 1G。ストレージは IOx メタデータと共有されます。</li> </ul>	Cisco IOS XE Amsterdam 17.3.3 <ul style="list-style-type: none"> <li>• Cisco Catalyst 9300 および 9300L シリーズ スイッチ</li> </ul> Cisco IOS XE Bengaluru 17.5.1 <ul style="list-style-type: none"> <li>• Cisco Catalyst 9400 シリーズ スイッチ</li> </ul> Cisco IOS XE Bengaluru 17.6.1 <ul style="list-style-type: none"> <li>• Cisco Catalyst 9300X シリーズ スイッチ</li> </ul>

Cisco IOS XE Bengaluru 17.6.1では、アドオンモードはCisco Catalyst9300、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 2: 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"> <li>「<a href="#">Install Enterprise Agents on Cisco Switches with Docker</a>」 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 <a href="http://www.cisco.com">www.cisco.com</a> 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。</li> <li>アプリケーションをダウンロードして展開するには、<b>install</b> コマンドを使用します。</li> </ul>	<ul style="list-style-type: none"> <li>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。</li> <li>アプリケーションをダウンロードして展開するには、<b>install</b> コマンドを使用します。</li> </ul>

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。

## ThousandEyes Enterprise Agent のインストール方法

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	<b>enable</b> 例： Device> enable	特権 EXEC モードを開始します。
ステップ 2	<b>configure terminal</b> 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	<b>app-hosting appid application-name</b> 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	<b>app-vnic AppGigabitEthernet trunk</b> 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	<b>vlan vlan-ID guest-interface guest-interface-number</b> 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。
ステップ 6	<b>guest-ip ip-address netmask netmask</b> 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	<b>exit</b> 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	<b>exit</b> 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	<b>app-default-gateway ip-address guest-interface network-interface</b> 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 10	<b>nameserver# ip-address</b> 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	<b>app-resource docker</b> 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	<b>run-opts options</b> 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	<b>prepend-pkg-opts</b> 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	<b>end</b> 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	<b>enable</b> 例： Device> enable	特権 EXEC 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 2	<b>configure terminal</b> 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	<b>interface appgigabitethernet number</b> 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	<b>switchport trunk allowed vlan vlan-ID</b> 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	<b>switchport mode trunk</b> 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	<b>end</b> 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	<b>enable</b> 例： Device> enable	特権 EXEC 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 2	<b>app-hosting install appid <i>application-name</i> package <i>package-path</i></b> 例 : <pre>Device# app-hosting install lkeys https://downloads.thousandeyes.com/ enterprise-agent/thousandeyes-enterprise-agent-3.0.cat9k.tar</pre> または <pre>Device# app-hosting install appid lkeys package flash:/apps/[greenfield-app-tar]</pre>	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	<b>app-hosting start appid <i>application-name</i></b> 例 : <pre>Device# app-hosting start appid lkeys</pre>	(オプション) アプリケーションを開始します。
ステップ 4	<b>end</b> 例 : <pre>Device# end</pre>	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
```

## ThousandEyes Enterprise Agent の設定例

### 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           : 1keyes
Owner            : iox
State           : RUNNING
Application
  Type           : docker
  Name           : thousandeyes/enterprise-agent
  Version        : 3.0
  Description    :
  Path           : flash:thousandeyes-enterprise-agent-3.0.cat9k.tar
  URL Path       :
Activated profile name : custom

Resource reservation
  Memory         : 0 MB
  Disk           : 1 MB
  CPU            : 1850 units
  CPU-percent    : 25 %
  VCPU          : 1

Attached devices
  Type           Name           Alias
  -----
serial/shell    iox_console_shell  serial0
serial/aux      iox_console_aux    serial1
serial/syslog   iox_syslog         serial2
serial/trace    iox_trace          serial3

Network interfaces
-----
eth0:
  MAC address    : 52:54:dd:c0:a2:ab
  IPv4 address   : 10.0.0.110
  IPv6 address   : ::
  Network name   : mgmt-bridge-v14

Docker
-----
Run-time information
  Command       :
  Entry-point   : /sbin/my_init
  Run options in use : -e TEAGENT_ACCOUNT_TOKEN=TOKEN_NOT_SET --hostname=$(SYSTEM_NAME)
  --cap-add=NET_ADMIN
  --mount type=tmpfs,destination=/var/log/agent,tmpfs-size=140m
  --mount
  type=tmpfs,destination=/var/lib/te-agent/data,tmpfs-size=200m
  -v $(APP_DATA)/data:/var/lib/te-agent -e TEAGENT_PROXY_TYPE=DIRECT
  -e TEAGENT_PROXY_LOCATION= -e TEAGENT_PROXY_USER= -e
TEAGENT_PROXY_AUTH_TYPE=
  -e TEAGENT_PROXY_PASS= -e TEAGENT_PROXY_BYPASS_LIST= -e
TEAGENT_KDC_USER=
  -e TEAGENT_KDC_PASS= -e TEAGENT_KDC_REALM= -e TEAGENT_KDC_HOST=
  -e TEAGENT_KDC_PORT=88
  -e TEAGENT_KERBEROS_WHITELIST= -e TEAGENT_KERBEROS_RDNS=1 -e
PROXY_APT=
  -e APT_PROXY_USER= -e APT_PROXY_PASS= -e APT_PROXY_LOCATION=
-e TEAGENT_AUTO_UPDATES=1
  -e TEAGENT_ACCOUNT_TOKEN=r3d29srpebr4j8451vnamwhswlori2xs
  --hostname=cat9k-9300-usb --memory=1g
  Package run options : -e TEAGENT_ACCOUNT_TOKEN=TOKEN_NOT_SET --hostname=$(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

関連項目	マニュアル タイトル
ThousandEyes URL	<a href="https://app.thousandeyes.com">https://app.thousandeyes.com</a>

### 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><a href="http://www.cisco.com/support">http://www.cisco.com/support</a></p>

## ThousandEyes Enterprise Agent の機能情報

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

プラットフォームのサポートおよびシスコソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、[www.cisco.com/go/cfn](http://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 3: 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
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	<p>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。</p> <ul style="list-style-type: none"> <li>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。</li> <li>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。</li> <li>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。</li> </ul> <p>(注) ThousandEyes 統合機能は、Cisco IOS XE Bengaluru 17.4.x リリースではサポートされていません。</p>
ThousandEyes BrowserBot	Cisco IOS XE Bengaluru 17.6.1	<p>ThousandEyes アドオンエージェントモードがサポートされています。アドオンモードは、トランザクションのスク립ト化テスト用の BrowserBot を提供します。</p> <p>Cisco IOS XE Bengaluru 17.6.1では、この機能は次のプラットフォームで導入されました。</p> <ul style="list-style-type: none"> <li>Cisco Catalyst 9300、9300L、および 9300X シリーズ スイッチ</li> <li>Cisco Catalyst 9400 シリーズ スイッチ</li> </ul>



## 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。