# ThousandEyes Enterprise Agent

ThousandEyes Enterprise Agent is an enterprise network-monitoring tool that provides you an end-to-end view across networks and services that impact your business. This module describes how to download and install the Enterprise Agent.


- Prerequisites for the ThousandEyes Enterprise Agent, on page 1
- Information About ThousandEyes Enterprise Agent, on page 2
- How to Install the ThousandEyes Enterprise Agent, on page 5
- Configuration Examples for ThousandEyes Enterprise Agent, on page 10
- Additional References, on page 14
- Feature Information for ThousandEyes Enterprise Agent, on page 14


# Prerequisites for the ThousandEyes Enterprise Agent

- The ThousandEyes Enterprise Agent image available at the ThousandEyes site must be signed by the same certificate authority (CA) that is used by www.cisco.com for HTTPS downloads; without an username or a password.
- Installation of the Enterprise Agent requires Internet connectivity, or a proxy server. For more information, see the *ThousandEyes documentation* at: https://docs.thousandeyes.com/product-documentation/enterprise-agents.
- The Enterprise Agent application can only be used after the user's license privileges are validated.
- Only Docker-based applications are supported.
- 1:1 stack mode is a must for ThousandEyes Stateful Switchover (SSO) support.

  1:1 mode is when the active and standby roles are assigned to specific devices in a stack. This overrides the traditional N+1 role selection algorithm, where any device in the stack can be the active or the standby.

# Information About ThousandEyes Enterprise Agent

## ThousandEyes Enterprise Agent Overview

ThousandEyes Enterprise Agent is an enterprise network-monitoring tool that provides you an end-to-end view across networks and services that impact your business. It monitors the network traffic paths across internal, external, carrier, and Internet networks in real time, to provide network performance data. Enterprise Agents are commonly installed in branch sites and data centers to provide a detailed understanding of WAN and Internet connectivity.

In previous Cisco IOS XE releases, ThousandEyes was supported as a third-party Kernel-based Virtual Machine (KVM) appliance on the SSD.

In Cisco IOS XE Amsterdam 17.3.3, a new version of the ThousandEyes Enterprise Agent, Version 3.0 is introduced. This is an embedded Docker-based application that runs on Cisco devices using the application-hosting capability. The Enterprise Agent is available on both the SSD and bootflash, and it supports all tests except browser tests (page load and transaction). The browser tests are available in Cisco IOS XE Bengaluru 17.6.1 and later releases with Enterprise Agent Version 4.0.

The ThousandEyes Enterprise Agent provides the following:

- Benchmarking the performance of networks and applications.
- Detailed hop-by-hop metrics.
- End-to-end path visualization from branch or campus to data center or cloud.
- Outage detection and resolution.
- User-experience analysis.
- Visualization of the traffic-flow pattern.

ThousandEyes Enterprise Agent Version 4.0 available in Cisco IOS XE Bengaluru 17.6.1, supports the following additional features that are not available in the ThousandEyes Agent Version 3.0:

- BrowserBot support when back-panel SSD is available.
- DNAC app icon and description.
- Docker health monitoring.
- The **app-hosting upgrade URL** command to upgrade the ThousandEyes Enterprise Agent.

## Resources Required for the ThousandEyes Enterprise Agent

This table describes the required resources for installing the ThousandEyes Enterprise Agent:

*Table 1: Resources Required for the ThousandEyes Enterprise Agent*

| App Media | Maximum Resource | Supported Release |
|---|---|---|
| SSD<br>**Note**<br>Only 120G SSD is supported. | • CPU: 2 vCPU<br>• Memory: 2G RAM<br>• Storage: No limit on SSD | Cisco IOS XE Amsterdam 17.3.3<br>• Cisco Catalyst 9300 and 9300L Series Switches<br>Cisco IOS XE Bengaluru 17.5.1<br>• Cisco Catalyst 9400 Series Switches<br>Cisco IOS XE Bengaluru 17.6.1<br>• Cisco Catalyst 9300X Series Switches |
| Flash | • CPU: 2 vCPU<br>• Memory: 2G RAM<br>• Storage: 1G for persistent logging by applications, out of the 4G partition in the flash file system. The storage is shared with the IOx metadata. | Cisco IOS XE Amsterdam 17.3.3<br>• Cisco Catalyst 9300 and 9300L Series Switches<br>Cisco IOS XE Bengaluru 17.5.1<br>• Cisco Catalyst 9400 Series Switches<br>Cisco IOS XE Bengaluru 17.6.1<br>• Cisco Catalyst 9300X Series Switches |

In Cisco IOS XE Bengaluru 17.6.1, add-on mode is supported on Cisco Catalyst 9300, 9300L, and 9300X Series Switches, and Cisco Catalyst 9400 Series Switches.

# ThousandEyes Enterprise Agent Download

BrownField and GreenField are two types of ThousandEyes Enterprise Agents. For existing devices, you can download the Brownfield version from the ThousandEyes website. However, new devices are shipped with the Greenfield application loaded in the bootflash.

This table lists the download options available for the agents.

*Table 2: ThousandEyes Enterprise Agent Download Options*

| BrownField | GreenField |
|---|---|
| • Download the file from the *Installing Enterprise Agents on Cisco Switches with Docker* page. The file is signed by the same certificate authority (CA) that is used by www.cisco.com for HTTPS downloads; without an username or a password.<br>• Use the **install** command to download and deploy the application. | • Available in the bootflash under the */apps* folder. Shipped with the device.<br>• Use the **install** command to download and deploy the application. |

This section describes the maximum resources required for the agent to run:

- CPU: 2 vCPUs
- Memory: 2G
- Storage: 1G for persistent logging by applications, out of the 4G partition in the flash file system. This storage is shared by the IOx metadata.
- Media storage:
  - 120G SSD for Cisco Catalyst 9300 and Cat9300 L Series Switches in Cisco IOS XE Amsterdam 17.3.3.
  - 240/480/960GB M2-SATA-HDD for Cisco Catalyst 9400 Series Switches in Cisco IOS XE Bengaluru 17.5.1.

After the download of the Enterprise Agent, it initiates a call to create a secure channel to the ThousandEyes cloud-based portal that provides the required application configuration, and gathers application data. The link to the TE portal is https://app.thousandeyes.com.

# ThousandEyes BrowserBot

ThousandEyes Enterprise Agent Version 4.0 provides a BrowserBot for transaction scripting test. The BrowserBot is a component of the Enterprise Agent that manages page load and transaction tests. The BrowserBot allows you to enable customized JavaScript tests which mimic the actions of your web browser on the ThousandEyes Cloud Portal. To protect the host operating system from any errant JavaScript operations, the ThousandEyes agent creates sandbox containers to run your JavaScript.

If an unrestricted disk is used by the application, the ThousandEyes agent will dynamically install the BrowserBot package during initialization that permits portal transaction scripting tests to be configured.

**Note** The BrowserBot support is not available in ThousandEyes Agent Version 3.0.

BrowserBot consumes a large amount of hardware resources. 2GB system memory and 2 VCPU loads are the maximum IOx system memory and CPU load allocated for all IOx apps. To allow multiple apps to concurrently run in the bootflash, lower the default package.yaml BrowserBot resources before activating the agent. Use the **app-resource profile custom** command to override the default package.yaml settings:

- CPU:1850 CPU units (1/4 VCPU)
- Memory: 500MB

For more information on transaction scripting, see the following links:

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

For examples of transaction scripting, see https://github.com/thousandeyes/transaction-scripting-examples.

# ThousandEyes Agent Upgrade and Downgrade

### ThousandEyes Agent Upgrade

The ThousandEyes Enterprise Agent 3.0 available in Cisco IOS XE Amsterdam 17.3.3 and Bengaluru 17.5.1 can be upgraded to Agent 3.0 or Agent 4.0 that is available in Cisco IOS XE Bengaluru17.6.1. Agent 3.0 is operationally restored after an upgrade.

Agent 4.0 is available in Cisco IOS XE Bengaluru 17.6.1, and the agent auto-upgrade updates to the latest Agent 4.0 binary on startup. No upgrade is available for Agent 4.0 at present.

Application upgrades can be done using the following methods:

- ThousandEyes agent auto-upgrade: Happens automatically when an application starts up. The agent binary within the running container is upgraded, but the application package is not upgraded.
- Using the **app-hosting upgrade** command.
- DNAC app upgrades.

### ThousandEyes Agent Downgrade

Agent 3.0 available in Cisco IOS XE Amsterdam 17.3.3, Cisco IOS XE Bengaluru 17.5.1, and Cisco IOS XE 17.6.1 cannot be downgraded.

Agent 4.0 available in Cisco IOS XE Bengaluru 17.6.1 can be downgraded to Agent 3.0 available in Cisco IOS XE Bengaluru 17.6.1. No other downgrade is possible.

When downgrading, if the application does not come to the same state as the previous release, deactivate or uninstall the application, and install or restart it.

# How to Install the ThousandEyes Enterprise Agent

To install the Enterprise Agent, follow these steps:

1. Configure IOx. For more information, see the "Enabling Ciso IOx" section.
2. Configure AppHosting.
3. Configure the AppGigabitEthernet port.
4. Install the ThousandEyes Enterprise Agent.

# Configuring AppHosting for the ThousandEyes Enterprise Agent

## SUMMARY STEPS

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**

## DETAILED STEPS

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enters privileged EXEC mode. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **app-hosting appid** *application-name*<br>**Example:**<br>`Device(config)# app-hosting appid appid 1keyes` | Configures an application and enters application-hosting configuration mode. |
| **Step 4** | **app-vnic AppGigabitEthernet trunk**<br>**Example:**<br>`Device(config-app-hosting)# app-vnic AppGigabitEthernet trunk` | Configures a trunk port as the front-panel port for an application, and enters application-hosting trunk-configuration mode. |
| **Step 5** | **vlan** *vlan-ID* **guest-interface** *guest-interface-number*<br>**Example:**<br>`Device(config-config-app-hosting-trunk)# vlan 10 guest-interface 2` | Configures a VLAN guest interface and enters application-hosting VLAN-access IP configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | **guest-ip** *ip-address* **netmask** *netmask*<br>**Example:**<br>`Device(config-config-app-hosting-vlan-access-ip)# guest-ipaddress 172.19.0.24 netmask 255.255.255.0` | Configures a static IP address for the guest interface. |
| **Step 7** | **exit**<br>**Example:**<br>`Device(config-config-app-hosting-vlan-access-ip)# exit` | Exits application hosting VLAN-access IP configuration mode and returns to application-hosting trunk-configuration mode. |
| **Step 8** | **exit**<br>**Example:**<br>`Device(config-config-app-hosting-trunk)# exit` | Exits application-hosting trunk-configuration mode and returns to application hosting configuration mode. |
| **Step 9** | **app-default-gateway** *ip-address* **guest-interface** *network-interface*<br>**Example:**<br>`Device(config-app-hosting)# app-default-gateway 172.19.0.23 guest-interface 0` | Configures the default management gateway. |
| **Step 10** | **nameserver#** *ip-address*<br>**Example:**<br>`Device(config-app-hosting)# name-server0 10.2.2.2` | Configures the DNS server. |
| **Step 11** | **app-resource docker**<br>**Example:**<br>`Device(config-app-hosting)# app-resource docker` | Enters application-hosting docker-configuration mode to specify application resource updates. |
| **Step 12** | **run-opts** *options*<br>**Example:**<br>`Device(config-app-hosting-docker)# run-opts 1 "-e TEAGENT_ACCOUNT_TOKEN=[account-token]"` | Specifies the Docker run time options. |
| **Step 13** | **prepend-pkg-opts**<br>**Example:**<br>`Device(config-app-hosting-docker)# prepend-pkg-opts` | Merges the package options with the Docker runtime options.<br>• Any duplicate variable is overwritten. |
| **Step 14** | **end**<br>**Example:**<br>`Device(config-app-hosting-docker)# end` | Exits application-hosting docker-configuration mode and returns to privileged EXEC mode. |

# Configuring AppGigabitEthernet Interface for the ThousandEyes Enterprise Agent

## SUMMARY STEPS

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

## DETAILED STEPS

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enters privileged EXEC mode. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface appgigabitethernet** *number*<br>**Example:**<br>`Device(config)# interface AppGigabitEthernet 1/0/1` | Configures the AppGigabitEthernet and enters interface configuration mode.<br>• For stackable switches, the *number* argument is *switch-number/0/1*. |
| **Step 4** | **switchport trunk allowed vlan** *vlan-ID*<br>**Example:**<br>`Device(config-if)# switchport trunk allowed vlan 10-12,20` | Configures the list of VLANs allowed on the trunk. |
| **Step 5** | **switchport mode trunk**<br>**Example:**<br>`Device(config-if)# switchport mode trunk` | Sets the interface into permanent trunking mode and negotiates to convert the neighboring link into a trunk link. |
| **Step 6** | **end**<br>**Example:**<br>`Device(config-if)# end` | Exits interface configuration mode and returns to privileged EXEC mode. |

# Installing the ThousandEyes Enterprise Agent

### Before you begin

You can install the ThousandEyes Enterprise Agent either from the URL given below or from the flash filesystem.

## SUMMARY STEPS

1. **enable**
2. **app-hosting install appid** *application-name* **package** *package-path*
3. **app-hosting start appid** *application-name*
4. **end**

## DETAILED STEPS

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enters privileged EXEC mode. |
| **Step 2** | **app-hosting install appid** *application-name* **package** *package-path*<br>**Example:**<br>`Device# app-hosting install 1keyes https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-3.0.cat9k.tar`<br>Or<br>`Device# app-hosting install appid 1keyes package flash:/apps/[greenfield-app-tar]` | Installs an application from the specified location. |
| **Step 3** | **app-hosting start appid** *application-name*<br>**Example:**<br>`Device# app-hosting start appid 1keyes` | (Optional) Starts the application. |
| **Step 4** | **end**<br>**Example:**<br>`Device# end` | Exits application hosting configuration mode and returns to privileged EXEC mode. |

The following is sample output from the **show app-hosting list** command:

```
Device# show app-hosting list

App id                                   State
---------------------------------------------------------
```

```
1keyes                                    RUNNING
```

# Configuration Examples for ThousandEyes Enterprise Agent

## Example: Installing ThousandEyes Enterprise Agent

This example shows how to:

- Enable IOx.
- Configure AppHosting.
- Configure the AppGigabitEthernet port.
- Install the ThousandEyes Enterprise Agent.

The following example shows how to enable IOx:

```
Device> enable
Device# configure terminal
Device(config)# iox
Device(config)# username cisco privilege 15 password 0 ciscoI
Device(config)# end
```

The following example shows how to configure AppHosting:

```
Device> enable
Device# configure terminal
Device(config)# app-hosting appid appid 1keyes
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
```

The following example shows how to configure the Appgigabitethernet interface:

```
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
```

The following example shows how to install the ThousandEyes Enterprise Agent.

**Note** You can either download the BrownField application from the ThousandEyes website or install the prepackaged Greenfield application from the flash filesystem.

```
Device> enable
Device# Device# app-hosting install 1keyes https://downloads.thousandeyes.com/
enterprise-agent/thousandeyes-enterprise-agent-3.0.cat9k.tar
OR
Device# app-hosting install appid 1keyes package flash:/apps/[greenfield-app-tar]
Device# app-hosting start appid 1keyes
Device# end
```

# Sample Configuration for ThousandEyes Enterprise Agent

The following is sample output from the **show app-hosting detail** command:

```
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=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    :
```

The following sample output from the **show running-configuration** command displays the static IP address configuration:

```
Device# show running-config | section app-hosting

app-hosting appid 1keyes
 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
```

The following sample output from the **show running-configuration** command displays the static IP address configuration and the proxy server information:

```
Device# show running-config | section app-hosting

app-hosting appid 1keyes
 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
```

The following is sample output from running the app-resource Docker package merged with the Docker runtime options:

```
// 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
```

# Additional References

| Related Topic | Document Title |
|---|---|
| ThousandEyes URL | https://app.thousandeyes.com |

### Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.<br>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.<br>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. | http://www.cisco.com/support |

# Feature Information for ThousandEyes Enterprise Agent

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

*Table 3: Feature Information for Application Hosting*

| Feature Name | Release | Feature Information |
|---|---|---|
| ThousandEyes Integration | Cisco IOS XE Amsterdam 17.3.3<br>Cisco IOS XE Bengaluru 17.5.1<br>Cisco IOS XE Bengaluru 17.6.1 | ThousandEyes is a cloud-ready, enterprise network-monitoring tool that provides an end-to-end view across networks and services.<br>• In Cisco IOS XE Amsterdam 17.3.3, this feature was implemented on Cisco Catalyst 9300 and 9300L Series Switches.<br>• In Cisco IOS XE Bengaluru 17.5.1, this feature was implemented on Cisco Catalyst 9400 Series Switches.<br>• In Cisco IOS XE Bengaluru 17.6.1, this feature was implemented on Cisco Catalyst 9300X Series Switches.<br>**Note**<br>The ThousandEyes Integration feature is not supported in Cisco IOS XE Bengaluru 17.4.x release. |
| ThousandEyes BrowserBot | Cisco IOS XE Bengaluru 17.6.1 | ThousandEyes add-on agent mode is supported. Add-on mode provides a BrowserBot for transaction scripting test.<br>In Cisco IOS XE Bengaluru 17.6.1, this feature was introduced on the following platforms:<br>• Cisco Catalyst 9300, 9300L, and 9300X Series Switches<br>• Cisco Catalyst 9400 Series Switches |