Cisco NFVIS ThousandEyes

Cisco NFVIS ThousandEyes support

Use this reference to understand the capabilities and benefits of Cisco NFVIS ThousandEyes support for network monitoring and performance visibility.

Cisco NFVIS ThousandEyes support is a pre-integrated solution that

- allows deploying ThousandEyes Enterprise Agents as a container on Cisco NFVIS
- enables running ThousandEyes network monitoring and testing capabilities directly on your Cisco NFVIS infrastructure, and
- provides visibility into the performance of the underlying network infrastructure.

ThousandEyes container upgrade and performance benefits

Cisco NFVIS also supports ThousandEyes Docker Container Upgrade using existing VM lifecycle workflow. The VM lifecycle allows seamless image updates on the existing ThousandEyes Container deployment. Additionally, ThousandEyes has released NFVIS specific Docker container images with install type as NFVIS. You must specify the nfvis.docker ThousandEyes container image download URL in the update deployment request. NFVIS will then download the image and complete the upgrade of existing container to the specified image.

This support provides these benefits:

- Gain end-to-end visibility into the performance of your network infrastructure, including cloud providers, WAN links, and internal data center networks within Cisco NFVIS providing end-to-end visibility into network performance.
- Reduce mean time to resolution (MTTR) for network issues, improve network reliability, and optimize application performance.

Prerequisites for Cisco NFVIS ThousandEyes support

Ensure that your Cisco NFVIS environment meets all prerequisite requirements before enabling ThousandEyes support.

- Ensure that the minimum software version for Cisco NFVIS devices is Cisco NFVIS Release 4.18.2a.
- Ensure that your devices are meeting the minimum hardware requirements. For more information on the minimum hardware requirements see, Enterprise Agent System Requirements in the ThousandEyes documentation.
- Configure nameservers on your Cisco NFVIS devices to enable the download of the ThousandEyes Docker image from downloads.thousandeyes.com.

ThousandEyes deployment on Cisco NFVIS

A ThousandEyes deployment on Cisco NFVIS is a network monitoring implementation that

- can be installed using the NFVIS web interface,
- supports command-line installation options, and
- provides network monitoring capabilities on the NFVIS platform.

Deploy ThousandEyes container from NFVIS web interface

This task enables you to deploy ThousandEyes containers on Cisco NFVIS infrastructure and perform upgrades when newer container images become available.

Before deploying ThousandEyes containers, you must download the latest ThousandEyes container image file from the **Add New Enterprise Agent** dialog. You also need to upload the ThousandEyes container image file to the Cisco NFVIS and register the image. For more information on uploading and registering the image files see, Uploading VM images to an NFVIS server and Register a remote virtual machine image.

Before you begin

- Download the latest **ThousandEyes** container image file from **Add New Enterprise Agent** dialog.
- Upload the **ThousandEyes** container image file to the Cisco NFVIS and register the image.

Follow these steps to deploy ThousandEyes VM:

Procedure

Step 1 In Cisco NFVIS portal, click **Configuration** > **Deploy**.

Step 2 In the **Select VM or Container** menu, click **TE** and the ThousandEyes node is displayed inside the topology.

Step 3 Configure the fields in the **VM Details** window.

Field	Description
VM Name	Enter the container's name.
Image	Choose the registered **thousandeyes.docker** image from the drop-down or click + button to add a new image.
Profile	Choose the profile for the ThousandEyes agent from the drop-down or click the + button to add a new profile. For more information, see Enterprise Agent System Requirements for agent without browser-bot version.
Group Name	(Optional) Choose a group name from the drop-down list.
Deployment Disk	Choose a deployment disk from the drop-down. **Note** Do not choose NFS Store as a deployment disk from the drop-down.
Add Bootstrap Config	Add bootstrap config data in the explanded **Add Bootstrap Config** pane. Check the **Is External File Path** check box if your bootstrap file path is from an external source.
Add Config Options	Add the additional configuration in the expanded **Add Config Options** pane. Choose the **Config Option Name**, Enter the **Config Option Value** after = in the pre-populated config option value. For example, TEAGENT_INET=4
Add Volume	Volume required to enable ThousandEyes on Cisco NFVIS are pre-populated with the minimum required volume. You can add or remove volumes based on your requirement.
Remove Volume	You can remove volumes based on your requirement.
Exclude Disks from Export	Not supported by ThousandEyes.

Step 4 Click **Deploy**.

Step 5 To upgrade ThousandEyes Agent Container from NFVIS Web Interface, use the **Upgrade** icon from the **Manage Deployments** page.

- To upgrade a registered image:
  a. From Cisco NFVIS portal, click **Configuration** > **Virtual Machine** > **Manage**.
  b. Click the **Upgrade** icon.
  c. Choose the image from a drop-down list, displays all previously registered images.
  d. Click **Submit** to upgrade your existing configuration to the selected image.

- To upgrade using a new image:
  a. From Cisco NFVIS portal, click **Configuration** > **Virtual Machine** > **Manage**.
  b. Click the **Upgrade** icon.

c. Click **New Image**.

d. Choose a protocol.

Field	Description
HTTPS	The server path is prefilled. Enter the image name.
Docker	The repository name is prefilled. Enter the tag name of the Docker image.

e. Click **Submit** to register your new image.

The ThousandEyes container is deployed on Cisco NFVIS and is ready for monitoring network performance. If you performed an upgrade, the existing configuration is updated with the selected image.

Deploy ThousandEyes container on NFVIS using CLI configs

This task allows you to deploy and manage ThousandEyes Enterprise Agent containers on NFVIS infrastructure using various configuration methods including CLI, NETCONF, and RESTCONF.

ThousandEyes Enterprise Agents can be deployed as containers on NFVIS platforms to provide network monitoring capabilities. This deployment can be accomplished through multiple methods depending on your operational preferences and automation requirements.

Before you begin

Follow these steps to deploy ThousandEyes Container on NFVIS using CLI configurations:

Procedure

Step 1 Register the ThousandEyes Docker Image

Note
The URL in **`src`** is provided by **Add New Enterprise Agent dialog**.

Example:

```
vm_lifecycle images image thousandeyes-enterprise-agent-0.33.0
 src
https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-0.33.0-nfvis.docker

 locator vim_id container
 properties property vnf_type
  value THOUSANDEYES
 !
!
```

Step 2 Configure Resource Details as a Flavor

Example:

```
vm_lifecycle flavors flavor thousandeyes-flavor vcpus 2 memory_mb 1024 root_disk_mb 20480
!
```

Step 3 Deploy the ThousandEyes Agent Container

Example:

```
vm_lifecycle tenants tenant admin
 deployments deployment TE_DEMO
  vm_group TE_DEMO
   vim_vm_name  TE_DEMO
   locator vim_id container
   image       thousandeyes-enterprise-agent-0.33.0
   flavor      thousandeyes-flavor
   bootup_time -1
   config_data configuration bootstrap_config
    data           "{ \"env_variables\" : { \"TEAGENT_ACCOUNT_TOKEN\" : \"${TEAGENT_ACCOUNT_TOKEN}\",
 \"TEAGENT_INET\" : \"${TEAGENT_INET}\"} }"
    template_engine VELOCITY
    variable TEAGENT_ACCOUNT_TOKEN
     val [ 53rettywagbuouhw06hu65767rtyuyyui ]
    !
    variable TEAGENT_INET
     val [ 4 ]
   !
  !
 !
```

Step 4 Commit the configuration

Step 5 To upgrade the ThousandEyes agent container using CLI configurations, choose one of two options

Option 1: Register and Update the Image

- Register the New Docker Image:

```
vm_lifecycle images image thousandeyes-enterprise-agent-0.34.0
 src
https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-0.34.0-nfvis.docker

 locator vim_id container
 properties property vnf_type
  value THOUSANDEYES
 !
!
```

- Update the Container Deployment Configuration to use the newly registered image:

```
vm_lifecycle tenants tenant admin
 deployments deployment TE_DEMO
  vm_group TE_DEMO
   vim_vm_name  TE_DEMO
   locator vim_id container
   image       thousandeyes-enterprise-agent-0.34.0
  !
```

- Commit the configuration. NFVIS will handle downloading the image and upgrading the container.

- *Option 2*: Direct Image URL Update - Skip the image registration step by directly providing the ThousandEyes agent container image URL. NFVIS will automatically download the image, register it with Docker, and upgrade the container to the specified image:

```
vm_lifecycle tenants tenant admin
 deployments deployment TE_DEMO
  vm_group TE_DEMO
   vim_vm_name  TE_DEMO
   locator vim_id container
   image
https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-0.34.0-nfvis.docker
```

```
     !
```

Step 6 Configure proxy settings if your NFVIS is behind a proxy

For more information, see Configuring an Enterprise Agent to Use a Proxy Server for other ThousandEyes Agent environment variables.

Image Registration with Proxy:

```
vm_lifecycle images image thousandeyes-enterprise-agent-0.34.0
 src
https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-0.34.0-nfvis.docker

 locator vim_id container
 properties property vnf_type
  value THOUSANDEYES
 !
 properties property http_proxy
  value [ http://proxy.com ]
 !
 properties property https_proxy
  value [ http://proxy.com ]
 !
 properties property no_proxy
  value [ .cisco.com,10.1.1.1 ]
 !
!
```

Deployment Configuration with Proxy:

```
vm_lifecycle tenants tenant admin
 deployments deployment TE_DEMO
  vm_group TE_DEMO
   vim_vm_name  TE_DEMO
   locator vim_id container
   image       thousandeyes-enterprise-agent-0.33.0
   flavor      thousandeyes-flavor1
   bootup_time -1
   config_data configuration bootstrap_config
    data           "{ \"env_variables\" : { \"TEAGENT_ACCOUNT_TOKEN\" : \"${TEAGENT_ACCOUNT_TOKEN}\",
 \"TEAGENT_INET\" : \"${TEAGENT_INET}\", \"http_proxy\" : \"${http_proxy}\", \"https_proxy\" :
\"${https_proxy}\", \"no_proxy\" : \"${no_proxy}\"} }"
    template_engine VELOCITY
    variable TEAGENT_ACCOUNT_TOKEN
     val [ your token goes here ]
    !
    variable TEAGENT_INET
     val [ 4 ]
    !
    variable http_proxy
     val [ http://proxy.com:80/ ]
    !
    variable https_proxy
     val [ http://proxy.com:80/ ]
    !
    variable no_proxy
     val [ .cisco.com,10.1.1.1 ]
    !
  !
 !
```

Step 7 Deploy ThousandEyes Container on NFVIS Using NETCONF

These NETCONF payloads are examples that can be sent to NFVIS via a NETCONF client to manage the lifecycle of the Cisco ThousandEyes Container

Note
Ensure you merge these payloads into the existing configuration to avoid deleting other configurations. Refer to your NETCONF client documentation for merging procedures.

Image Registration

From Docker Hub:

```
<?xml version="1.0"?>
<vm_lifecycle xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
  <images>
    <image>
      <name>te_latest</name>
      <src>docker://thousandeyes/enterprise-agent:latest-agent</src>
      <credentials>
        <username>${DOCKER_USERNAME_HERE}</username> <!--Credentials are optional-->
        <password>${DOCKER_PAT}</password>
      </credentials>
      <properties>
        <property>
          <name>http_proxy</name> <!--Proxies are optional and depend on your topology-->
          <value>http://example.com:80/</value>
        </property>
        <property>
          <name>https_proxy</name>
          <value>http://example.com:80/</value>
        </property>
        <property>
          <name>vnf_type</name>
          <value>THOUSANDEYES</value>
        </property>
      </properties>
      <locator>
        <vim_id>container</vim_id>
      </locator>
    </image>
  </images>
</vm_lifecycle>
```

From ThousandEyes Webserver:

```
<?xml version="1.0"?>
<vm_lifecycle xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
  <images>
    <image>
      <name>te_latest</name>

<src>https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-0.34.0-nfvis.docker</src>

      <properties>
        <property>
          <name>http_proxy</name> <!--Proxies are optional and depend on your topology-->
          <value>http://example.com:80/</value>
        </property>
        <property>
          <name>https_proxy</name>
          <value>http://example.com:80/</value>
        </property>
        <property>
          <name>vnf_type</name>
          <value>THOUSANDEYES</value>
```

```
        </property>
      </properties>
      <locator>
        <vim_id>container</vim_id>
      </locator>
    </image>
  </images>
</vm_lifecycle>
```

Flavor Registration

```
<?xml version="1.0"?>
<vm_lifecycle xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
  <flavors>
    <flavor>
      <name>TE</name>
      <vcpus>2</vcpus>
      <memory_mb>1024</memory_mb>
      <root_disk_mb>8192</root_disk_mb>
    </flavor>
  </flavors>
</vm_lifecycle>
```

Deployment

```
<?xml version="1.0" encoding="UTF-8"?>
<vm_lifecycle xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
  <tenants>
    <tenant>
      <name>admin</name>
      <deployments>
        <deployment>
          <name>te_nfvis</name>
          <vm_group>
            <name>te_nfvis</name>
            <image>te_latest</image>
            <flavor>TE</flavor>
            <bootup_time>-1</bootup_time>
            <config_data>
              <configuration>
                <dst>bootstrap_config</dst>
                <data>{
    "env_variables" : {
        "TEAGENT_ACCOUNT_TOKEN" : "${TEAGENT_ACCOUNT_TOKEN}",
        "TEAGENT_INET" : "${TEAGENT_INET}",
        "http_proxy" : "${http_proxy}",
        "https_proxy" : "${https_proxy}",
        "no_proxy" : "${no_proxy}"
    }
}</data>
                 <variable>
                    <name>TEAGENT_INET</name>
                    <val>4</val>
                </variable>
                <variable>
                    <name>TEAGENT_ACCOUNT_TOKEN</name>
                    <val>53rettywagbuouhw06hu65767rtyuyyui</val>
                </variable>
                <variable>
                    <name>https_proxy</name>
                    <val>http://example.com:80/</val>
                </variable>
                <variable>
                    <name>http_proxy</name>
                    <val>http://example.com:80/</val>
```

```
                </variable>
                <variable>
                    <name>no_proxy</name>
                    <val>.example.com,1.2.3.4</val>
                </variable>
              </configuration>
            </config_data>
          </vm_group>
        </deployment>
      </deployments>
    </tenant>
  </tenants>
</vm_lifecycle>
```

ThousandEyes Container Deployment Upgrade on NFVIS Using NETCONF:

```
<?xml version="1.0" encoding="UTF-8"?>
<vm_lifecycle xmlns="http://www.cisco.com/nfvis/vm_lifecycle">
  <tenants>
    <tenant>
      <name>admin</name>
      <deployments>
        <deployment>
          <name>te_nfvis</name> <!--The deployment name and vm_group name MUST match the existing
deployment that you wish to upgrade-->
          <vm_group>
            <name>te_nfvis</name>

<image>https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-0.35.0-nfvis.docker</image>
 <!--Only image tag needs to be updated with URL or docker tag pointing to the new TE image-->
          </vm_group>
        </deployment>
      </deployments>
    </tenant>
  </tenants>
</vm_lifecycle>
```

Step 8 Deploy ThousandEyes Container on NFVIS Using RESTCONF

The following are sample RESTCONF payloads that can be sent to NFVIS using **curl** or an equivalent client to manage the lifecycle of the Cisco ThousandEyes Container:

Image Registration

From Docker Hub:

```
curl -k -v -u admin:Cisco123# -X POST 'https://172.29.91.33/restconf/data/vmlc:vm_lifecycle/images'
 \
--header 'Content-Type: application/yang-data+xml' \
--data '<image>
        <name>thousandeyes-enterprise-agent-0.34.0</name>
        <src>docker://thousandeyes/enterprise-agent:0.34.0-agent</src>
    <locator>
        <vim_id>container</vim_id>
    </locator>
    <properties>
        <property>
            <name>vnf_type</name>
            <value>THOUSANDEYES</value>
        </property>
        <property>
            <name>https_proxy</name> <!--Proxies are optional and depend on your topology-->
            <value>http://proxy.com:80/</value>
        </property>
        <property>
```

```
            <name>no_proxy</name>
            <value>.cisco.com,10.1.1.1</value>
        </property>
    </properties>
</image>'
```

From ThousandEyes Webserver:

```
curl -k -v -u admin:password -X POST 'https://nfvis_host_ip/restconf/data/vmlc:vm_lifecycle/images'
 \
--header 'Content-Type: application/yang-data+xml' \
--data '<image>
        <name>thousandeyes-enterprise-agent-0.34.0</name>

<src>https://downloads.thousandeyes.com/enterprise-agent/thousandeyes-enterprise-agent-0.33.0-nfvis.docker</src>

    <locator>
        <vim_id>container</vim_id>
    </locator>
    <properties>
        <property>
            <name>vnf_type</name>
            <value>THOUSANDEYES</value>
        </property>
        <property>
            <name>https_proxy</name> <!--Proxies are optional and depend on your topology-->
            <value>http://proxy.com:80/</value>
        </property>
        <property>
            <name>no_proxy</name>
            <value>.cisco.com,10.1.1.1</value>
        </property>
    </properties>
</image>'
```

Flavor Creation

```
curl -k -v -u admin:password -X POST 'https://nfvis_host_ip/restconf/data/vmlc:vm_lifecycle/flavors'
 \
--header 'Content-Type: application/yang-data+xml' \
--data '<flavor>
    <name>thousandeyes-flavor</name>
    <vcpus>2</vcpus>
    <memory_mb>1024</memory_mb>
    <root_disk_mb>20480</root_disk_mb>
</flavor>'
```

ThousandEyes Agent Container Deployment

```
curl -k -v -u admin:password -X POST
'https://<nfvis_host_ip>/restconf/data/vmlc:vm_lifecycle/tenants/tenant=admin/deployments' \
--header 'Content-Type: application/yang-data+xml' \
--data '<deployment>
          <name>TE_DEMO</name>
          <vm_group>
            <name>TE_DEMO</name>
            <vim_vm_name>TE_DEMO</vim_vm_name>
            <locator>
              <vim_id>container</vim_id>
            </locator>
            <image>thousandeyes-enterprise-agent-0.33.0</image>
            <flavor>thousandeyes-flavor</flavor>
            <bootup_time>-1</bootup_time>
            <config_data>
              <configuration>
              <dst>bootstrap_config</dst>
```

```
              <data>{ "env_variables" : { "TEAGENT_ACCOUNT_TOKEN" : "${TEAGENT_ACCOUNT_TOKEN}",
"TEAGENT_INET" : "${TEAGENT_INET}" } }</data>
              <template_engine>VELOCITY</template_engine>
              <variable>
                <name>TEAGENT_ACCOUNT_TOKEN</name>
                <val>53rettywagbuouhw06hu65767rtyuyyui</val>
              </variable>
              <variable>
                <name>TEAGENT_INET</name>
                <val>4</val>
              </variable>
              </configuration>
            </config_data>
          </vm_group>
        </deployment>
    '
```

Step 9 Upgrade ThousandEyes Container Deployment on NFVIS Using RESTCONF

To upgrade the existing ThousandEyes container deployment with a newer container image, you need to update the deployment configuration with the new image details. Here's a sample RESTCONF command using a Docker Hub image:

```
curl -k -v -u admin:password -X PUT
'https://<nfvis_host_ip>/restconf/data/vmlc:vm_lifecycle/tenants/tenant=admin/deployments/deployment=TE_DEMO/vm_group=TE_DEMO/image'
 \
--header 'Content-Type: application/yang-data+xml' \
--data '<image>docker://thousandeyes/enterprise-agent:0.34.0-agent</image>'
```

Note
If an upgrade fails due to issues such as an image download error, verification failure, or any other problem, the container will continue running with the old image. However, the NFVIS configuration will reflect the latest settings. This behavior is similar to how VM deployment updates are handled.

The ThousandEyes Enterprise Agent container is successfully deployed on NFVIS and ready to monitor network performance. The container will be registered with ThousandEyes and begin collecting network metrics according to the configured parameters.