Docker Sensor Setup, Release 5.1.x

First Published: 2025-01-09

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

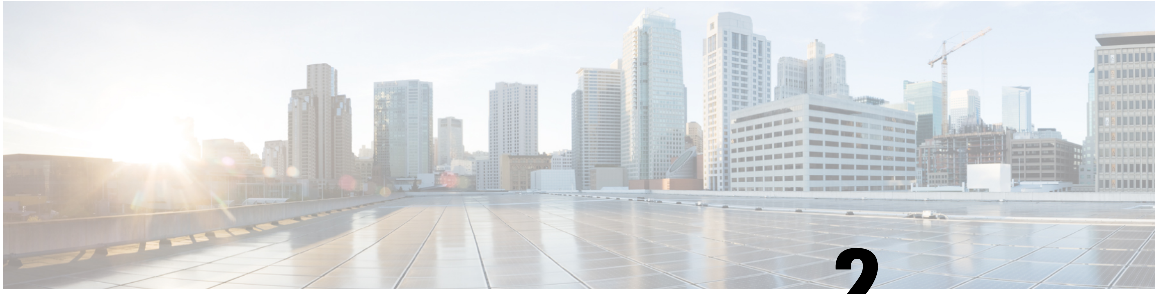
CONTENTS

CHAPTER 1

Document Purpose

Purpose

This guide explains how to perform a clean installation of the Cisco Cyber Vision Sensor on appliances running Ubuntu and Docker.



CHAPTER 2

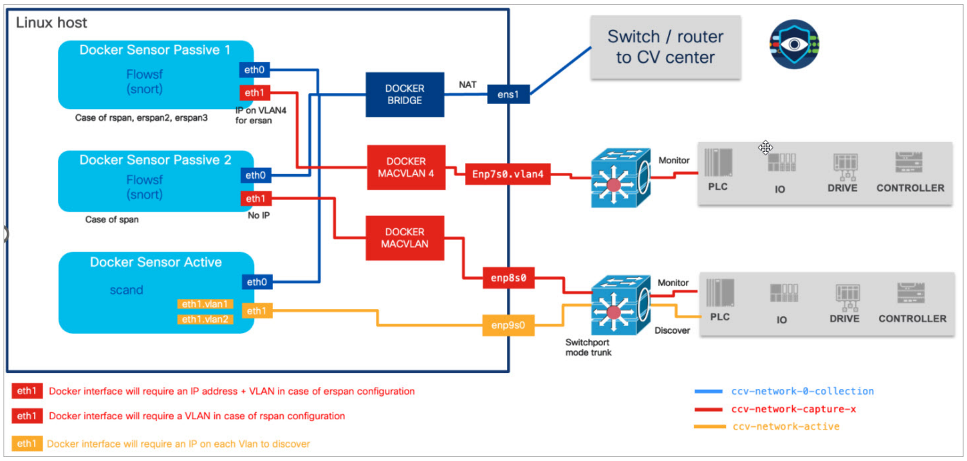
Overview

Overview

This document demonstrates the proposed architecture. Consult the local network engineer before applying the parameters. Verify IP addresses, port numbers, and VLAN IDs to ensure that configurations do not interrupt normal exchanges or halt the process.

The schema illustrates how the architecture virtually deploys on the Linux host to embed the sensor application. Configuring VLAN and physical ports allows copying OT traffic and establishing communication with the Cisco Cyber Vision Center.

The communication between the Cisco Cyber Vision Center and the sensor appears in blue. Mirrored OT traffic appears in red.



CHAPTER 3

Requirement

Docker Sensor Prerequisites

Cisco Cyber Vision 5.1.x and above provides a new way to host sensors. Cyber Vision supports Docker sensors. Docker sensor containers run on a Linux operating system. This guide lists the supported operating systems and software.

- Hardware
  - x86-64 and aarm64 devices
  - Memory: Ensure 4G of memory is available for the container.
  - Network Interface Card (NIC): The Cisco Cyber Vision Docker sensor uses promiscuous mode based on a Docker macvlan interface (passthru mode). Not all adapters are compatible. Cisco supports only Intel network adapters.
- Supported host operating systems: Ubuntu Desktop or Server versions 20.04, 22.04, and 24.04
- Docker version: 27.0 and above
- Cisco Cyber Vision release: 5.1.x and above.
- Configure the host Linux operating system with access to the CLI (ssh).
- To deploy the Docker sensor, the Linux host must have access to the center (collection interface).

Docker Sensor Definitions

- **Cisco Cyber Vision Center**: The Center receives metadata from sensors and stores it in an internal database (PostgreSQL). The Center manages all sensors.

- **Docker**: It is an open platform for developing, shipping, and running applications. It allows you to package and run an application in a loosely isolated environment called a container. Cyber Vision supports Docker.
- **Host / Linux Host**: Cyber Vision supports Docker running on Linux OS (Ubuntu V20.04, 22.04 and 24.04).
- **Docker Container**: A container is a runnable instance of an image. You can create, start, stop, move, or delete a container using the Docker API or CLI.
- **Docker Images**: An image is a read-only template with instructions for creating a Docker container.
- **Docker Compose**: Docker Compose uses a YAML configuration file, called the Compose file, to configure application services. Use the Compose CLI to create and start all services from your configuration.
- **Compose File**: The Compose file, or compose.yaml file, follows the Compose Specification rules to define multicontainer applications.
- **Docker Image Registry**: A registry centralizes the storage and management of container images. A repository collects related container images within a registry. It functions like a folder organizing images based on projects. Each repository contains one or more container images.

Cisco Cyber Vision Docker Sensor Characteristics

The Cisco Cyber Vision 5.1.x Docker sensor has specific characteristics compared to other Cisco Cyber Vision sensor types:

- There are two different Docker images:
    - A passive image includes flow, DPI service, and Snort (NIDS).
    - An active image contains only the Active Discovery service.
- Deploy multiple Docker images on the same host to support both active and passive sensors or to manage multiple capture interfaces. For example, use this setup to receive several ERSPANs. Each passive container has one capture interface.
- Each container consumes one deployment token, even if they run on the same host.
- The persistent directory is /data instead of /iox_data.
- eth0 is behind NAT.
- Use eth1 for Active Discovery (instead of eth2 on IOx).
- The deliverable is in the form of a Docker Compose file, which specifies a Docker image that is hosted on the Center. It comes with the correct networks, environment variables, volume, and so on.
- Containers have no default resource limitations and can consume all host resources. Use the commented lines in the compose.yml file to limit CPU and RAM usage.
- Active and Capture Interfaces that declared in the compose file must exist.
- Disk usage grows if there is a Center communication issue.

Cyber Vision Docker Sensor Architecture and Network Settings Recommendations

The Docker host must be able to communicate with the Cisco Cyber Vision Center on interface eth0. The Docker sensor must be able to communicate with the Cisco Cyber Vision Center:

- For a single interface, use eth0.
- For a dual interface, use eth1.

The host interface that reaches the center allows the sensor container to communicate with it.

The Cyber Vision Docker sensor deployment uses Docker Compose and leverages the Cyber Vision sensor ZTP feature. A Docker application pull requests a connection with the center. The first sensor connection that is established through the ZTP feature also requests a connection with the center.

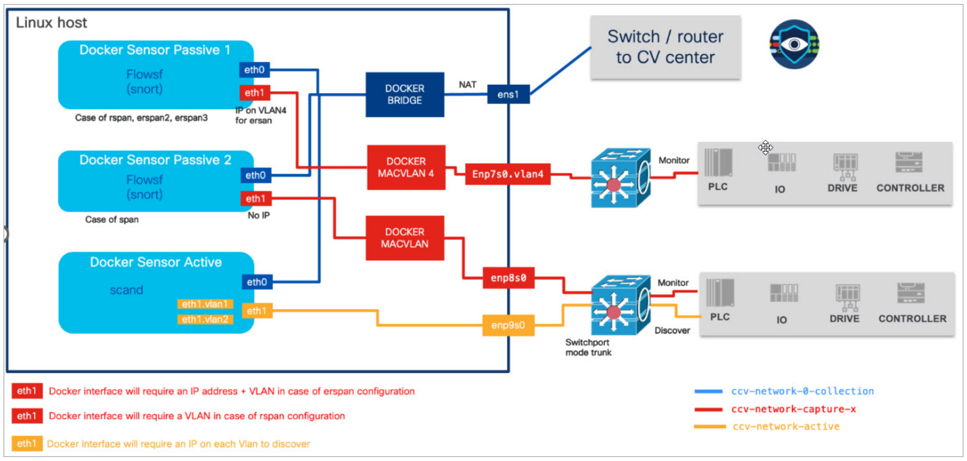
The sensor, once deployed, regularly sends results of DPI, NIDS, or Active Discovery to the center. It requires a valid communication path to the center.

The sensor has other interfaces that are described below:

- The passive container includes an extra interface connected to a Docker macvlan network in passthru mode, making it promiscuous with a host interface. This interface is used for data collection (SPAN, RSPAN, ERSPAN).
- The active container includes one or several extra interfaces, all connected to a Docker macvlan network in passthru mode, making them promiscuous with a host interface. These interfaces are used for the active discovery of OT networks.

The host networking equipment must handle "promiscuous mode," where one physical interface can have multiple MAC addresses. Not all physical network interfaces support this mode.

Basic schema of the network connections used by CV sensor docker container:



This is an example of a host with two passive Docker sensors (to monitor two different sources of traffic) and one active Docker sensor that can discover two VLANs.

CHAPTER 4

Known Issues

Known Issues

Docker reserves the first address of a defined network. Do not assign this first address when configuring the DPI interface in an Encapsulated Remote Switched Port Analyzer (ERSPAN).

CHAPTER 5

Initial Configuration

Docker Setup

Install Docker from the Docker repository. Use the following commands to install Docker on a fresh OS.

Procedure

Step 1 Uninstall all other packages.

```
for pkg in docker.io docker-doc docker-compose docker-compose-v2 podman-docker containerd runc; do
sudo apt-get remove $pkg; done
```

Step 2 Set up Docker's APT repository.

a) Add Docker's official GPG key.

```
sudo apt-get update

sudo apt-get install ca-certificates curl

sudo install -m 0755 -d /etc/apt/keyrings

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

sudo chmod a+r /etc/apt/keyrings/docker.asc
```

b) Add the repository to APT sources.

```
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]

https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update
```

Step 3 Install the Docker packages.

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

Note

Reboot the system to start Docker.

Step 4 Verify that the Docker Engine installation is successful by running the hello-world image using the command:

```
sudo docker run hello-world
```

This command downloads a test image and runs it in a container. When the container runs, it prints a confirmation message and exits.

Cisco Cyber Vision Docker Sensor Host Configuration

Docker Registry

Docker uses registries to pull container images. It classifies a registry as either secure or insecure.

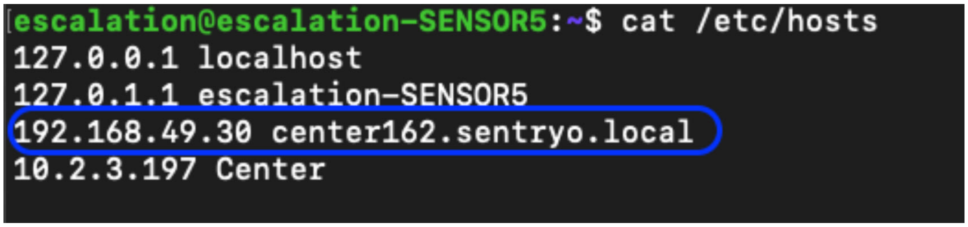
- A secure registry uses TLS and places a copy of its CA certificate on the Docker host at /etc/docker/certs.d/registry-FQDN:443/ca.crt.
- An insecure registry does not use TLS (listens on plaintext HTTP) or uses TLS with a CA certificate that is not recognized by the Docker daemon.

This issue occurs when the certificate is missing in /etc/docker/certs.d/registry-FQDN:443/ or when certificate verification fails due to an incorrect CA.

By default, Docker assumes all registries are secure, except for local ones. If Docker assumes a registry is secure, communication with an insecure registry fails. Configure the Docker daemon specifically to communicate with an insecure registry.

Docker Registry Secure Configuration

1. **FQDN**: Ensure that the host resolves the Cyber Vision Center FQDN. If the host cannot resolve the Center FQDN, specify the correct IP address in the host's configuration file.



```
escalation@escalation-SENSOR5:~$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 escalation-SENSOR5
192.168.49.30 center162.sentryo.local
10.2.3.197 Center
```

2. **Certificate**: If the Cyber Vision Center and the host share the same Certificate Authority, you do not need additional configuration. Otherwise, add the Center certificate to a specific folder on the host for authentication.

   Download the ca.pem file of your Center, rename it as ca.crt, and copy it into the folder **/etc/docker/certs.d/Center FQDN:443/**.

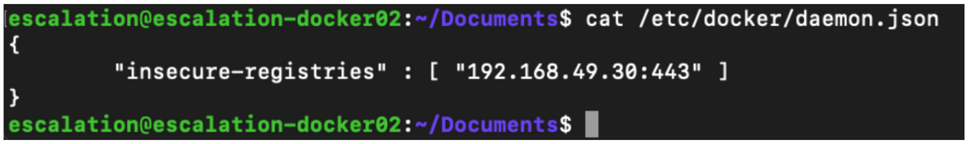
   Create a folder with the Center FQDN + ‘:443’, for example, center162.sentryo.local:443, and add the ca.crt file to it.

/etc/docker/certs.d/center162.sentryo.local:443/ca.crt

Docker Registry Insecure Configuration

Use the Center IP directly without FQDN resolution. Define the Center IP as an insecure registry in the Docker configuration by adding it to **/etc/docker/daemon.json**. Restart Docker with **sudo systemctl restart docker.service**.

For example:



```
escalation@escalation-docker02:~/Documents$ cat /etc/docker/daemon.json
{
        "insecure-registries" : [ "192.168.49.30:443" ]
}
escalation@escalation-docker02:~/Documents$ 
```

Configuration example:

```
{

        "insecure-registries" : ["192.168.49.30:443"]

}
```

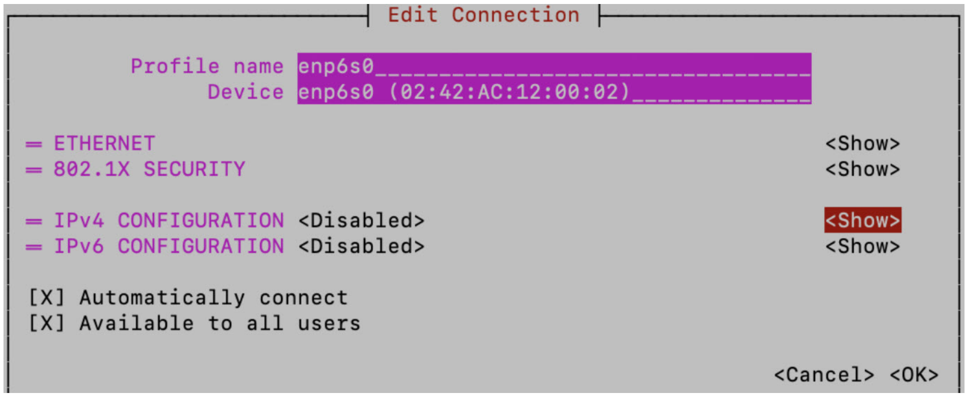
Other Host Configurations

1. **Network Configuration**: Disable IPv4 and IPv6 on all interfaces that are used for capturing traffic or performing active discovery.

   ```
   ipv4.method: "disabled"
   ipv6.method: "disabled"
   ```

   Use nmtui (Network Manager Text User Interface) to configure each interface for the Docker sensor application that is used in passive monitoring and active discovery. Launch the tool by typing: sudo nmtui, then use the graphical interface to disable IPv4 and IPv6 on all sensor interfaces.

   For example:

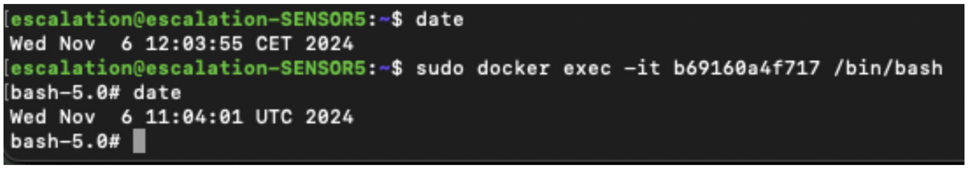
   ```
                              ┤ Edit Connection ├

             Profile name enp6s0________________________________________
                   Device enp6s0 (02:42:AC:12:00:02)___________________

    ═ ETHERNET                                                      <Show>
    ═ 802.1X SECURITY                                               <Show>

    ═ IPv4 CONFIGURATION <Disabled>                                 <Show>
    ═ IPv6 CONFIGURATION <Disabled>                                 <Show>

    [X] Automatically connect
    [X] Available to all users

                                                              <Cancel> <OK>
   ```

2. **Time Zone**: Set the host time zone to UTC using the command 'sudo timedatectl set-timezone UTC'. Alternatively, set it to any other time zone with a valid source of synchronization. The system requires a valid NTP server.

```
sudo nano /etc/systemd/timesyncd.conf (add at the end NTP=valid ntp server)
```

```
sudo systemctl restart systemd-timesyncd
```

For example:



```
escalation@escalation-SENSOR5:~$ date
Wed Nov  6 12:03:55 CET 2024
escalation@escalation-SENSOR5:~$ sudo docker exec -it b69160a4f717 /bin/bash
bash-5.0# date
Wed Nov  6 11:04:01 UTC 2024
bash-5.0#
```

CHAPTER 6

Set Up Cisco Cyber Vision Docker Sensor

Cisco Cyber Vision Docker Compose Creation

The Cisco Cyber Vision Docker Sensor deployment requires two steps:

1. Compose file generation.
2. Container creation.

Procedure

Step 1 If it's not already done, create some new deployment tokens.

See Create Deployment Tokens from Cisco Cyber Vision Administration Guide.

Step 2 From the main menu, choose **Admin** > **Sensors** > **Sensor Explorer**.

Step 3 Click **New sensor**.

Step 4 Click **Docker sensors** from the drop-down list.

Step 5 Fill in the details in the **Sensor Application** page.

a) Enter the name of your sensors in the **Name** field.

Note
The name of your sensors, treated as a Serial Number, must be unique for each center.

b) Click the drop-down arrow in the **Deployment token** field and select your deployment token name.

c) Click the drop-down arrow in the **Sensor Mode** field and select your Sensor Mode: **Passive or Active Discovery**.

d) Check the **Use insecure pull mode** check box.

The center IP address replaces the center FQDN.

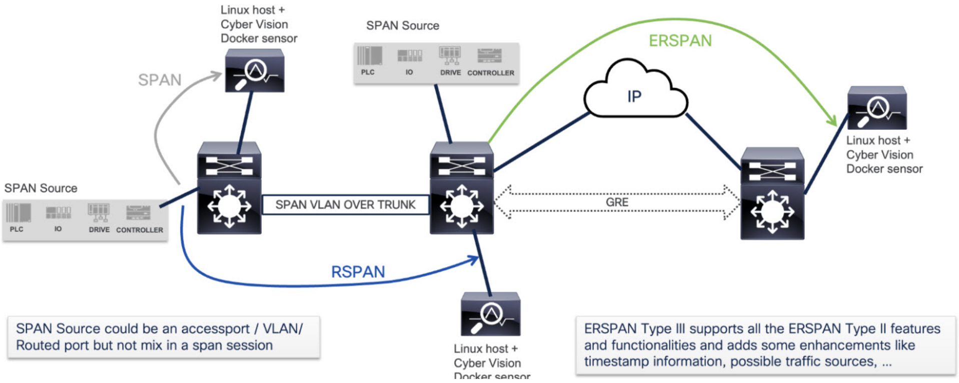
e) Check the **Center is behind NAT** checkbox if applicable to add another Center IP address.

Step 6 Click **Next**.

The **Capture Configuration** page appears.

Step 7 Fill out the **Capture Configuration** form with the capture interface configuration.

Note
The form changes based on the mirrored traffic type: SPAN, RSPAN, ERSPAN2, or ERSPAN3.


Linux host + Cyber Vision Docker sensor
SPAN
SPAN Source
PLC
IO
DRIVE
CONTROLLER
SPAN Source
PLC
IO
DRIVE
CONTROLLER
ERSPAN
IP
Linux host + Cyber Vision Docker sensor
SPAN VLAN OVER TRUNK
GRE
RSPAN
Linux host + Cyber Vision Docker sensor
SPAN Source could be an accessport / VLAN/ Routed port but not mix in a span session
ERSPAN Type III supports all the ERSPAN Type II features and functionalities and adds some enhancements like timestamp information, possible traffic sources, ...

a) For SPAN configuration:

- Click the drop-down arrow and select the **Mirrored traffic type** from the list.
- Enter the host interface in the **Capture Interface** field that will receive the capture (for example, eth2, enp7s0).
- Click the drop-down arrow of the **Capture Mode** field and select sensor filter (for example, **Optimal (default)**, **All**, **Industrial**, **Custom**). See Set a Capture Mode.

b) For RSPAN configuration:

- Click the drop-down arrow and select the **Mirrored traffic type** from the list.
- Add the host interface in the **Capture Interface** field that will receive the capture (for example, eth2, enp7s0).
- Click the drop-down arrow of the **Capture Mode** field and select sensor filter (for example, **Optimal (default)**, **All**, **Industrial**, **Custom**). See Set a Capture Mode.
- Add a VLAN ID configured for the SPAN destination.

c) For ERSPAN configuration:

- Click the drop-down arrow and select the **Mirrored traffic type** from the list.
- Add the host interface in the **Capture Interface** field that will receive the capture (for example, eth2, enp7s0).
- Click the drop-down arrow of the **Capture Mode** field and select sensor filter (for example, **Optimal (default)**, **All**, **Industrial**, **Custom**). See Set a Capture Mode.
- Enter the Capture IP address of the interface in the **Capture IP** field that will receive the spanned traffic.
- Add a VLAN ID if needed.

Step 8 Click **Continue with interfaces**.

The **Active Discovery** page appears.

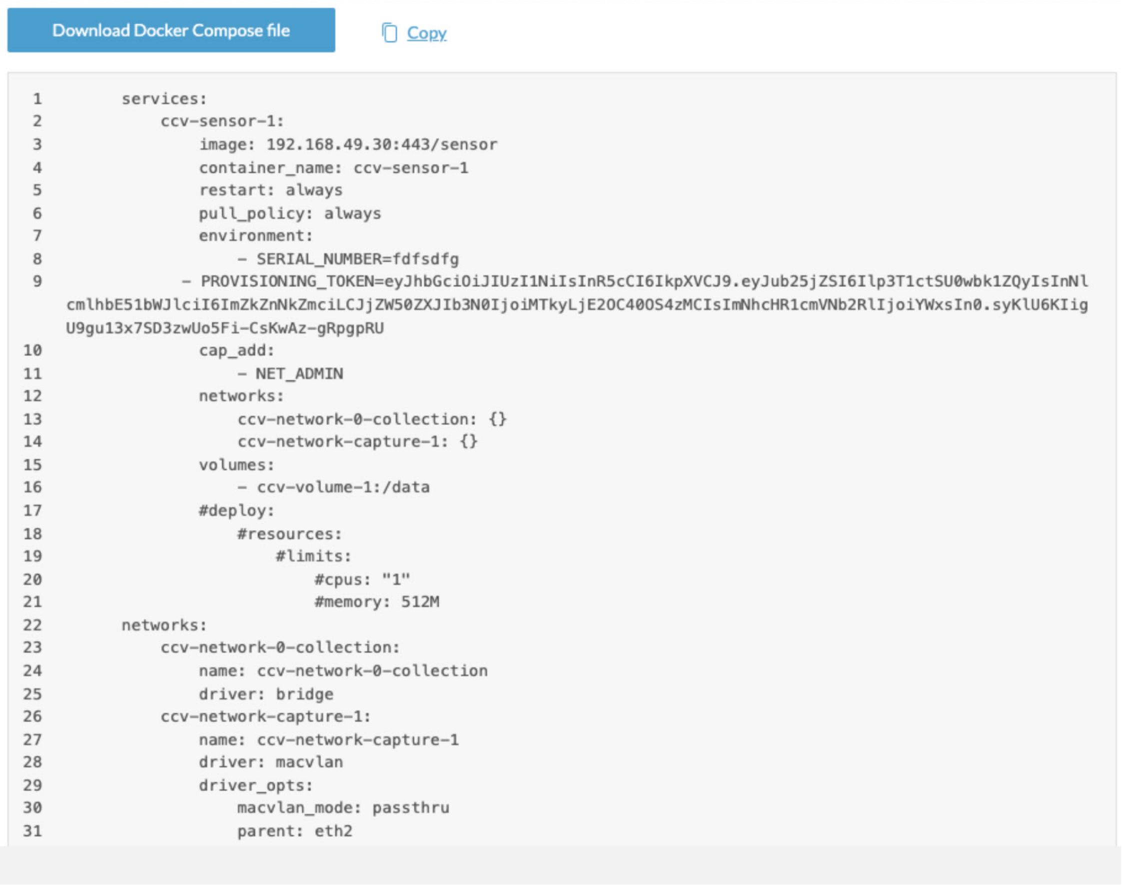
Step 9 Fill in the details on the **Active Discovery** page.

- Enter the host interface in the **Active Discovery Interface** field that will be used for active discovery (for example, eth2, enp7s0).
- Enter an IP address for active discovery in the **IP** field under **Target Interface**.
- If needed, add a **VLAN**.
- To add a new target, click **Add a new target**.

Step 10 Click **Continue with target interface**.

The **Docker Compose** page appears.

Step 11 The **Docker Compose** page provides the compose file needed to deploy sensor applications. Users need to download or copy it and add the file to their Linux system where Docker is running.



```
services:
    ccv-sensor-1:
        image: 192.168.49.30:443/sensor
        container_name: ccv-sensor-1
        restart: always
        pull_policy: always
        environment:
            - SERIAL_NUMBER=fdfsdfg
      - PROVISIONING_TOKEN=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJub25jZSI6Ilp3T1ctSU0wbk1ZQyIsInNlcmlhbE51bWJlciI6ImZkZnNkZmciLCJjZW50ZXJIb3N0IjoiMTkyLjE2OC40OS4zMCIsImNhcHR1cmVNb2RlIjoiYWxsIn0.syKlU6KIigU9gu13x7SD3zwUo5Fi-CsKwAz-gRpgpRU
        cap_add:
            - NET_ADMIN
        networks:
            ccv-network-0-collection: {}
            ccv-network-capture-1: {}
        volumes:
            - ccv-volume-1:/data
        #deploy:
            #resources:
                #limits:
                    #cpus: "1"
                    #memory: 512M
networks:
    ccv-network-0-collection:
        name: ccv-network-0-collection
        driver: bridge
    ccv-network-capture-1:
        name: ccv-network-capture-1
        driver: macvlan
        driver_opts:
            macvlan_mode: passthru
            parent: eth2
```

Cisco Cyber Vision Docker Container Deployment

After creating the compose file on the Linux system, use the command `docker compose` to launch and create the container.

Execute the command from the folder with the compose file named **compose.yml**. This way, you do not need to specify the file name in the command. Use the option -f if the file has a different name or is in another folder (for example, `-f compose2.yml`).

- To test the creation, use the command: `sudo docker compose up`. The system displays container logs. Press Ctrl+C to stop the container.
- To create and launch the container while leaving it running, use: `sudo docker compose up -d`.
- To force the recreation of the container, use: `sudo docker compose -f compose3.yml up --force-recreate -d`.

Once launched, the command creates the sensor container and configures it.

1. The system pulls the sensor application from the Center.
2. It configures the sensor application by creating networks and volumes.
3. It adds a provisioning token to the application.
4. When the container starts, the provisioning token creates the sensor on the Center side.
5. The Center sends the enrollment package to provide the sensor with its configuration and all necessary elements for communication, including certificates.

The sensor appears as **Connected** on the Cisco Cyber Vision Center's sensor list if the network configuration is correct. To see the sensor, select **Admin** > **Sensors** > **Sensor Explorer** from the main menu.

Cisco Cyber Vision Docker Sensor Additional Commands

- **sudo docker ps**: lists Docker containers
- **sudo docker exec -it ccv-sensor-1 /bin/bash**: accesses the command line of container ccv-sensor-1
- **sudo docker stop <sensor-name>**: stops the sensor container
- **sudo docker ps –a**: lists all containers, including stopped ones
- **sudo docker start <sensor-name>**: starts a sensor container
- **sudo docker rm -fv <agent-name>**: removes a sensor container
- **sudo docker compose logs**: displays Docker Compose logs

To delete a running container and its volume, use the appropriate command.

- **sudo docker rm -f ccv-sensor-1**: stops and kills the container
- **sudo docker volume ls**: lists the different volumes on your system
- **sudo docker volume rm documents_ccv-volume-1**: removes a specific volume
- **sudo docker images**: lists all images available on your system
- **sudo docker rmi [Image]**: deletes one selected image
- **sudo docker system prune -a**: removes unused images, containers, volumes, and networks

To completely clean up the system of all running images, containers, volumes, and networks, use the appropriate command:

- **sudo docker stop $(sudo docker ps -a -q)**
- **sudo docker system prune -a**
- **sudo docker volume prune -a**

CHAPTER 7

Configuration

Configure Active Discovery in Docker Sensor

Once the sensor connects, you can modify the Active Discovery network interface configuration to:

- Add multiple network interfaces for the sensor to perform Active Discovery on several subnetworks simultaneously.
- Delete an Active Discovery network interface.
- Change the configuration of an Active Discovery network interface.

Procedure

Step 1 From the main menu, choose **Admin** > **Sensors** > **Sensor Explorer**.

Step 2 Click the sensor to configure and click the **Active Discovery** button on its right side panel.

The **ACTIVE DISCOVERY CONFIGURATION** appears with the interface currently set.

Step 3 To add a new network interface, click **New network interface**.

Step 4 Enter the following parameters to set dedicated network interfaces:

- IP address
- Prefix length
- VLAN number

Step 5 Click **Add**.

You can add as many network interfaces as needed.

Step 6 Click **Configure**.

A message indicates that the configuration applies successfully.

Configure Sensor Configuration Template

Templates

This page allows you to create and set templates with protocol configurations and assign them to specific sensors.

Sensor templates contain protocol configurations which allow you:

- To enable or disable protocol DPI (Deep Packet Inspection) engines.
- To map UDP and TCP ports for each protocol's packet received by the sensor.

Enable or disable a protocol DPI engine to choose which protocols to analyze.

Disable a protocol DPI engine to avoid false positives in Cisco Cyber Vision. This occurs when a protocol appears on the user interface but is not present because the same UDP/TCP ports can be used by other non-standardized protocols.

The Default template disables some protocols because they are not commonly used or are specific to fields like transportation. The Default template applies to all compatible sensors.

Although UDP/TCP port configurations are mostly standardized, conflicts still occur with field-specific or with limited usage. Map UDP/TCP port numbers to ensure packets are sent to the correct DPI engine for accurate analysis and representation in the user interface.

Sending the protocol's packet to the wrong port results in related information appearing in Security Insights/Flows without a tag.

A sensor associates with only one template. Template deployment fails

- if the sensor is disconnected,
- if there is connection issues, or
- if the sensor version is too old.

Create Templates

Procedure

Step 1 From the main menu, choose **Admin** > **Sensors** > **Templates**.

Step 2 Click the **Add sensor template** button.

The **CREATE SENSOR TEMPLATE** window appears.

Step 3 Add a name to the template.

(Optional) You can add a description.

Step 4 Click **Next**.

The list of protocol DPI engines with their basic configurations appears.

Step 5 In the search bar, type the protocol you want to configure.

Step 6 To edit its settings, click the **pen** icon under the **Port Mapping** column, .

The protocol's port mapping window appears.

Step 7 Enter the port numbers you want to add.

Note
If you have continuous port numbers, you can enter a port range. For example, type 15000-15003 for ports 15000, 15001, 15002, and 15003.

Step 8 Click **OK**.

The port number is added to the protocol's default settings.

Step 9 Enable the toggle button **Displayed modified only** to quickly find the protocol.

Step 10 Click **Next**.

Step 11 Select the checkboxes for the sensors to which you want to apply the template.

Step 12 Click **Next**.

Step 13 Check the template configurations and click **Confirm**.

The configuration is sent to the sensors. Configuration deployment will take a few moments.

The OPCUA template appears in the template list with its two assigned sensors.

Export Templates

You can use this feature to define the template at one center and then migrate it to another. To export the template, follow these steps:

Procedure

Step 1 From the main menu, choose **Admin** > **Sensors** > **Templates**.

Step 2 Locate the template and hover over the ellipsis (…) in the **Actions** column.

Step 3 Click **Export** from the drop-down list.

Your system downloads the template to its local location.

Import Templates

To import the template, follow these steps:

Procedure

Step 1 From the main menu, choose **Admin** > **Sensors** > **Templates**.

Step 2 Click **Import sensor template**.

The system's local folder will opens.

Step 3 Select the template and click **Open**.

The system displays the imported template on the **Configuration Template** page.

Step 4 Locate the template and hover over the ellipsis (…) in the **Actions** column.

Step 5 Click **Edit** from the dropdown list.

Step 6 From the **Select sensors** tab, check the checkboxes of the sensors to which you want to apply the template.

Step 7 Click **Next**.

Step 8 Check the details and click **Update**.

The template recovers all the changes made in the previous center, and will be applied to the selected sensors.

Set a Capture Mode

The Capture Mode feature allows you to select which network communications will be analyzed by the sensors. To access the Capture Mode feature, follow these steps:

1. From the main menu, choose **Admin** > **Sensors** > **Sensor Explorer**.
2. Click the name of the sensor from the label column.

   The right side panel appears with the sensor details.
3. Click **Capture mode**.

   The **CAPTURE MODE** window appears.
4. Click the radio button to select **Capture Mode**.

The aim is mainly to focus the monitoring on relevant traffic but also to reduce the load on the Center.

For example, a common filter in a firewall can consist of removing the network management flows (SNMP). This can be done by setting a filter like "not (port 161 and host 10.10.10.10)" where "10.10.10.10" is the network management platform.

By using Capture Mode, Cisco Cyber Vision performance can be improved on large networks.

Capture modes operate because of filters applied on each sensor. Filters are set to define which types of incoming packets are to be analyzed by the sensors. You can set a different filter on each sensor according to your needs.

You can set the capture mode in the installation wizard when enrolling the sensors during the Center installation. This option is recommended if you already know which filter to set. Otherwise, you can change it at any time on the Sensor Explorer page in the GUI (provided that the SSH connection is allowed from the Center to the sensors).

The different capture modes are:

- **ALL**: The sensor analyzes all incoming flows without applying a filter. It stores all flows in the Center database.
- **OPTIMAL (Default)**: The filter selects the most relevant flows based on Cisco Cyber Vision expertise. It does not record multicast flows. Use this capture mode for long-term capture and monitoring.
- **INDUSTRIAL ONLY**: The filter selects only industrial protocols like Modbus, S7, and EtherNet/IP. This means that the sensor does not analyze IT flows of the monitored network, and they do not appear in the GUI.
- **CUSTOM (advanced users)**: Use this capture mode to fully customize the filter. Use the tcpdump syntax to define the filtering rules.

CHAPTER 8

Maintenance

Sensor Self Update

Cisco Cyber Vision now allows sensor updates regardless of the installation method (for example, without the extension) and provides the necessary foundation for sensor self-updates. However, the self-update feature will only be functional in future releases. You can update all sensors automatically. The required steps are:

- Select sensors to update.
- The Center adds a new job to the sensor queue.
- The sensor automatically collects and validates the update file.
- The sensor restarts with the new version.

Update Warnings

In the Cisco Cyber Vision Center on the Sensor Explorer page, you receive an alert to update the sensor. When this occurs, the latest version number appears in red, and a blue arrow with a tooltip indicates the sensor is upgradeable.

To update the senosr, follow thses steps:

- From the main menu, choose **Admin** > **Sensors** > **Sensor Explorer**.
- Click the sensor that is upgradeble from the **Label** column.
- The right side panel appears with sensor details.
- Click **Update**.

Update Procedure

Procedure

Step 1 From the main menu, choose **Admin** > **Senors** > **Sensor Explorer**.

Step 2 Check the checkboxes to select multiple sensors.

Step 3 Click the drop-down arrow of the **More Actions** button.

Step 4 Click **Update sensors** from the drop-down list.

The **UPDATE SENSORS** pop-up appears.

Step 5 Click **OK**.

During the update, a blue circle appears in the **Update status** column. After the update is complete, the version number turns black, and a green symbol appears in the same column.

Update Failure

If the update is unsuccessful, the **Update Status** column displays a red cross and a detailed message. To view the failure message, choose **Admin** > **Sensors** > **Sensor Explorer** from the main menu. Hover over the red cross in the **Update Status** column to see the details of the update failure.

CHAPTER 9

Troubleshooting

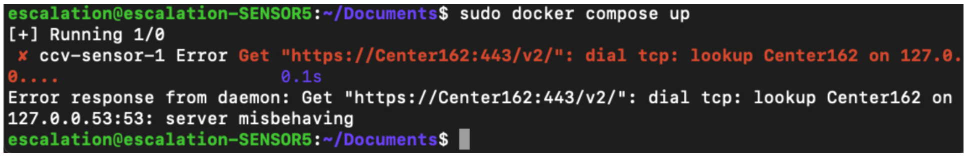
Docker Pull Issues

Default Secure Configuration

By default, the system pulls the Docker image using the Center's FQDN with TLS certificate verification. Configure two settings in this scenario:

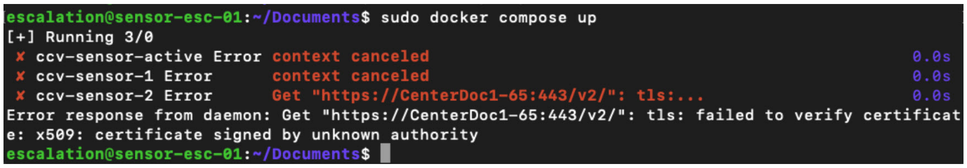
1. **Name Resolution**: Resolve the FQDN through a reachable DNS server or a local configuration.
2. **Certificate**: Ensure the Center and Docker access the same certificate authority, or manually share the Center's certificate.

Observe name resolution issues during Docker container creation.



```
escalation@escalation-SENSOR5:~/Documents$ sudo docker compose up
[+] Running 1/0
 ✘ ccv-sensor-1 Error Get "https://Center162:443/v2/": dial tcp: lookup Center162 on 127.0.
0....                      0.1s
Error response from daemon: Get "https://Center162:443/v2/": dial tcp: lookup Center162 on
127.0.0.53:53: server misbehaving
escalation@escalation-SENSOR5:~/Documents$ 
```

If there is a certificate issue, the following problem can occur:



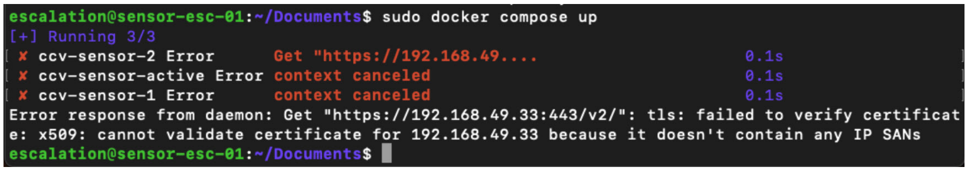
```
escalation@sensor-esc-01:~/Documents$ sudo docker compose up
[+] Running 3/0
 ✘ ccv-sensor-active Error context canceled                                       0.0s
 ✘ ccv-sensor-1 Error      context canceled                                       0.0s
 ✘ ccv-sensor-2 Error      Get "https://CenterDoc1-65:443/v2/": tls:...           0.0s
Error response from daemon: Get "https://CenterDoc1-65:443/v2/": tls: failed to verify certificat
e: x509: certificate signed by unknown authority
escalation@sensor-esc-01:~/Documents$ 
```

This means that the certificate found is incorrect, or the folder containing the local certificate does not have the correct name.

Default Insecure Configuration

If the Center's FQDN and TLS certificate verification are not usable, pull the Docker image using the Center's IP address directly. Add an exception on the Docker host to allow insecure registry usage.

If the option to allow insecure registry usage is not properly configured, the following message appears:

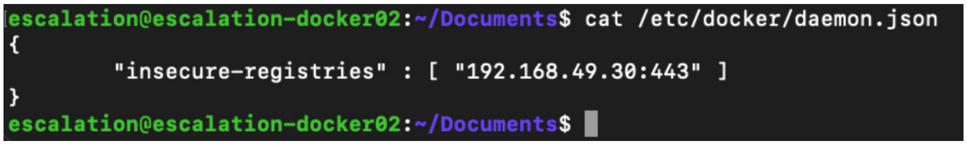


```
escalation@sensor-esc-01:~/Documents$ sudo docker compose up
[+] Running 3/3
 ✘ ccv-sensor-2 Error      Get "https://192.168.49....                   0.1s
 ✘ ccv-sensor-active Error context canceled                              0.1s
 ✘ ccv-sensor-1 Error      context canceled                              0.1s
Error response from daemon: Get "https://192.168.49.33:443/v2/": tls: failed to verify certificat
e: x509: cannot validate certificate for 192.168.49.33 because it doesn't contain any IP SANs
escalation@sensor-esc-01:~/Documents$
```

Error response from daemon: Get "https://192.168.49.30:443/v2/": tls: failed to verify certificate: x509: cannot validate certificate for 192.168.49.30 because it doesn't contain any IP SANs.

In this case, configure the system to allow insecure registry usage. Add the Center's IP to **/etc/docker/daemon.json** and restart Docker with `sudo systemctl restart docker.service`.

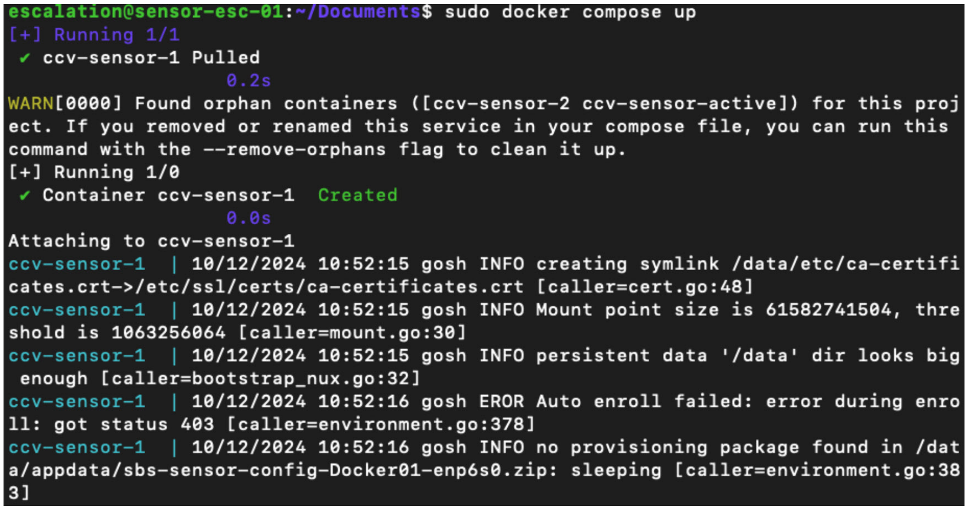
For example:



```
escalation@escalation-docker02:~/Documents$ cat /etc/docker/daemon.json
{
        "insecure-registries" : [ "192.168.49.30:443" ]
}
escalation@escalation-docker02:~/Documents$
```

Deployment Token Issue

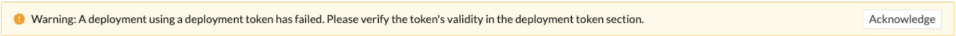
A deployment token issue prevents the sensor from starting. Observe the sensor logs in the Docker container during the sensor startup. For example:



```
escalation@sensor-esc-01:~/Documents$ sudo docker compose up
[+] Running 1/1
 ✔ ccv-sensor-1 Pulled
                   0.2s
WARN[0000] Found orphan containers ([ccv-sensor-2 ccv-sensor-active]) for this proj
ect. If you removed or renamed this service in your compose file, you can run this
command with the --remove-orphans flag to clean it up.
[+] Running 1/0
 ✔ Container ccv-sensor-1  Created
                   0.0s
Attaching to ccv-sensor-1
ccv-sensor-1  | 10/12/2024 10:52:15 gosh INFO creating symlink /data/etc/ca-certifi
cates.crt->/etc/ssl/certs/ca-certificates.crt [caller=cert.go:48]
ccv-sensor-1  | 10/12/2024 10:52:15 gosh INFO Mount point size is 61582741504, thre
shold is 1063256064 [caller=mount.go:30]
ccv-sensor-1  | 10/12/2024 10:52:15 gosh INFO persistent data '/data' dir looks big
 enough [caller=bootstrap_nux.go:32]
ccv-sensor-1  | 10/12/2024 10:52:16 gosh EROR Auto enroll failed: error during enro
ll: got status 403 [caller=environment.go:378]
ccv-sensor-1  | 10/12/2024 10:52:16 gosh INFO no provisioning package found in /dat
a/appdata/sbs-sensor-config-Docker01-enp6s0.zip: sleeping [caller=environment.go:38
3]
```

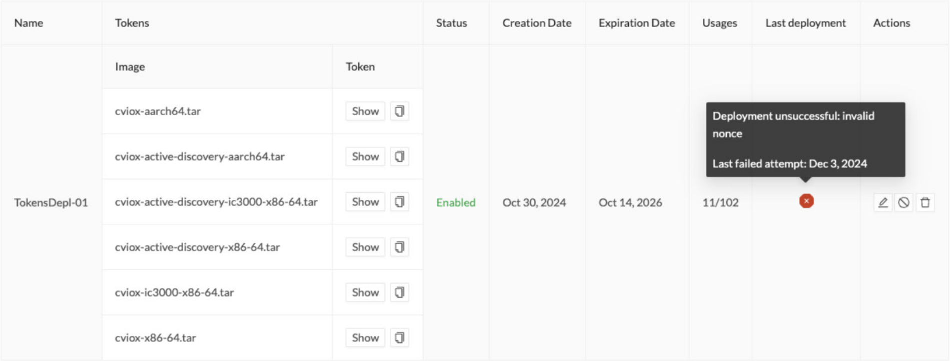
- ccv-sensor-1 | 10/12/2024 10:52:16 gosh EROR Auto enroll failed: error during enroll: got status 403 [caller=environment.go:378]
- ccv-sensor-1 | 10/12/2024 10:52:16 gosh INFO no provisioning package found in /data/appdata/sbs-sensor-config-Docker01-enp6s0.zip: sleeping [caller=environment.go:383]

In the Cisco Cyber Vision Center user interface, some messages can help with troubleshooting. The Sensor Explorer page displays a following message if the deployment token is no longer valid or has already been used.



Warning: A deployment using a deployment token has failed. Please verify the token's validity in the deployment token section.

The deployment token page identifies other causes. For example, a token that is already used displays the message: "invalid nonce."



Name	Tokens		Status	Creation Date	Expiration Date	Usages	Last deployment	Actions
	Image	Token						
TokensDepl-01	cviox-aarch64.tar	Show	Enabled	Oct 30, 2024	Oct 14, 2026	11/102	✕	
	cviox-active-discovery-aarch64.tar	Show						
	cviox-active-discovery-ic3000-x86-64.tar	Show						
	cviox-active-discovery-x86-64.tar	Show						
	cviox-ic3000-x86-64.tar	Show						
	cviox-x86-64.tar	Show						

Deployment unsuccessful: invalid nonce

Last failed attempt: Dec 3, 2024

Docker Issues with Ethernet Boards

Some Ethernet boards are not supported, as explained in the prerequisites. The Cisco Cyber Vision Docker sensor cannot use them as DPI interfaces because they do not support the applied Docker network configuration.

Cisco tests different vendors and recommends using Intel Ethernet adapters. It does not support USB adapters.

An incompatible Ethernet board produces error messages during Docker container creation, such as:

- Error response from daemon: failed to create task for container: failed to create shim task: OCI runtime create failed: runc create failed: unable to start container process: error during container init: error running hook #0: error running hook: exit status 1, stdout: , stderr: failed to add interface vethd56cba0 to sandbox: error setting interface "vethd56cba0" MAC to "02:42:ac:13:00:02": device or resource busy: unknown