Deployment Guide

Onboarding and Deploying CloudVision Sensor

Version 1.1

Headquarters	Support	Sales
5453 Great America Parkway Santa Clara, CA 95054 USA +1-408-547-5500	+1-408-547-5502 +1-866-476-0000	+1-408-547-5501 +1-866-497-0000
www.arista.com/en/	support@arista.com	sales@arista.com

Contents

Chapter 1

Overview

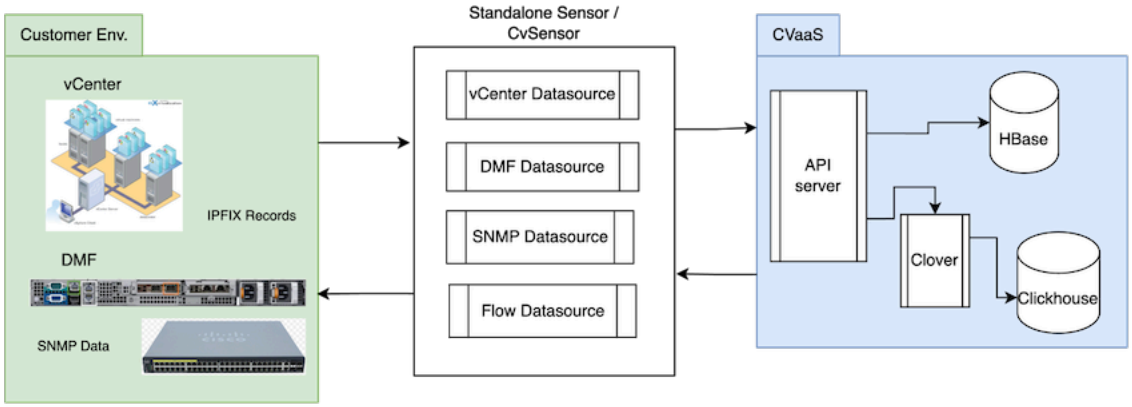
CloudVision® Universal Network Observability™ (CV UNO) is a multi-domain network observability platform that integrates application visibility with CloudVision's network telemetry. This integration helps provide insights into the applications and workload performance across data centers, campuses, and wide area networks.

CV UNO is available along with the CloudVision-as-a-Service (CVaaS) platform. It offers cloud-based onboarding and feature delivery, using secure state-streaming to an Arista-managed cloud-native architecture.

The CV sensor is an integral component of CV UNO. The sensor is a VM deployed on-premises and facilitates viewing application data in CloudVision. The sensor collects, normalizes, and curates flow and SNMP data from various data sources. It also polls data from vCenter and subscribes to vCenter events, allowing you to view them in CloudVision. This data gets forwarded to the EOS Network Data Lake (NetDL), where the network data lake combines diverse datasets and performs a machine-learning-based analysis. With the help of this data, CV UNO quickly determines the source of an anomaly, whether a network or application-based anomaly. If it is a network anomaly, CV UNO determines where the issue occurs and why.

The following image provides a high-level overview of the functionality of the CV Sensor:

Figure 1-1: CV Sensor Architecture



Familiarize with the following terminology in this document:

- CV Sensor - refers to the collector that streams the data from one or more data sources. The Sensor is responsible for starting different data sources, collecting third-party device data, and streaming it to CVP.
- Data Source - refers to the target device in the onboarding work flow. For example: vCenter, Flow, DMF, SNMP (Cisco router/switch).
  - vCenter Data source includes:

        - State Provider - Virtual Machines (VMs), Hosts, Distributed Virtual Switches (DVS), etc
        - Counters Provider - system counters, network counters, etc
        - Tags Provider - vCenter tags
        - Events Provider - vCenter events
    - DMF Data source includes: DMF Provider
    - SNMP Data sources include:
        - SNMP Provider: SNMP Walk for Fetching System, LLDP, and Interfaces Information.
    - Flow Data source includes:
        - IPFIX Provider
        - NetFlow Provider
        - sFlow Provider
- Provider - A worker or go-routine responsible for pulling or receiving a single type of data, and sending it to CVP. For example: State Provider, IPFIX Provider, DMF Provider, etc.

Chapter 2

Deploying the CV Sensor

To view data from external data sources in CloudVision, you must deploy the CV Sensor and on board it as a data source so that it can listen to external data sources. The CV Sensor is deployed as an OVA appliance and is intended to run on top of an ESXi server.

When you deploy the sensor using the sensor OVA, it generates a VM with the following specifications:

- Number of CPU cores: 12
- Memory: 16 Gibibytes (GiB)
- Disk Space: 124 Gibibytes (GiB)

Note: Ensure that your system/host has sufficient resources available to accommodate the sensor OVA deployment.

Note: You must also on board any external *data sources* to CloudVision so that the sensor can stream or poll them for their data.

To deploy the CV Sensor, follow the steps described here:

1. Generate a Service Account Token
2. Add the Sensor in the UI
3. Get the latest Sensor OVA
4. Deploy Sensor OVA
5. Boot up the Sensor
6. Add Data Source

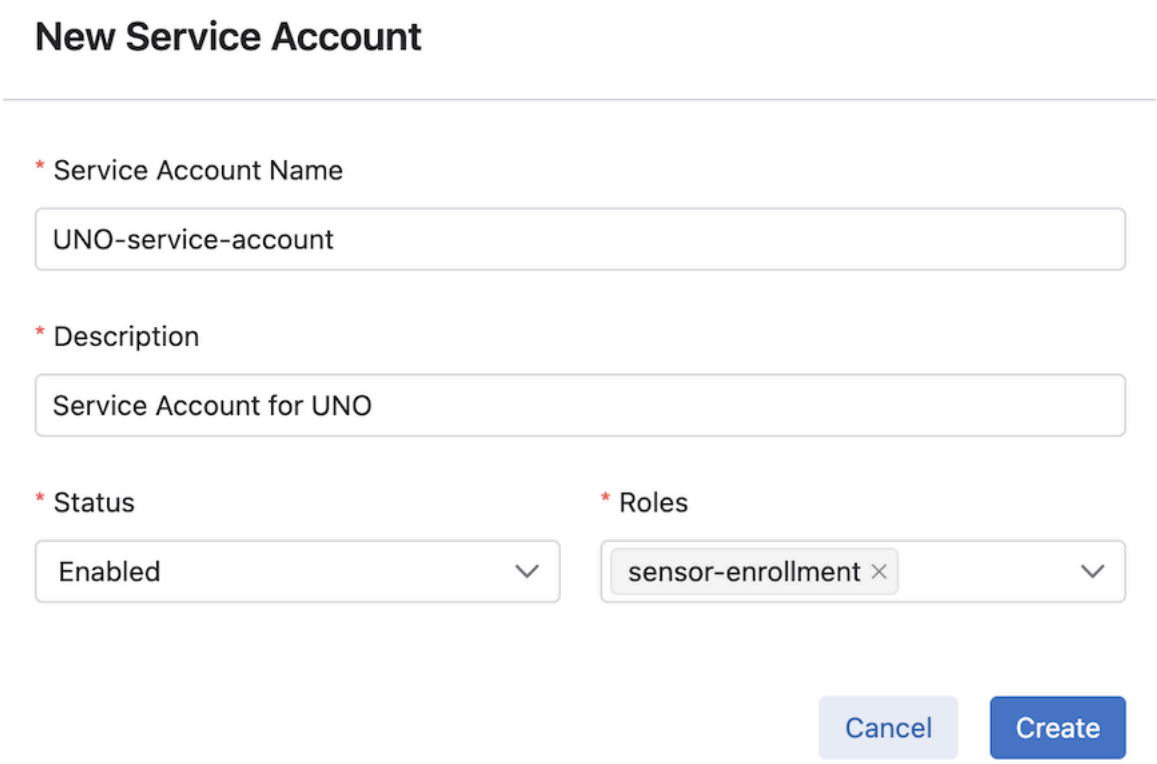
2.1 Generating a Service Account Token

To generate a service account token:

1. Login to CVaaS cluster using the URL - www.arista.io
2. Navigate to **Settings -> Access Control -> Service Accounts -> New Service Account.**
3. Create a new service account for UNO Sensor (see image):
   a. Service Account Name (example, UNO-service-account)
   b. Description
   c. Status: **Enabled**
   d. Roles: Select the pre-defined role **sensor-enrollment**.

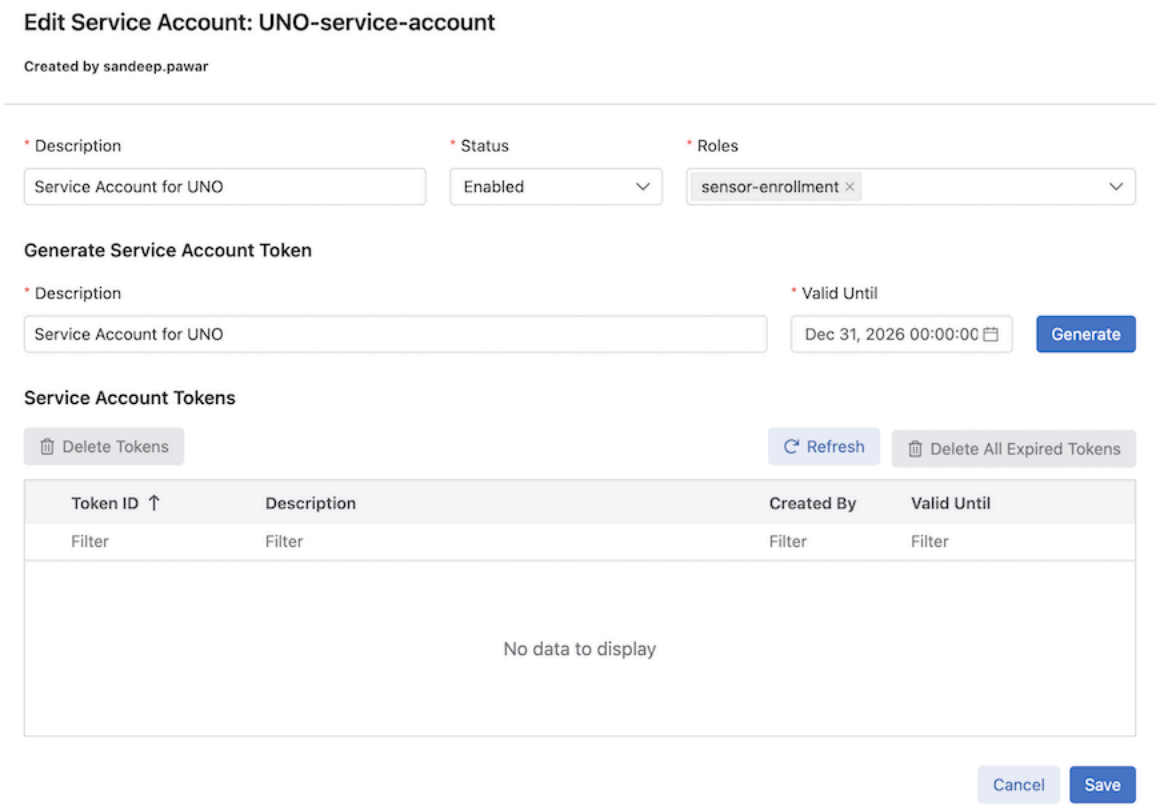
4. Click the **Create** button. The newly added account (UNO-service-account) appears in the list of Service Accounts.

Figure 2-1: New Service Account



5. Click on the newly created Service Account (UNO-service-account).
6. To generate the Service Account Token:
   a. Enter a **Description** and select a **Valid Until** field.
   b. Select an **expiry date** that is at least after a year from the current date.
   c. Click the **Generate** button.

Figure 2-2: Edit Service Account



Note: When the token is generated, copy and securely save it in a location where it can be accessed during sensor deployment.

2.2 Adding the Sensor to the UI

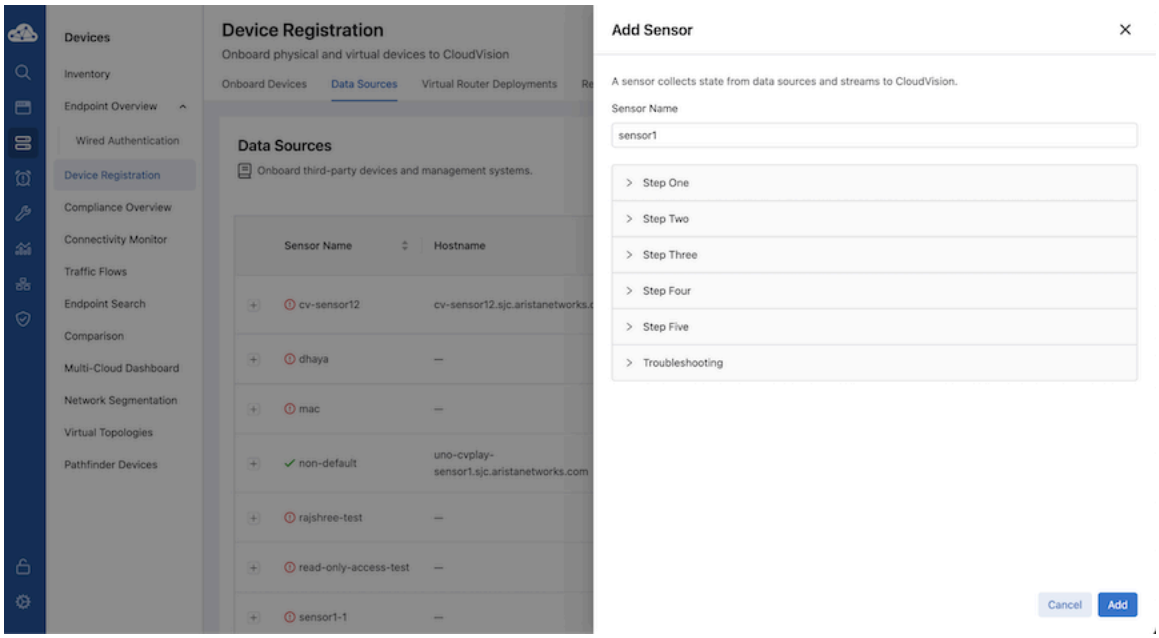
To add the sensor to the CVaaS UI:

1. Navigate to **Devices** -> **Device Registration** -> **Data Sources**
2. Click the **+ Add Sensor** button
3. Enter a desired sensor name (for example, **sensor1**). Make sure to use the same name while deploying the sensor.

   Note: Do not use *default* as the sensor name.
4. Click the **Add** button.

No additional information is required except for the Sensor Name.

Figure 2-3: Adding the Sensor



2.3 Getting the Latest Sensor OVA

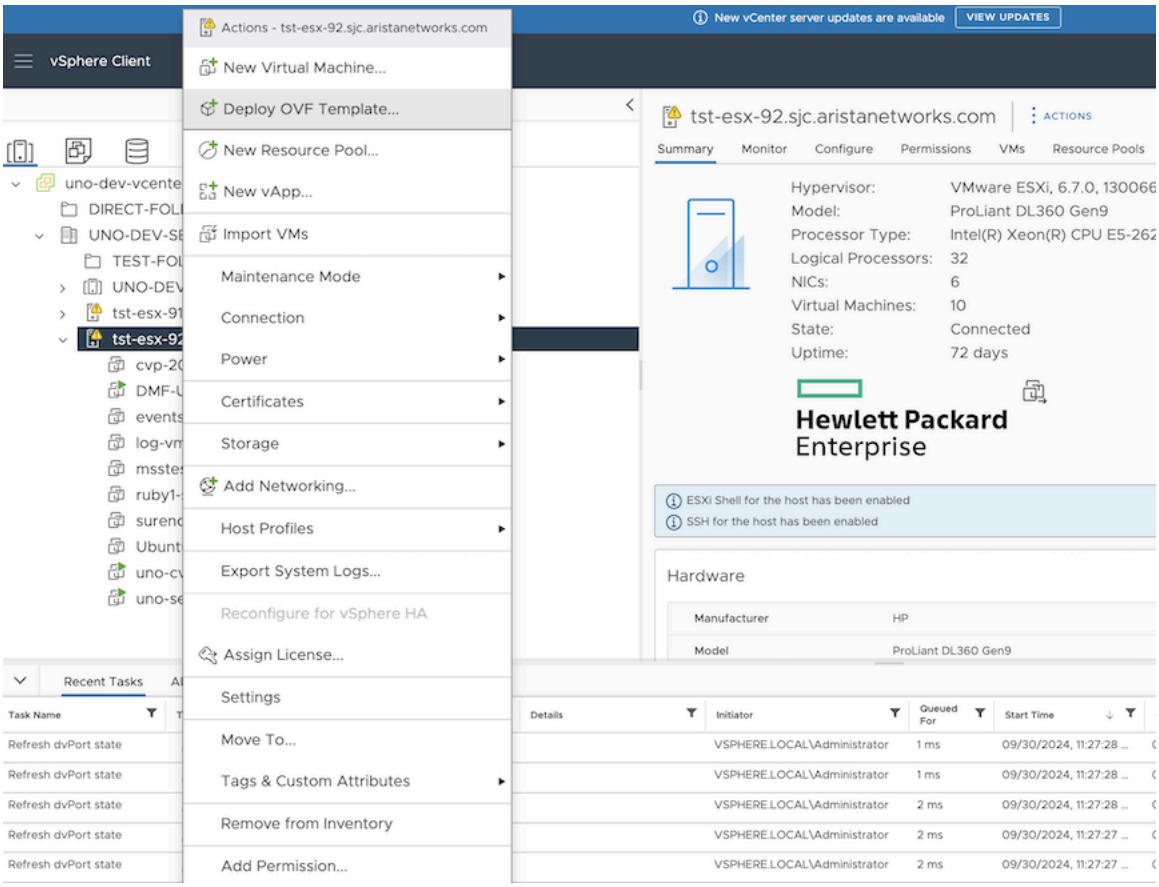
Download the UNO Sensor if you already have the OVA file or contact your Arista support representative for download instructions.

2.4 Deploying the Sensor OVA

To deploy the Sensor OVA:

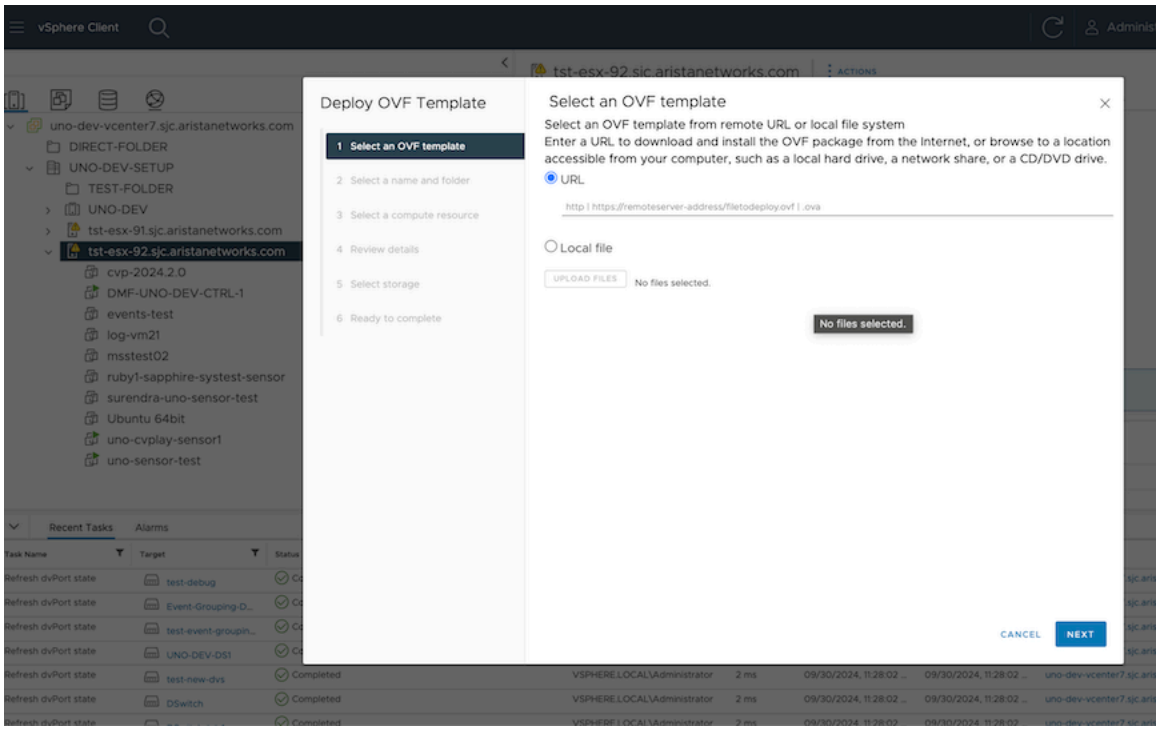
1. Navigate to the vCenter where you intend to deploy the sensor OVA. Right-click on the ESXi server.
2. Proceed to **Deploy OVF Template** and enter the URL of the latest Sensor OVA (see images below).

Figure 2-4: Deplying OVF Template



3. Specify the **VM name**, **datastore**, and other required details during the deployment (see image below).

Figure 2-5: Deploy OVF Template page



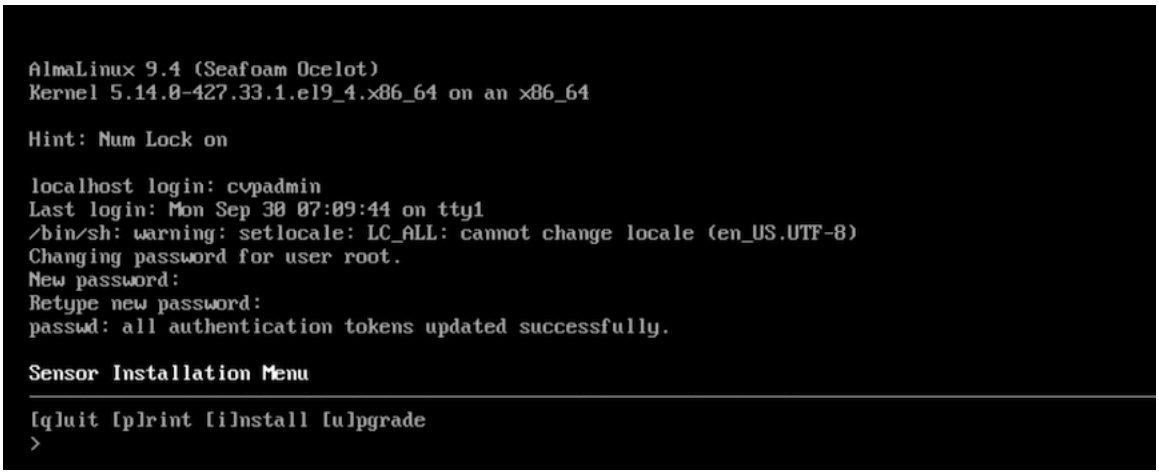
2.5 Booting up the Sensor

To boot up the Sensor for the first time after the Sensor deployment is completed:

1. Power on the VM and choose to **LAUNCH REMOTE** or **WEB CONSOLE**.
2. Log in using the credential:

   User name: **cvpadmin**
3. Set a password for the root user.
4. When the sensor installation menu is displayed, select the **install** option by typing "**i**" or "**install**" (case sensitive).

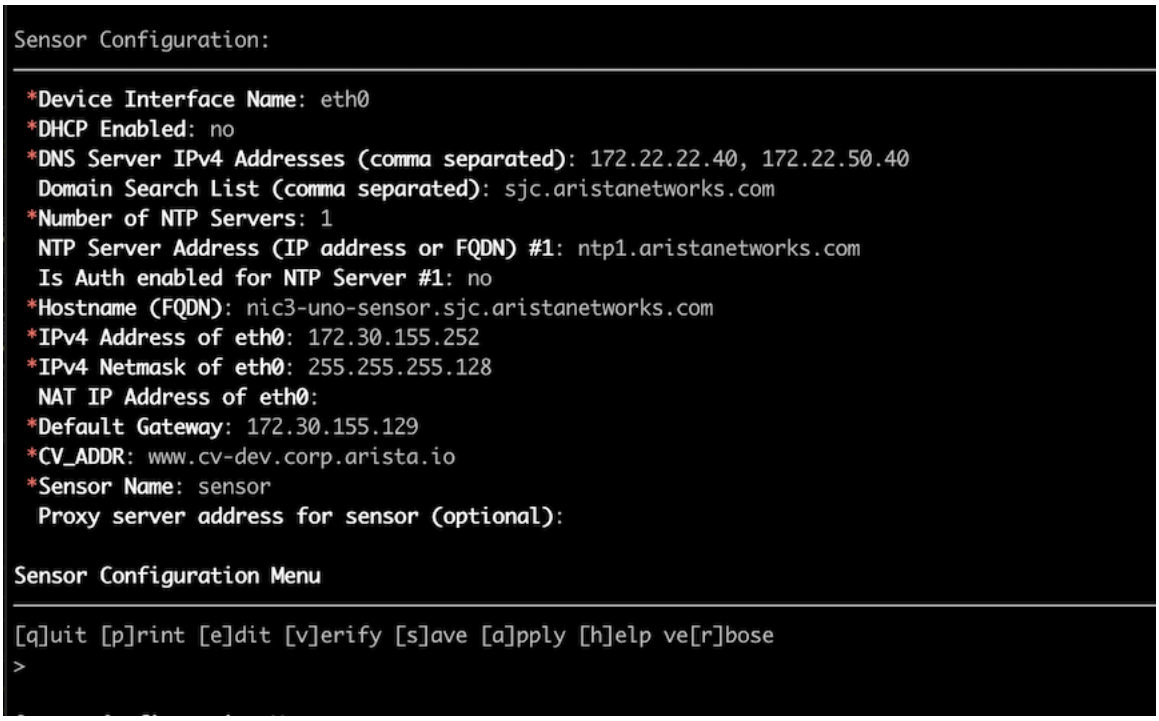
Figure 2-6: Installing Sensor from CLI



The initial configuration screen appears (see image).

5. Enter the following details:
   a. **IP Address of eth0**: Obtain the static IP from the vCenter administrator.
   b. **DNS Domain Search List**: Multiple entries can be added using a comma separator.
   c. **CV_ADDR**: This is a preconfigured field depending on the region, please refrain from making any changes in a production deployment. The expected URLS based on the regions are:
      1. United States 1a: *www.arista.io*
      2. United States 1c: *www.cv-prod-us-central1-c.arista.io*
      3. Japan: *www.cv-prod-apnortheast-1.arista.io*
      4. Germany: *www.cv-prod-euwest-2.arista.io*
      5. Australia: *www.cv-prod-ausoutheast-1.arista.io*
      6. Canada: *www.cv-prod-na-northeast1-b.arista.io*
      7. United Kingdom: *www.cv-prod-uk-1.arista.io*
   d. **Sensor Name**: Provide the same name used while adding the sensor on UI (For example, **sensor1**).
6. Verify the configuration by typing “**v**” or “**verify**.”

Figure 2-7: Sensor Configuration



7. Once verification is successful, apply the configuration by typing "**a**" or "**apply**".

   While the configuration is being applied, you are prompted to add the access_token in the file */etc/cvpi/access_token* as follows. The setup wizard waits for you to create this token file.

Figure 2-8: Sensor Configuration Menu

8. To add the token, SSH to the VM as the root user and use the token generated in Generating a Service Account Token section and enter it in the */etc/cvpi/access_token* file.
9. Copy the service account token and execute the following command on the sensor VM to set it:

```
echo "paste_token_here" > /etc/cvpi/access_token
```

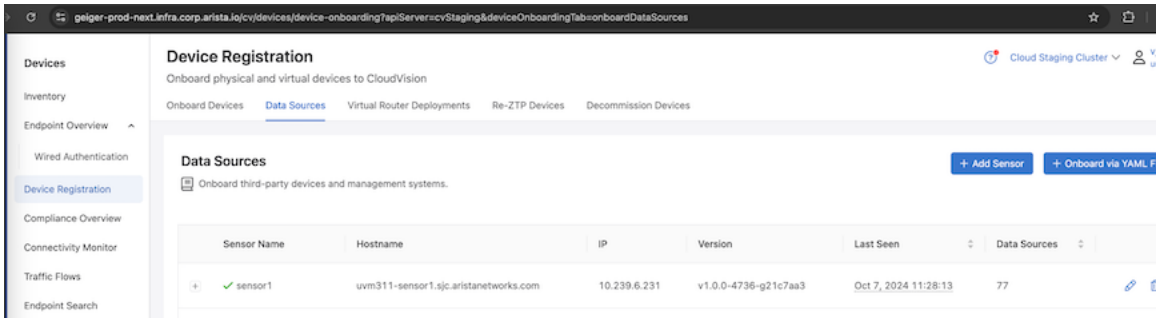
   The above command writes the copied token to the `/etc/cvpi/access_token` file on the Sensor VM. Once you create and save this token file, the setup wizard automatically proceeds with the installation process.

10. Type **s** or **save** to save the configuration.
11. Once the installation is successful, all the components, including the sensor, will be up and running.
12. Verify the status by **SSH**ing to the VM and by using the command:

```
cvpi status all -v3
```

For Sensor Streaming to CVaaS, the sensor name configured in earlier steps (sensor1) shows up with a green tick indicating that deployment of Sensor OVA is successful and the Sensor is able to communicate with CVaaS.

Figure 2-9: Sensor Deployed Successfully

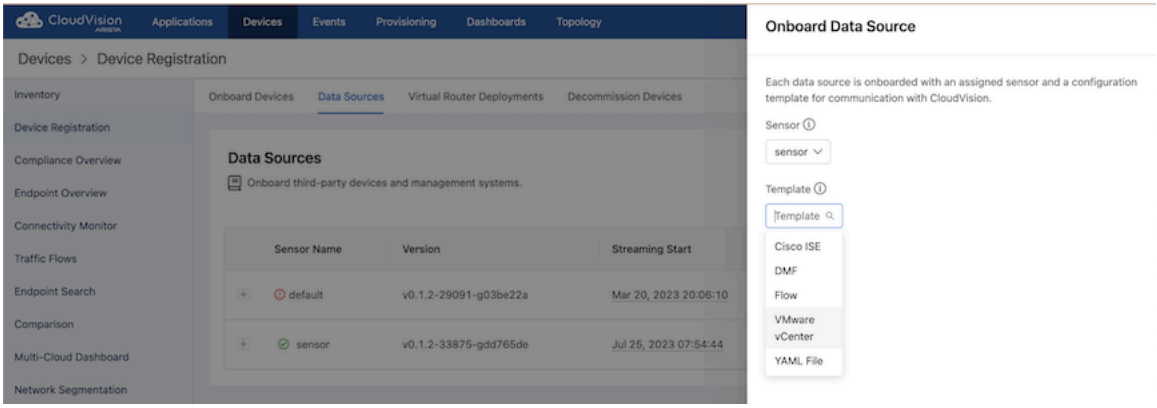


2.6 Adding the Data Sources

To add data sources:

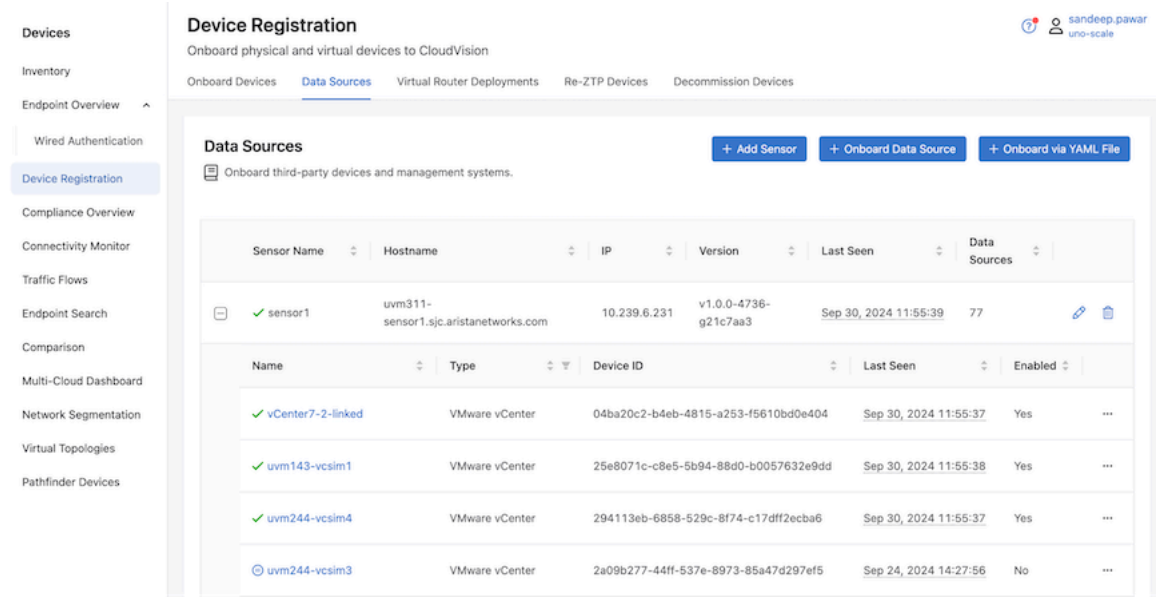
1. Go to **Network** -> **Device Registration** -> **Data Sources**
2. Click the **+ Onboard Data Source** button.
3. Choose the sensor from the drop-down list (for example, **sensor1**)
4. Select the required device type template, such as Cisco ISE, DMF, Flow, VMware vCenter, or YAML file.
5. Enter the necessary fields and click **Onboard** to add the data source.

Figure 2-10: Add Data Sources



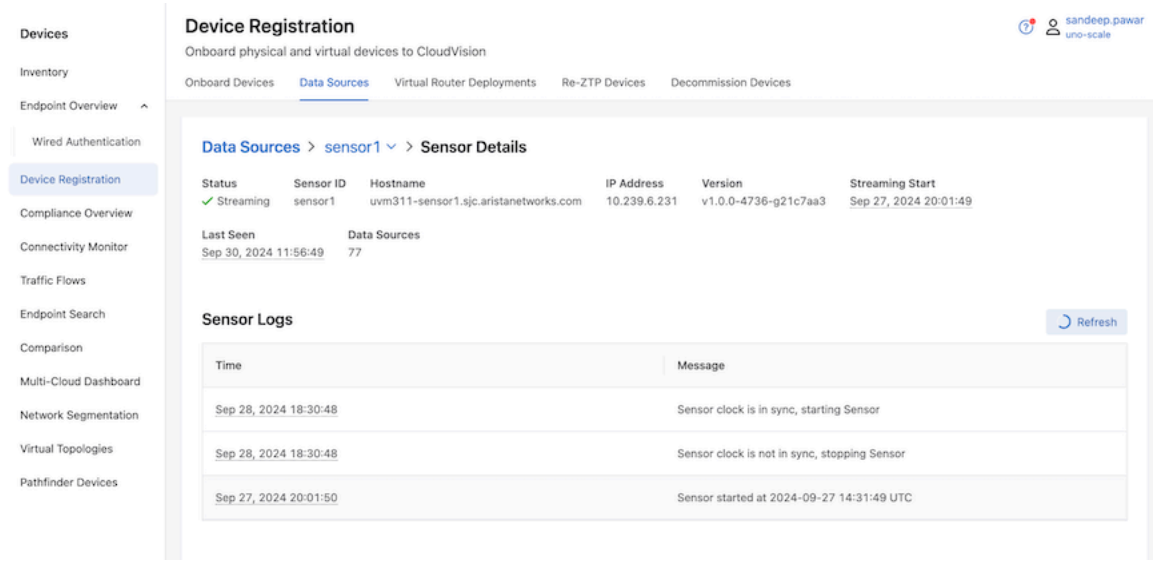
After adding the data sources, check if the data is streaming successfully. A green tick in front of each data source (under sensor1) indicates successful streaming, and a red mark indicates an issue with the streaming (see image below).

Figure 2-11: Device Registration - Data Sources



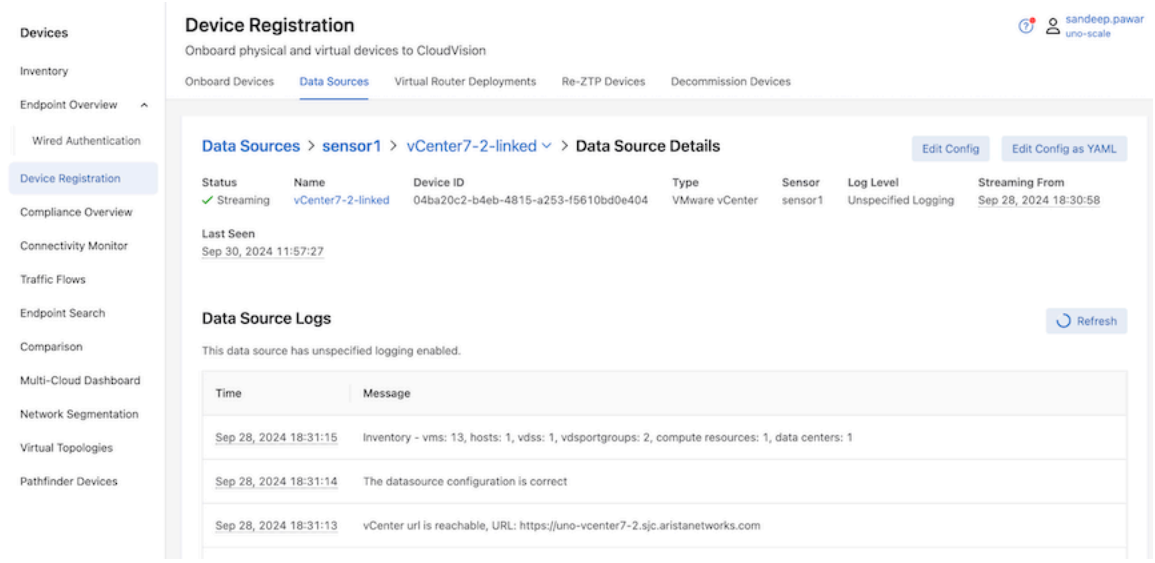
6. Click the sensor to access the streamed data source details and for any status message indicating if the sensor has started or there is an error message under *Sensor Details*.

Figure 2-12: Device Registration Data Sources



Similarly, click on each on-boarded data source to display the respective data source status messages (whether the data source has started or if there are any errors).

Figure 2-13: Data Source Details



Now, you can view the on-boarded data sources and confirm that data streaming has started.

2.6.1 Adding VMware vCenter as a Data Source

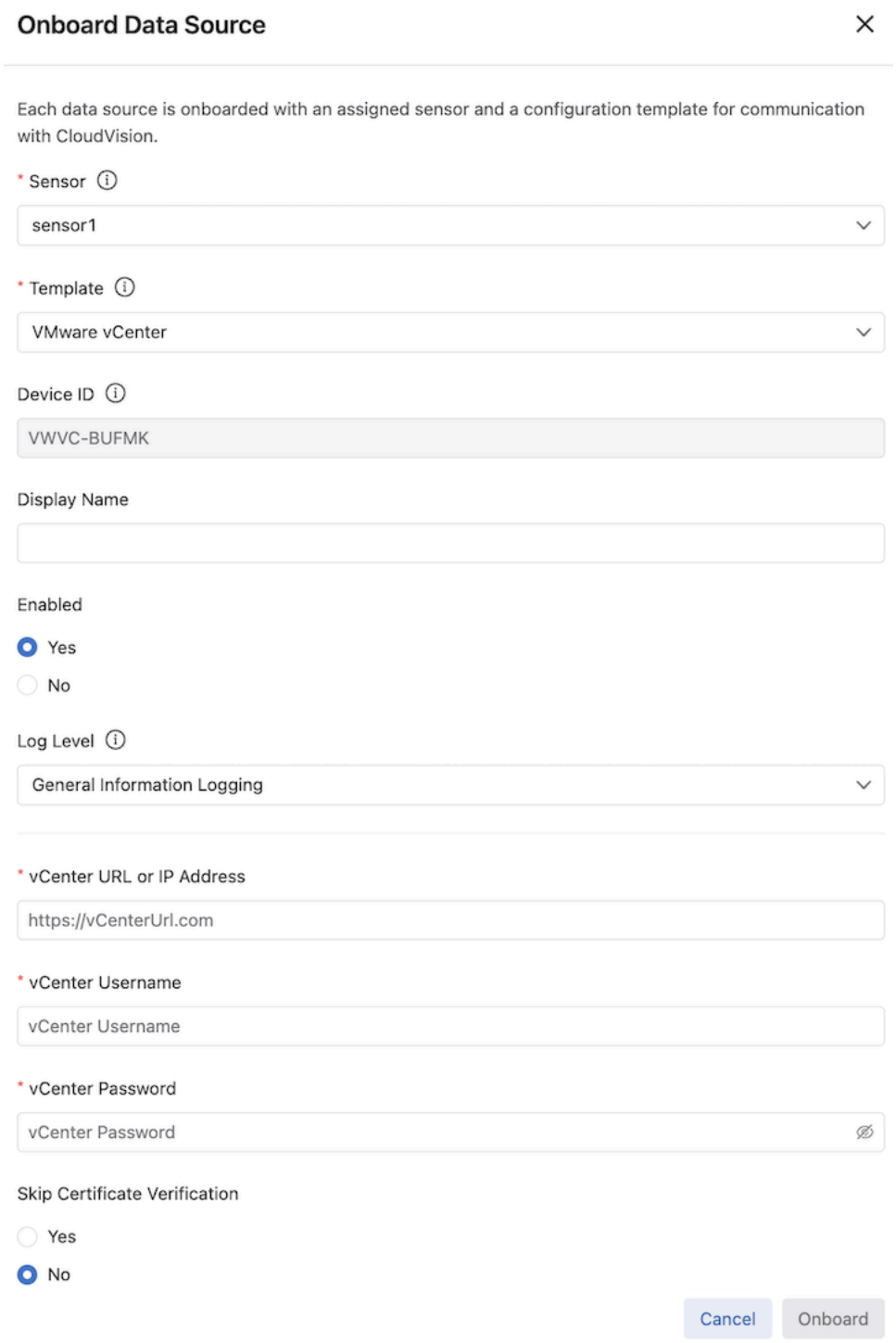
To add VMware vCenter as a Data source:

Select the **VMware vCenter** template to onboard vCenter as a Data Source in CloudVision. Use the *read-only credentials* to onboard your vCenters. CloudVision does not perform any write operations in vCenter.

Note: If you choose the option **Skip Certificate Verification** as ***no*** for vCenter data sources, provide the CA certificates if the vCenter servers are using certificates issued by a private or internal CA. These certificates are required for successful TLS verification between the Sensor and vCenter servers.

Or, if you do not have the CA certificate or wish to continue without CA certificate verification, choose the option **Skip Certificate Verification** as yes.

Figure 2-14: Onboard a Data Source



After adding VMware vCenter as a Data Source in CloudVision, it is recommended to configure the following in the vCenter to enable proper CV UNO functionalities:

- Enable LLDP transmission on Distributed Virtual Switches (DVS)

- Enable NetFlow on Distributed Virtual Switches

2.6.1.1 Enabling LLDP in vCenter

To enable LLDP for ESXi hosts managed by a DVS:

1. Log in to the vCenter.
2. Navigate to **Hosts and Clusters** → **Networking**.
3. Right-click on the **Distributed Virtual Switch** used by the ESXi host in question by navigating to **Settings** → **Edit settings** → **Advanced** → **Discovery Protocol**
4. Choose the Discovery Protocol as Link Layer Discovery Protocol, and Both operations.
5. Click the **OK** button.

Figure 2-15: Enable LLDP in vCenter

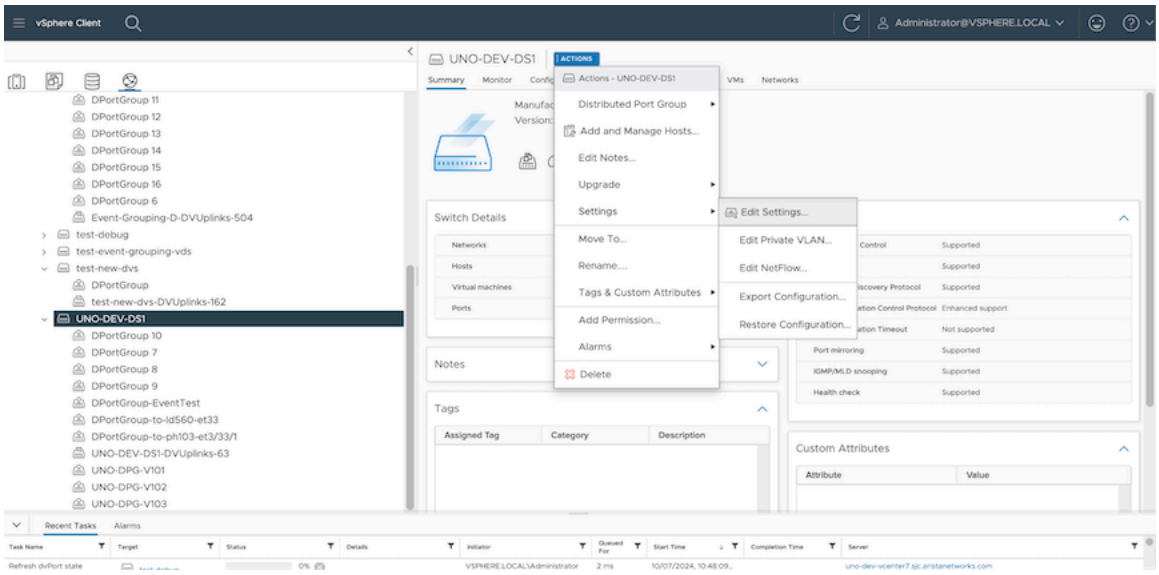
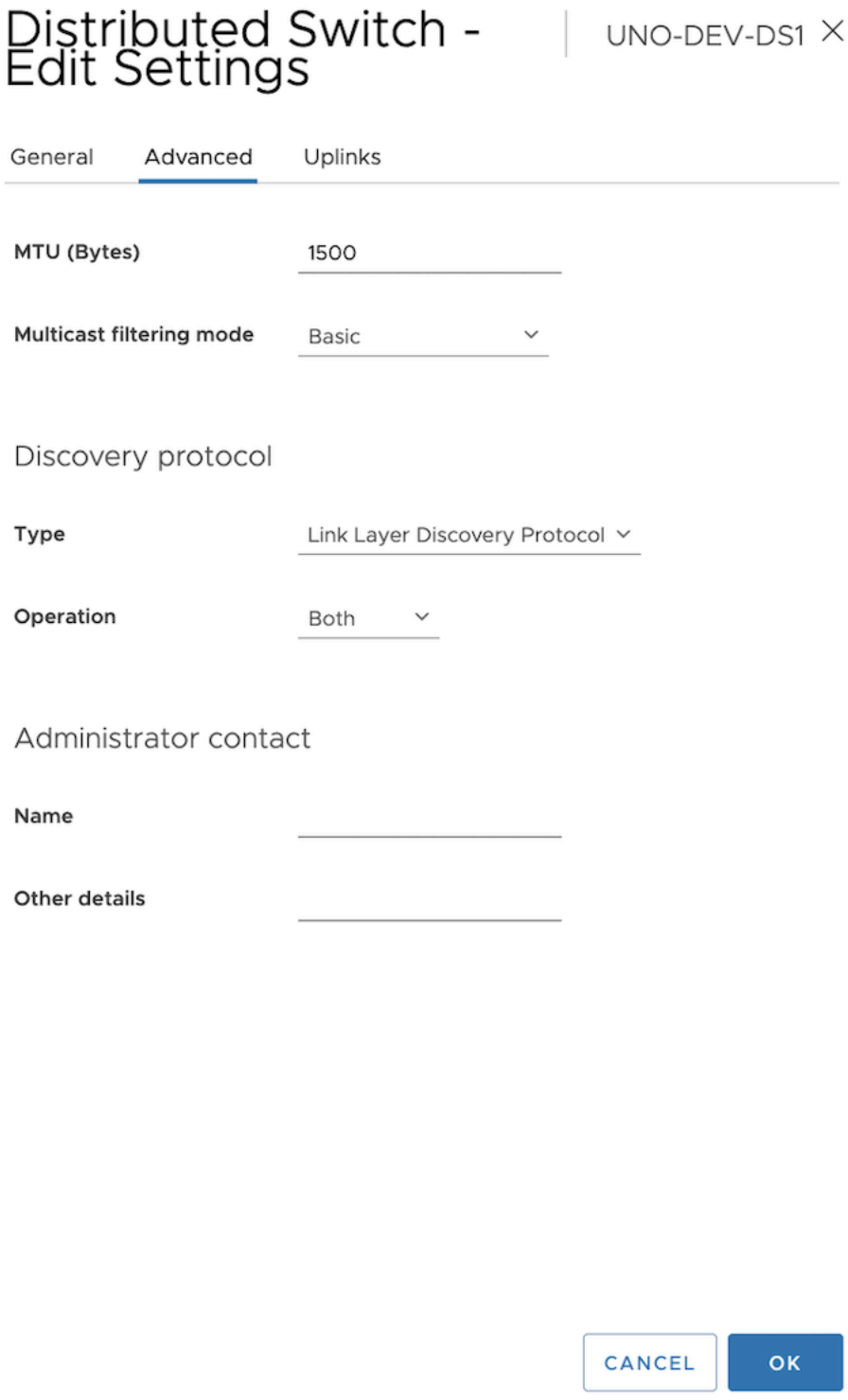


Figure 2-16: Edit Settings on a Distributed Switch



CV Sensor can receive Netflow records from the vCenter. The Sensor consumes the NetFlow records from the vCenter and sends processed flow information to the CVaaS instance.

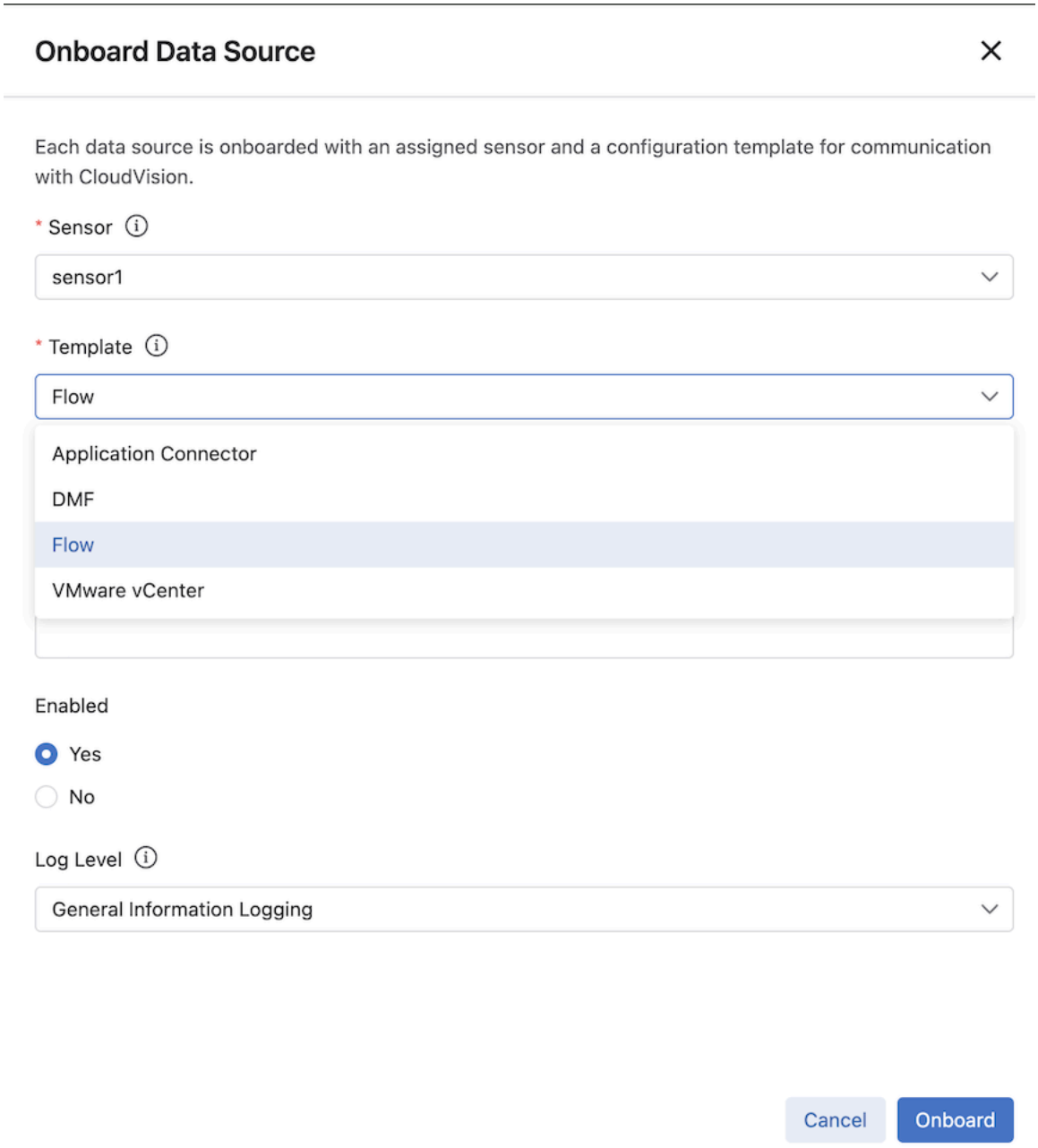
Follow these configuration steps to enable Netflow:

- Sensor Configuration for Enabling Netflow
- vCenter Configuration for Enabling Netflow

2.6.1.2 Sensor Configuration for Enabling Netflow

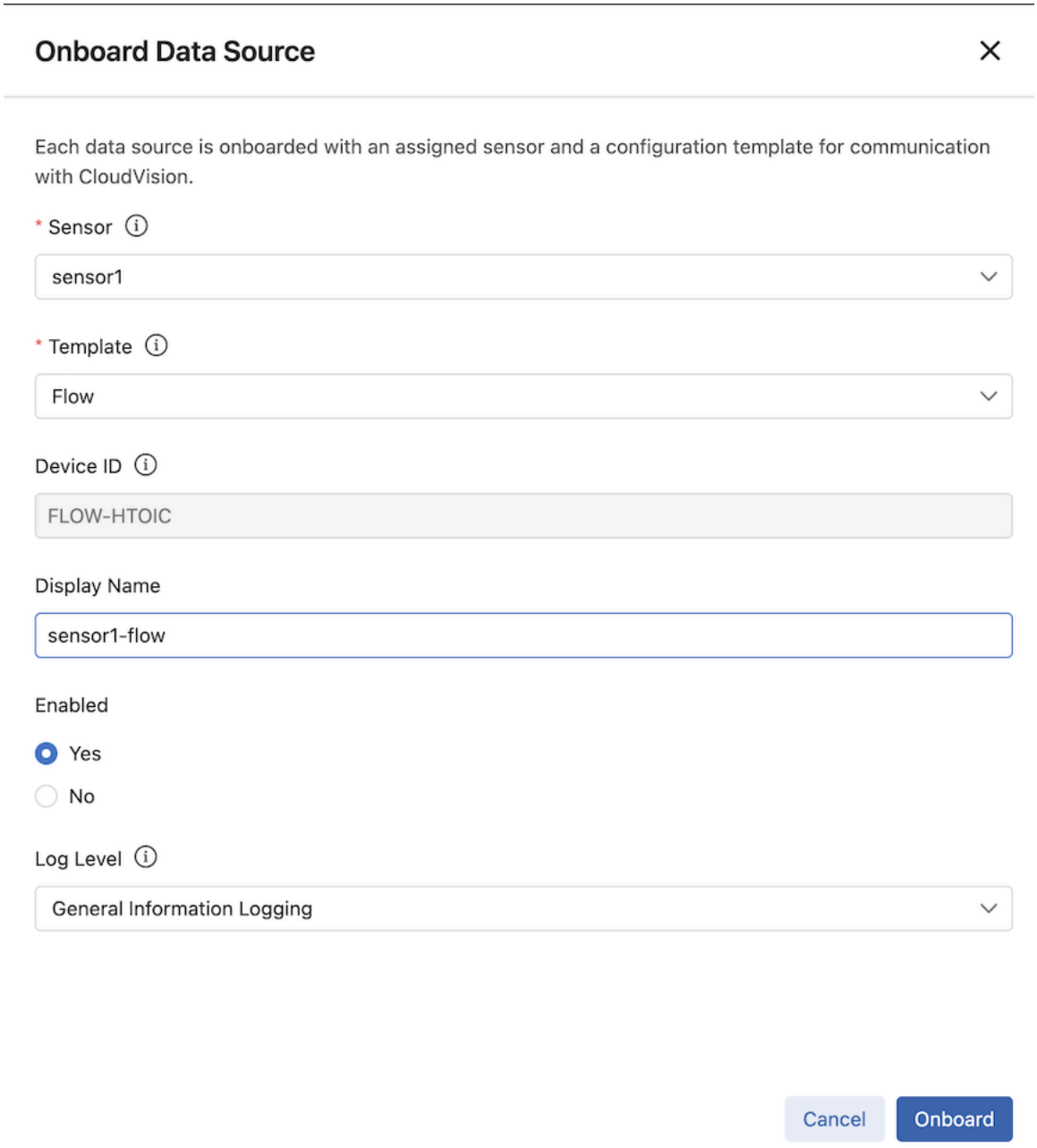
On the **Data Sources** screen, click the **Onboard Data Source**. Select the sensor name and then select **Flow** as the Template (see image).

Figure 2-17: Enabling NetFlow on Sensor



Enter a name for the data source and click the **Onboard** button (see image).

Figure 2-18: Onboard Data Source

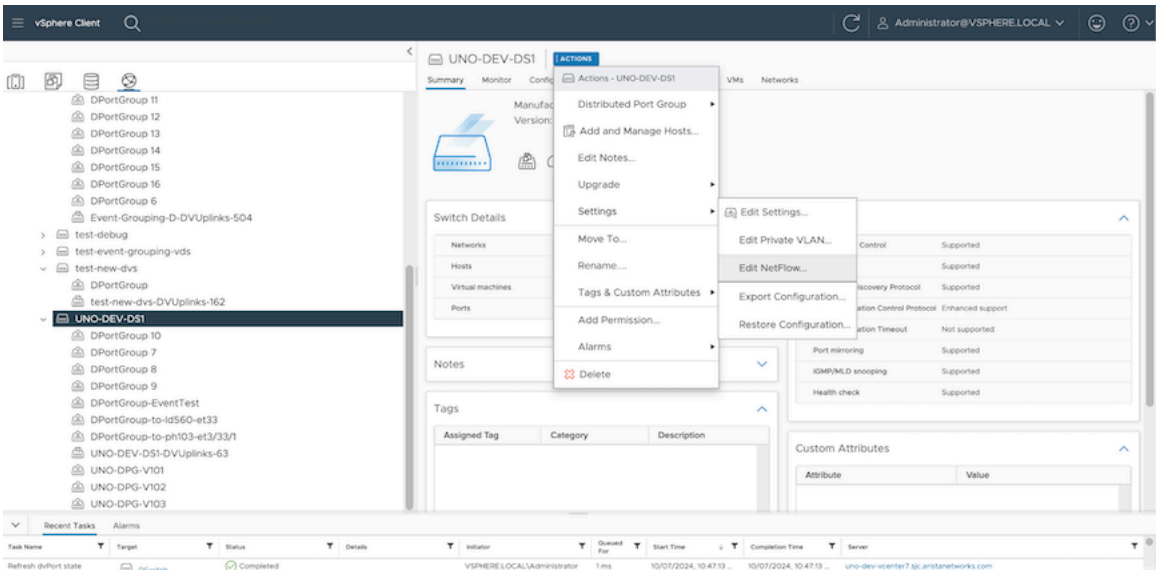


2.6.1.3 vCenter Configuration for Enabling Netflow

To enable Netflow on a vCenter, you must configure each Distributed Virtual Switch (DVS). On each of the Distributed Switch in your vCenter, follow the below steps:

1. Right-click the DVSwitch used by the ESXi host by navigating to **Settings** → **Edit NetFlow**

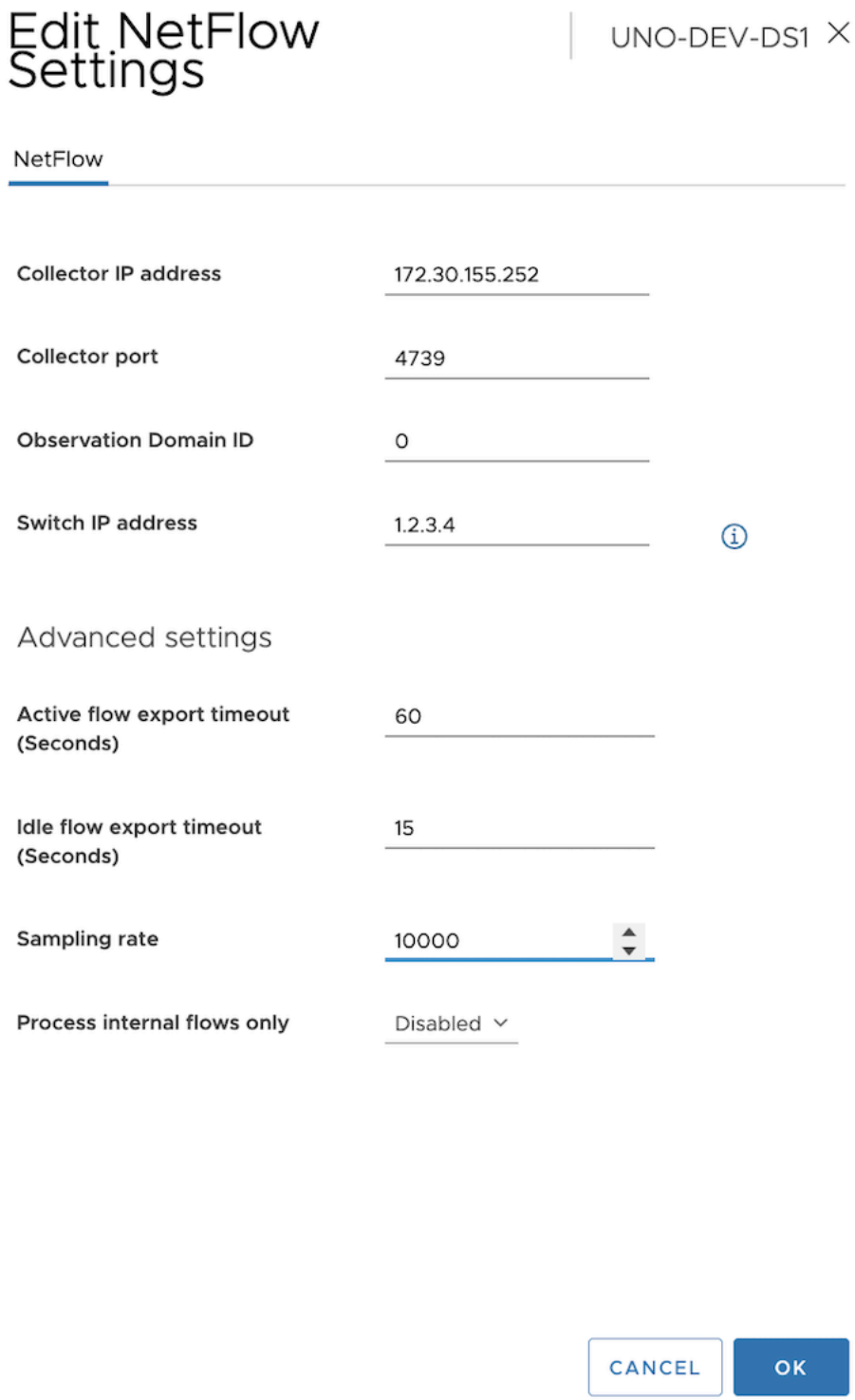
Figure 2-19: Enabling NetFlow on vCenter



2. Add the necessary details in the form as shown in the image below.
   a. Collector IP: Use the Sensor IP
   b. Collector port: **4739**
   c. Sampling Rate: **10000**

      Note: A sampling rate of 10,000 means that one packet will be sampled for every 10,000 packets. To capture more samples and improve visibility on the topology page, reduce the sampling rate to 1000 or less. Remember that reducing the sampling rate may introduce a slight increase in network load.
   d. Switch IP address: Unique IPv4 address across VDSs in a vCenter (not necessarily a pingable IPv4 address)

Figure 2-20: Edit NetFlow Setting for vCenter



3. Click **OK** to save the changes.

After enabling NetFlow on all the DV switches, ensure to enable NetFlow on all Distributed Port Groups of the DV switches by:

1. Right-click on the **DVS** → **Distributed Port Group** → **Manage Distributed Port Groups**
2. Select **Monitoring**
3. Select all of the Distributed port groups (Or select the applicable port groups in your environment)
4. Enable the **Netflow**
5. Click the **Finish** button.

After NetFlow is enabled on a port group, it sends NetFlow data to the collector specified in the DVS settings. However, the port group sends NetFlow data only for ingress packets (entering the port group) and not for egress packets (exiting the port group).

To collect data for all traffic, enable NetFlow for the Uplink port group as well. If you do not enable NetFlow for the uplink port group, the UNO sensor will not receive NetFlow for any traffic going out from the VMs to the physical network.

Note: In the bulk port group configuration, it is not possible to enable NetFlow for the Uplink port group. You must enable the uplink port group separately.

To enable the uplink port group:

1. Right-click on the **Uplink Port group** under the **Distributed Virtual Switch** section → **Settings** (The port group name should have the **DVUPlinks** on it).
2. Navigate to the **Monitoring** tab
3. Enable **Netflow**
4. Click the **OK** button to save the changes.

For details on Adding VMware vCenter as a Data Source, refer to: https://faddom.com/network-visibility-in-virtual-environments-part-2/

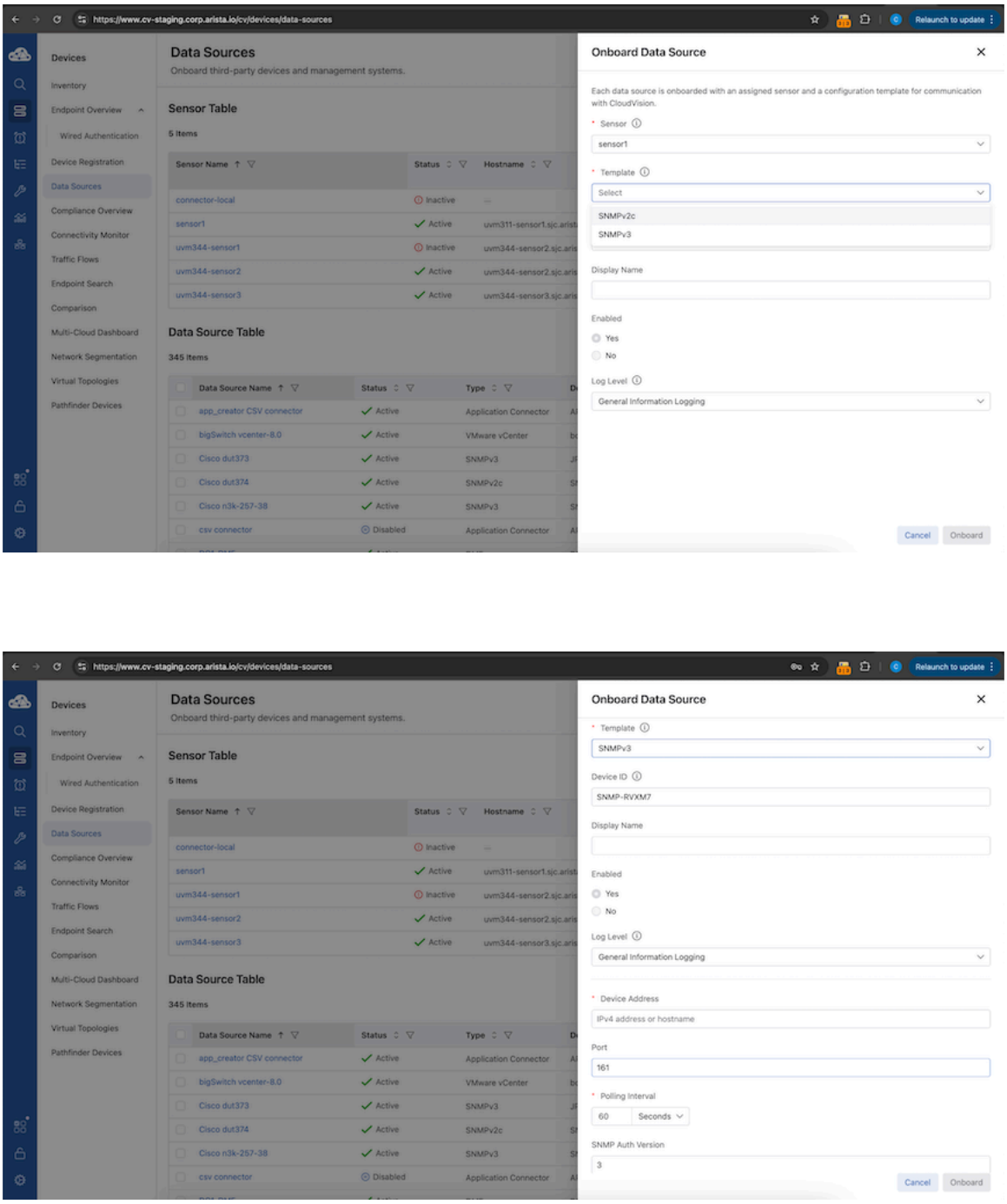
2.6.2 Adding SNMP Data Source

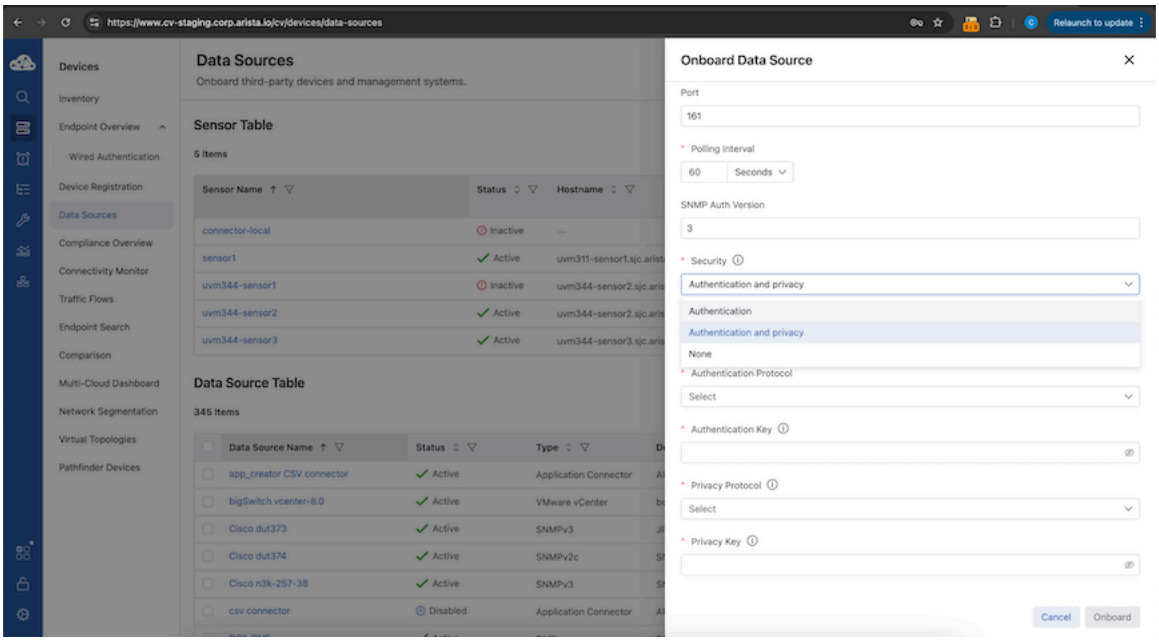
Devices from third-party vendors that support SNMP can be integrated into Cloud Vision as SNMP data sources. Both SNMPv2c and SNMPv3 versions are supported.

To add SNMP data source:

1. Click the **Onboard Data Source** button, and choose either the **SNMPv2c** or **SNMPv3** template from the Template drop down menu.

Figure 2-21: Onboarding SNMP Data Source





2. Enter the required information based on the selected template.
3. Common fields:
   a. Device address
   b. Port
   c. SNMP version
4. For the SNMPv2c template, enter the Community string.
5. For the SNMPv3 template, specify the
   a. Security Level
   b. Security Name
   c. Authentication Protocol
   d. Authentication Key
   e. Privacy Protocol
   f. Privacy Key according to the chosen Security Level.

If the entered details are correct and the Sensor can successfully communicate with the device, then the device gets added to the sensor, and starts the data streaming from SNMP walk/get to CVaaS.

If there are any issues, the data source will be marked as red (inactive). Check the relevant log messages to identify the cause of the issue.

Chapter 3

Upgrading the Sensor

To upgrade a sensor, perform the following steps:

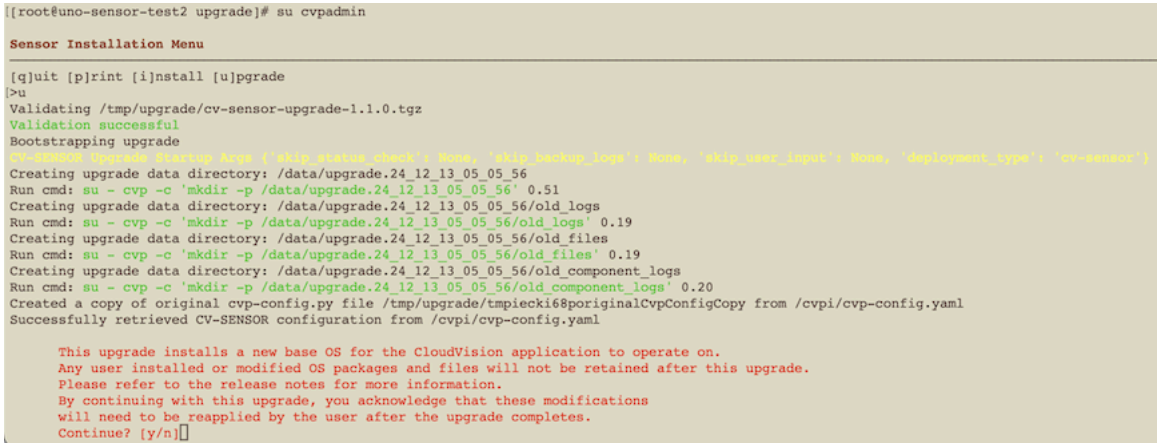
1. To prepare for the upgrade, run following commands on the sensor VM:

```
rm -rf /tmp/upgrade
mkdir /data/upgrade
ln -s /data/upgrade /tmp/upgrade
```

   Note: If the directory already exists, you may see an error message during the mkdir step. Ignore the error message.

2. Get the latest sensor upgrade package (for example, cv-sensor-upgrade-1.1.0.tgz): Download the latest file from the Software Download page, scroll down to the **CloudVision** > **CloudVision Sensor** > **cv-sensor-1.1.0**.
3. Run the upgrade command:
   a. Access the VM by logging in as '**cvpadmin**'.
   b. Type '**u**' or '**upgrade**' (case sensitive) to select the upgrade option:

Figure 3-1: Upgrading the Sensor CLI Output



   Confirm the base OS upgrade warning for the upgrade process to proceed.

4. Confirm reboot to complete the upgrade process.

```
Ready to upgrade.
Current version:        1.1.0
Upgrading to:           1.1.0
Config saved as:        /tmp/upgrade/cvp-config.yaml
Cluster nodes are:      ['uno-sensor-test2.sjc.aristanetworks.com']
Data directory:         /data/upgrade.24_12_13_05_05_56
Upgrade log is at:      /data/upgrade.24_12_13_05_05_56/cvpUpgrade.log

This upgrade will require a reboot.

Do you want to continue? [y/n]
```

After the system reboots, review the upgrade logs in the *cvpUpgrade.log* file available in the */data/* directory. Look for the latest directory with a name formatted as *upgrade.<timestamp>*. For example, /data/upgrade.24_12_13_05_05_56/cvpUpgrade.log.

Once the upgrade is successful, the log file displays a message confirming the upgrade: "*CV-SENSOR successfully upgraded from version x.x.x to y.y.y*"

```
[root@uno-sensor-test2 data]# tail -f upgrade.24_12_13_05_33_39/cvpUpgrade.log
[Fri Dec 13 05:34:40 UTC 2024] Waiting for cvpi service to start.
[Fri Dec 13 05:34:40 UTC 2024] cmd    : |sudo systemctl is-active --quiet cvpi|
[Fri Dec 13 05:34:40 UTC 2024] stdout: ||
[Fri Dec 13 05:34:40 UTC 2024] stderr: ||
[Fri Dec 13 05:34:40 UTC 2024] rc     : |0|
[Fri Dec 13 05:34:40 UTC 2024] Cvpi service has started
[Fri Dec 13 05:34:40 UTC 2024] Starting all components.
[Fri Dec 13 05:34:40 UTC 2024] Checking status.. (it may take up to 0.0h 30.0m)
[Fri Dec 13 05:38:23 UTC 2024] All components are up and running
[Fri Dec 13 05:38:23 UTC 2024] CV-SENSOR successfully upgraded from version 1.1.0 to 1.1.0
```

If the upgrade step fails due to any components not running, conduct an initial investigation to determine the cause of the particular component's unavailability.

After you have identified the issue causing the component to be inactive and still wants to upgrade, proceed to the next step to initiate the sensor upgrade, even though the component is not running.

During a failure, an error message is displayed:

```
CV-SENSOR upgrade failed with the following error:
:..
CvpRuntimeError: Some components are not running, CV-SENSOR cannot be upgraded
```

Perform the following steps to proceed with the Sensor upgrade:

1. Navigate to the '*/tmp/upgrade*' directory by using the command:

   ```
   cd /tmp/upgrade
   ```

2. Check the availability of cvpUpgrade.py file by using the command:

   ```
   ls cvpUpgrade.py
   ```

   If the above command returns no results, untar the upgrade.tgz file using the command:

   ```
   tar -xvf cv-sensor-upgrade-*.tgz
   ```

3. Execute the upgrade process by using the following command:

```
./cvpUpgrade.py --deployment-type cv-sensor --skip-status-check SKIP_STATUS_CHECK
```

The upgrade process is initiated.

Chapter 4

Troubleshooting [New Installation]

This section provides information on common issues that may arise during the CV Sensor deployment and suggests possible solutions to address them.

4.1 How to restart the sensor component?

To restart the sensor:

1. SSH to the VM
2. Execute the following cvpi commands to restart the sensor:

```
cvpi stop sensor --is-local-action
cvpi start sensor --is-local-action
```

Below is an example of the restart process:

```
[root@cvp230 ~]# cvpi stop sensor --is-local-action
- command: kubectl delete -f /cvpi/conf/kubernetes/sensor.yaml --ignore-not-found=true
  cmdfunc: StopActionCmd
  component: sensor
  action: stop
  node: primary
  result:
   ip: ""
   stdout: ""
   stderr: ""
   err: null
   exitcode: 0

root@cvp230 ~]# cvpi start sensor --is-local-action
- command: bash -c "envsubst < /cvpi/conf/kubernetes/sensor.yaml | kubectl apply -f
    -"
  cmdfunc: StartActionCmd
  component: sensor
  action: start
  node: primary
  result:
    ip: ""
    stdout: ""
    stderr: ""
    err: null
    exitcode: 0
```

3. After the restart, verify if all components are running correctly:

```
cvpi status all -v3
```

Below is an example:

```
[root@cvp230 ~]# cvpi status all -v3
Executing command. This may take some time...
```

```
(E) => Enabled
(D) => Disabled
(?) => Zookeeper Down

Action Output
COMPONENT             ACTION              NODE                STATUS
 ERROR
containerd                status              primary             (E) RUNNING
   -
coredns                   status              primary             (E) RUNNING
   -
descheduler               status              primary             (E) RUNNING
   -
etcd                      status              primary             (E) RUNNING
   -
flannel                   status              primary             (E) RUNNING
   -
fluent-bit                status              primary             (E) RUNNING
   -
kube-apiserver            status              primary             (E) RUNNING
   -
kube-controller-manager   status              primary             (E) RUNNING
   -
kube-proxy                status              primary             (E) RUNNING
   -
kube-scheduler            status              primary             (E) RUNNING
   -
kubelet                   status              primary             (E) RUNNING
   -
mutating-webhook          status              primary             (E) RUNNING
   -
mutating-webhook-server   status              primary             (E) RUNNING
   -
sensor                    status              primary             (E) RUNNING
   -
sensor-monitor            status              primary             (E) RUNNING
   -
zookeeper                 status              primary             (E) RUNNING
   -
```

4.2 How to Monitor Sensor Resource Utilization?

The sensor metric is exported to the CVaaS to track sensor resource utilization, which is displayed on the dashboard. Enable the UNO features toggle to view the sensor dashboard. You can also access the Sensor Health Dashboard from the Dashboards:

To access the Sensor dashboards:

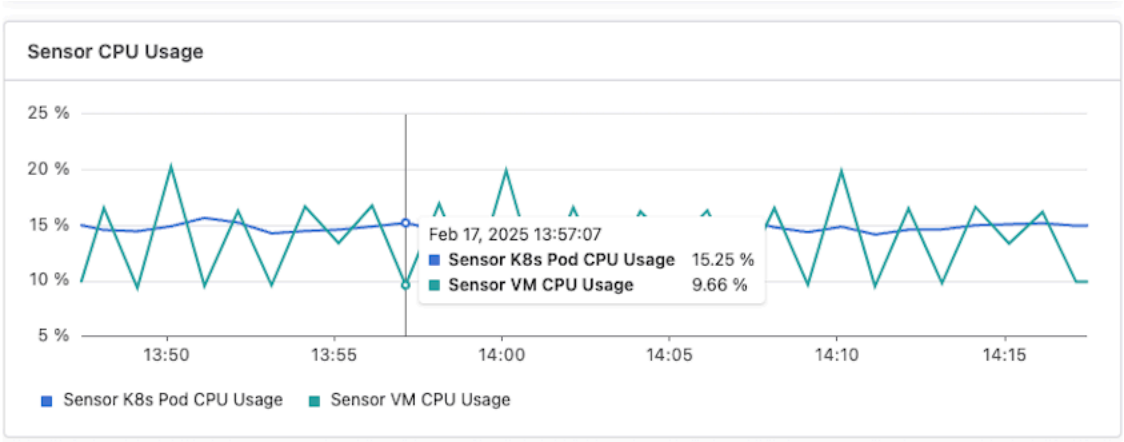
1. Login to the CVaaS.
2. To access the Sensor Dashboard:
   a. Navigate to **Dashboards** > **Sensor Health Monitor**
   b. Select a sensor from the drop-down list.
3. For embedded sensor dashboard on the sensor details screen:
   a. Navigate to **Devices** > **Data sources**
   b. Click an available sensor
   c. Scroll down for graphs

The dashboard includes the following metrics:

- Sensor VM CPU utilization

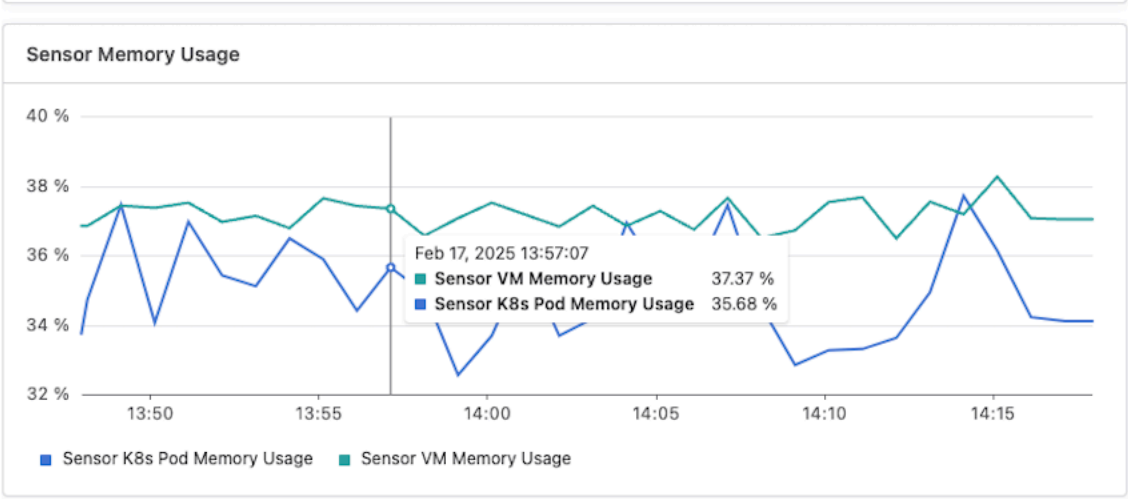
- Sensor K8s Pod CPU utilization

Figure 4-1: Sensor CPU Usage



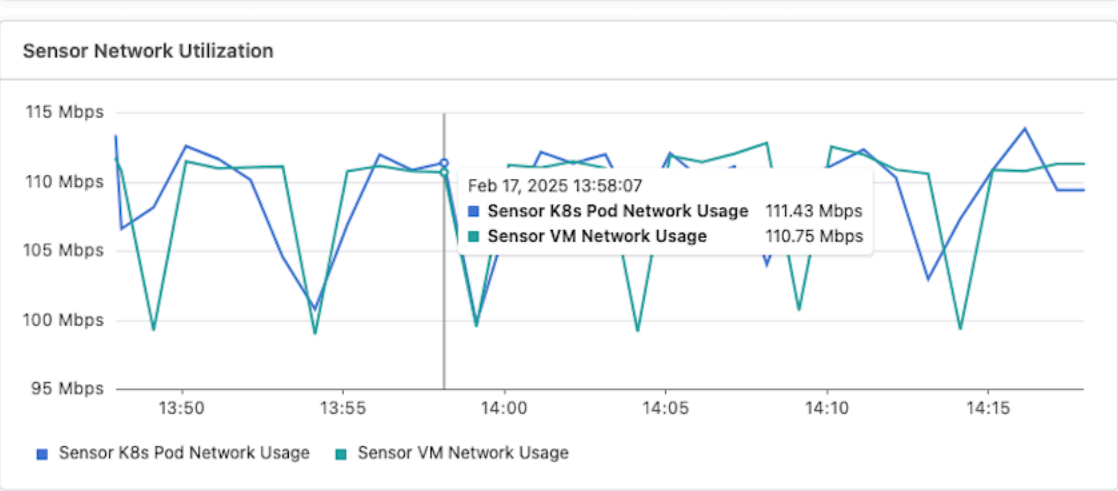
- Sensor VM Memory utilization
- Sensor K8s Pod Memory utilization

Figure 4-2: Sensor Memory Usage



- Sensor VM Network utilization
- Sensor K8s Pod Network utilization

Figure 4-3: Sensor Network Usage



4.3 Where to check for logs?

To check for the log files:

1. SSH to the VM. The logs are managed by `journald` and can be viewed using `journalctl` commands:

```
[root@cvp230 ~]# journalctl IO_KUBERNETES_CONTAINER_NAME=sensor
```

Below is a sample log content for reference:

```
Jul 01 04:59:52 nic3-uno-sensor.sjc.aristanetworks.com sensor[3759]: time="2024-07-01T04:59:52Z" level=debug
 msg="v2Client: SetRequest {origin:\"openconfig\" elem:{name:\"interfaces\"} elem:{name:\"interface\" key:
{key:\"name\" value:\"GigabitEthernet1/0/12\"}} elem:{name:\"state\"} elem:{name:\"counters\"} target:\"FJB2
338B0GR\"}: 5 updates, 0 replaces, 0 deletes" file="v2/v2client.go:80"
Jul 01 04:59:52 nic3-uno-sensor.sjc.aristanetworks.com sensor[3759]: time="2024-07-01T04:59:52Z" level=debug
 msg="Send SetRequest: prefix:{elem:{name:\"interfaces\"} elem:{name:\"interface\" key:{key:\"name\" value:
\"TenGigabitEthernet1/1/7\"}} elem:{name:\"state\"} elem:{name:\"counters\"}} update:{path:{elem:{name:\"in-
multicast-pkts\"}} val:{uint_val:119653}} update:{path:{elem:{name:\"in-octets\"}} val:{uint_val:32160350}}"
 file="dedup/dedup_proxy.go:203"
Jul 01 04:59:52 nic3-uno-sensor.sjc.aristanetworks.com sensor[3759]: time="2024-07-01T04:59:52Z" level=debug
 msg="v2Client: SetRequest {origin:\"openconfig\" elem:{name:\"interfaces\"} elem:{name:\"interface\" key:
{key:\"name\" value:\"TenGigabitEthernet1/1/7\"}} elem:{name:\"state\"} elem:{name:\"counters\"} target:
\"FJB2338B0GR\"}: 2 updates, 0 replaces, 0 deletes" file="v2/v2client.go:80"
Jul 01 04:59:52 nic3-uno-sensor.sjc.aristanetworks.com sensor[3759]: time="2024-07-01T04:59:52Z" level=debug
 msg="Send SetRequest: prefix:{elem:{name:\"interfaces\"} elem:{name:\"interface\" key:{key:\"name\" value:
\"TenGigabitEthernet1/1/4\"}} elem:{name:\"state\"} elem:{name:\"counters\"}} update:{path:{elem:{name:
\"in-multicast-pkts\"}} val:{uint_val:5416661}} update:{path:{elem:{name:\"in-octets\"}} val:{uint_val
:1009059178}} update:{path:{elem:{name:\"out-multicast-pkts\"}} val:{uint_val:1794289}} update:{path:{elem:
{name:\"out-octets\"}} val:{uint_val:226080394}}" file="dedup/dedup_proxy.go:203"
Jul 01 04:59:52 nic3-uno-sensor.sjc.aristanetworks.com sensor[3759]: time="2024-07-01T04:59:52Z" level=debug
 msg="v2Client: SetRequest {origin:\"openconfig\" elem:{name:\"interfaces\"} elem:{name:\"interface\" key:
{key:\"name\" value:\"TenGigabitEthernet1/1/4\"}} elem:{name:\"state\"} elem:{name:\"counters\"} target:
\"FJB2338B0GR\"}: 4 updates, 0 replaces, 0 deletes" file="v2/v2client.go:80"
```

Note: Append `-f` to `journalctl` command to follow logs.

2. Check the logs between a specific time interval using the command:

```
journalctl IO_KUBERNETES_CONTAINER_NAME=sensor --since "2024-07-26 12:10:46" --until
 "2024-07-26 12:11:46
```

Below are examples of `journalctl` commands to filter logs:

- To check all the error logs of system:`journalctl -p err -b`

  You can change level from err to info, warning, alert, debug
- To check only stdout logs: `journalctl _TRANSPORT=stdout`
- To check logs from specific time: `journalctl --since "2024-01-24 17:15:00"`
- To check logs for specific service: `journalctl -u zookeeper.service --since today`
- To check logs for specific process id: `journalctl _PID=3918`
- To check last 100 lines of logs: `journalctl -n 100`
- To follow logs: `journalctl -f`
- Some helpful grep commands for data source specific logs: `journalctl IO_KUBERNETES_CONTAINER_NAME=sensor -n 1000 | grep Flow_Datasource_name ⇒ logs by datasource namejournalctl IO_KUBERNETES_CONTAINER_NAME=sensor -n 1000 | grep provider=events ⇒ logs for events providerjournalctl IO_KUBERNETES_CONTAINER_NAME=sensor -n 1000 | grep datasource=uvm244-vcsim3 ⇒ logs for specific datasourcejournalctl IO_KUBERNETES_CONTAINER_NAME=sensor -n 1000 | grep vcenterId=fda4fd5c-bd4e-4554-925d-f142a3232667 ⇒ logs for vcenter datasource matching given vcenter uuid`

Below are some cvpi commands to check logs:

- To check current sensor pod logs: `cvpi logs sensor`
- To check all sensor logs: `cvpi logs sensor --full`
- To pack sensor logs to tar file: `cvpi debug logs`

4.4 How to change the access_token?

To change the access token:

1. SSH to the VM.
2. Obtain the new service account token.
3. Write down the new service account token into the file `/etc/cvpi/access_token`. You can do this using a text editor or the commands:

```
echo "your_new_access_token_here" | sudo tee /etc/cvpi/access_token
```

4. Run the following commands:

```
cvpi stop sensor --is-local-action
```

```
cvpi init sensor --is-local-action
cvpi start sensor --is-local-action
```

The above commands stop the sensor, initiate the sensor, and then start the sensor with the updated access_token. For example:

```
root@cvp230 ~]# cvpi stop sensor --is-local-action
- command: kubectl delete -f /cvpi/conf/kubernetes/sensor.yaml --ignore-not-found=true
  cmdfunc: StopActionCmd
  component: sensor
  action: stop
  node: primary
  result:
    ip: ""
    stdout: ""
    stderr: ""
    err: null
    exitcode: 0

[root@cvp230 ~]# cvpi init sensor --is-local-action
+ '[' 0 '!=' 1 ']'
+ TOKEN_DIR=/etc/cvpi
+ TOKEN_FILE=/etc/cvpi/access_token
+ '[' '!' -f /etc/cvpi/access_token ']'
+ '[' '!' -s /etc/cvpi/access_token ']'
+ echo 'Creating kubernetes secret for cvp service access token'
+ kubectl delete secret secret-service-access-token --ignore-not-found=true
+ kubectl create secret generic secret-service-access-token --from-file=/etc/cvpi/access_token
+ rm /etc/cvpi/access_token
+ touch /etc/cvpi/access_token
- command: /cvpi/apps/sensor/bin/init.sh
  cmdfunc: DefaultActionCmd
  component: sensor
  action: init
  node: primary
  result:
    ip: ""
    stdout: |
      Creating kubernetes secret for cvp service access token
      secret "secret-service-access-token" deleted
      secret/secret-service-access-token created
    stderr: |
      + '[' 0 '!=' 1 ']'
      + TOKEN_DIR=/etc/cvpi
      + TOKEN_FILE=/etc/cvpi/access_token
      + '[' '!' -f /etc/cvpi/access_token ']'
      + '[' '!' -s /etc/cvpi/access_token ']'
      + echo 'Creating kubernetes secret for cvp service access token'
      + kubectl delete secret secret-service-access-token --ignore-not-found=true
      + kubectl create secret generic secret-service-access-token --from-file=/etc/cvpi/access_token
      + rm /etc/cvpi/access_token
      + touch /etc/cvpi/access_token
    err: null
    exitcode: 0

root@cvp230 ~]# cvpi start sensor --is-local-action
- command: bash -c "envsubst < /cvpi/conf/kubernetes/sensor.yaml | kubectl apply -f
    -"
  cmdfunc: StartActionCmd
  component: sensor
  action: start
  node: primary
  result:
    ip: ""
    stdout: ""
    stderr: ""
    err: null
    exitcode: 0
```

4.5 How to check if access_token is expired or not?

To check whether the access_token is expired or not, examine the sensor log file `/cvpi/apps/sensor/logs/sensor-sensor1.log` for the below error messages:

- Wrong service account token or expired token:

```
time="2022-11-18T14:14:41+05:30" level=error msg="Unable to reach gNMI service: rpc error: code
 = Unauthenticated desc = unexpected HTTP status code received from server: 401 (Unauthorized). retrying
 (attempt 5)..."
```

- No device enrollment read & write permissions:

```
time="2022-11-18T14:54:07+05:30" level=error msg="Datasource stopped unexpectedly: error starting providers
 for device \"45733aec-c463-412c-a301-ea6f294d9692\" (vCenter_v2): gRPC connection to device 45733aec-c463
-412c-a301-ea6f294d9692 failed: AddEnrollmentToken failed: rpc error: code = PermissionDenied desc = not
 authorized" datasource=vcenter1 sensor=sensor1
```

- No gNMI read & write permissions:

```
time="2022-11-18T15:01:48+05:30" level=info msg="Config subscription returned: error: rpc error: code
 = PermissionDenied desc = unexpected HTTP status code received from server: 403 (Forbidden)" sensor=sensor1
time="2022-11-18T15:01:48+05:30" level=info msg="Terminating 0 datasources..." sensor=sensor1
time="2022-11-18T15:01:48+05:30" level=info msg="All datasources closed" sensor=sensor1
time="2022-11-18T15:01:48+05:30" level=fatal msg="group returned with error: rpc error: code = PermissionDen
ied desc = unexpected HTTP status code received from server: 403 (Forbidden)"
```

Problem Resolution:

The above errors in the log files indicate either the service account token has expired or some issue with the token. To resolve this, re-generate the service account token by following the steps in Generating a Service Account Token section.

4.6 How to change the log level and verbosity of the sensor on a VM?

To change the log level and verbosity of the sensor on a VM, perform the following steps:

1. SSH to the VM.
2. Edit the file `/cvpi/conf/helm/standalone-sensor/template/values.yaml`.
3. Change the following parameters with the appropriate values: `logLevel: debugverbosityLevel: 1`

   where,

   - `logLevel`: Log level verbosity (available levels: trace, debug, info, warning, error, fatal, panic. The default value is "debug").
   - `verbosityLevel`: Log level for V logs.
4. Restart the sensor component with the updated configuration:

   `cvpi stop sensor --is-local-actioncvpi config sensorcvpi start sensor --is-local-action`

   These commands stop the sensor, apply the configuration changes, and then start the sensor with the new log level and verbosity settings.

4.7 Are all components on the server running fine?

Check for any error messages in the log file (as in example):

```
E0613 16:28:15.590280   39508 publisher.go:723] ae4aa295-ed2e-48b3-95fb-b5d1e031f1c6Unable to send 206
 messages to dataset: type:"device" name:"ae4aa295-ed2e-48b3-95fb-b5d1e031f1c6" for req #1: rpc error: code
 = Unimplemented desc = unimplemented
```

This error indicates that one or more components on the server are down, causing an issue. To resolve the problem, check and restart the components that are not running correctly.

Chapter 5

Troubleshooting Upgrade Issues

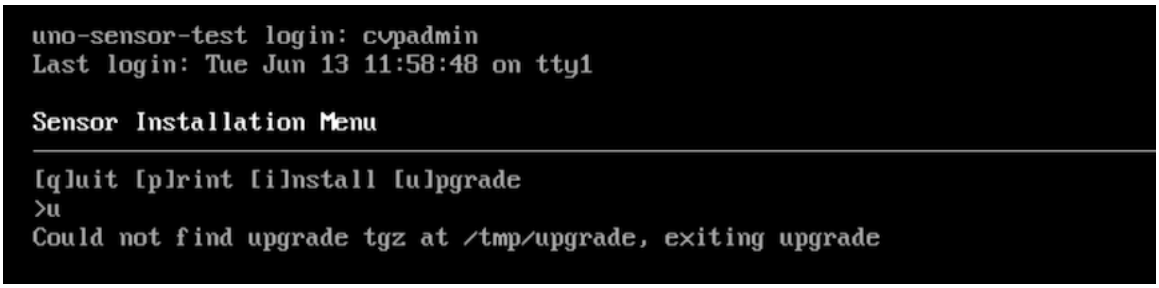
This section provides information on common issues that may arise during Sensor upgrade and suggests possible solutions to address them.

5.1 Where to check upgrade logs?

To check upgrade logs:

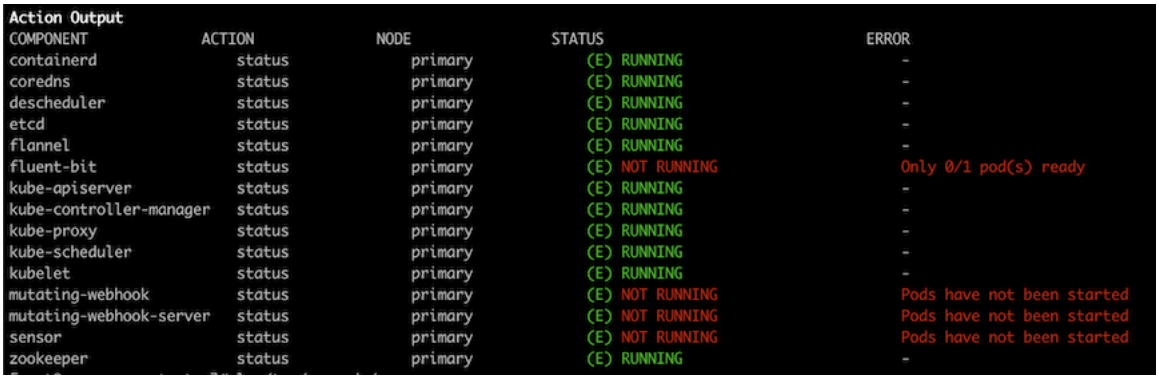
- Locate the log file named `cvpUpgrade.log` in the `/data/upgrade.<timestamp>` location.
- When an upgrade file is missing or not copied properly, an error indicating the issue is displayed.

Figure 5-1: Upgrade Error



- When an upgrade is not required, the following is displayed: `The upgrade will not modify any existing RPMs. All RPMs are up to date. Exiting.`
- After the upgrade, if some components are in a **Not Running** state:

Figure 5-2: Post Upgrade Errors



Resolution:

Try the following possible solutions:

- Restart the container service by using the command: `systemctl restart containerd.service`.
- Check if all components are running by using the command: `cvpi status all -v3`.

If the issue persists, proceed to the next step:

- Re-install the sensor.
- Go to the Sensor Installation Menu and select the "`[i]nstall`" option.
- Follow the steps and apply the configuration without making any modifications.
- Check if all components are running now. If the issue still persists, further debugging is required.

For more advanced troubleshooting issues or to address issues beyond these steps, contact the Arista TAC team for assistance.

Chapter 6

General Troubleshooting Issues

This section provides information on common issues that you may encounter with CV Sensor and suggests possible solutions.

6.1 Active sensor and data sources with no updates for applications in CVaaS.

If data sources are active but there are no updates on the application view page or the VM/host counters on the CVaaS side, it indicates a problem with data transmission.

To troubleshoot the sensor pod status, SSH into the VM and execute the following command:

```
kubectl get pods
```

In the output, check the "**Age**" column of the sensor pod.

If the sensor pod's age is over 90 days and the data source was added around that time, expired certificates might be the cause.

To resolve this issue, restart the sensor pod by executing the following commands:

```
cvpi stop sensor
cvpi start sensor
```

6.2 How to check the sensor version on the sensor VM?

To obtain the sensor version, choose one of the following two options:

Option 1:

Run the following command:

```
rpm -qa | grep sensor
```

This command displays the sensor RPM package installed. Below is a sample command output:

```
[root@uno-cvplay-sensor1 ~]# rpm -qa | grep sensor
sensor-v0.1.2_35826_g3876957_2023.3.0-base_1694543478.x86_64
```

In this output, **v0.1.2_35826_g3876957** is the sensor version. You can verify this against the sensor version on the UI.

Option 2:

Run the following command to display the logs with messages such as: "`Running sensor <sensor_name>, version <sensor_version>`".

Below is a sample command uutput:

```
[root@uno-cvplay-sensor1 ~]# journalctl | grep 'Running sensor .* version' | tail -n 1
Sep 11 13:11:39 uno-cvplay-sensor1.sjc.aristanetworks.com sensor[6429]: time="2023-09-11T13:11:39Z
" level=info msg="Running sensor \"non-default\", version v0.1.4-1997-g71e49ce" file="device/
sensor.go:972" sensor=non-default
```

6.3 What does Error- dial TCP i/o timeout mean?

Certificate problems are the likely cause of sensor status errors like the one shown below:

```
sensor status primary (E) NOT RUNNING Get "https://10.42.128.1:443/apis/apps/v1/daemonsets?lab
elSelector=app%3Dsensor": dial tcp 10.42.128.1:443: i/o timeout
```

To troubleshoot:

1. Backup the `/cvpi/tls/certs` directory.
2. Empty the `/cvpi/tls/certs` directory.
3. Run `/cvpi/bin/certs-gen.sh` with user `cvp`
4. Now, access the sensor installation menu with the command `su cvpadmin`.
5. Choose the `[i]nstall` option followed by `[a]pply config`.
6. During this process, if you see the following error:

   ```
   Unable to connect to the server: dial tcp 10.42.128.1:443: i/o timeout
   ```
7. Terminate the installation and stop `kube-apiserver` with the command:
   `systemctl stop kube-apiserver`
8. Restart the installation using the sensor installation menu.

6.4 How to collect debug logs?

Run the following commands to gather all debug logs:

```
[root@uno-cvplay-sensor1 ~]# cvpi debug all --no-prom
```

Or

```
[root@uno-cvplay-sensor1 ~]# cvpi debug all -n
```

Execute the following command to gather debug logs specific to a sensor component:

```
[root@uno-cvplay-sensor1 ~]# cvpi debug sensor --no-prom
```

Or

```
[root@uno-cvplay-sensor1 ~]# cvpi debug sensor -n
```

6.5 How to change the cluster name(CV_ADDR)?

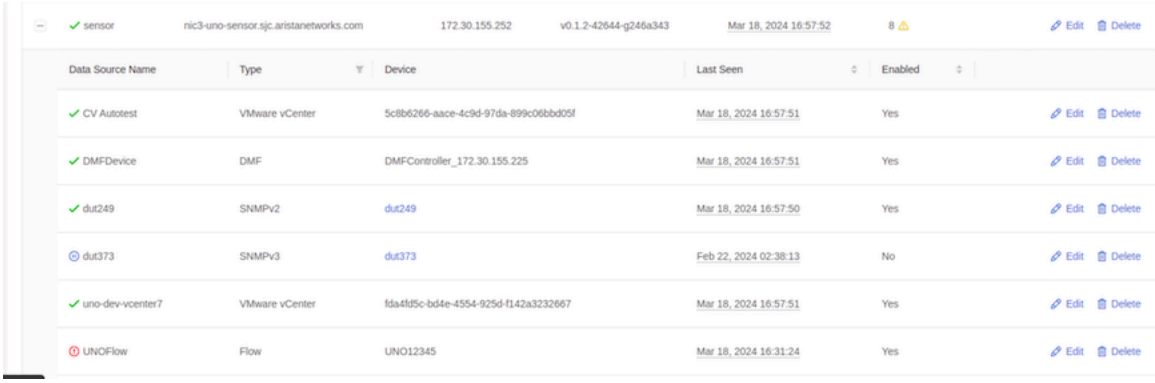
To change the cluster name:

1. SSH to the VM.
2. Log in using `su cvpadmin`.
3. Select the install option from the menu.
4. Select the edit option.
5. Update the `CV_ADDR` field.
6. Apply the configuration.

If you change the `CV_ADDR` to point to a different cluster, update the service account token as well. For details, see the How to change the access token section.

6.6 What action to take when the Flow data source becomes inactive?

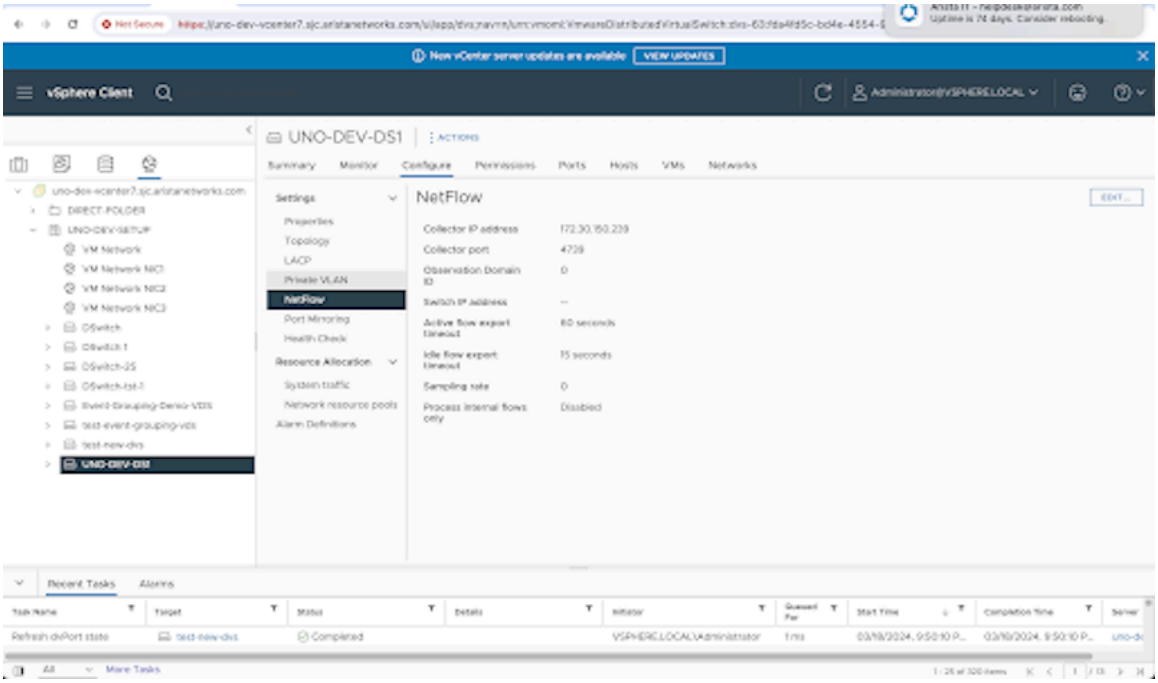
The flow data source on-boarded on the Sensor collects the flow records. These records are sent to the CVaaS via the Sensor VM after the Netflow details are configured in the VDS settings with the correct collector IP and port. If the VDS settings are not configured with the correct collector IP, the Flow data source will become inactive.



To check if vCenter VDS is configured with the correct collector IP:

Login to **vCenter** and navigate to **VDS settings** > **NetFlow settings**.

Figure 6-1: NetFlow Settings



The Collector IP address should match the Sensor VM IP address, and the Collector Port should be 4739. Sometimes, despite the correct collector IP address and port, the flow records don't reach the Sensor VM; that could happen if port 4739 is blocked on the Sensor VM.

6.7 The `Cvpi status all` command do not show component status

To check the status of all components, run the `cvpi status all` command. When there is any issue with the system, the `cvpi` commands fail and displays an error message.

For example,

```
[root@uno-sensor ~]# cvpi status all
Please run sudo systemctl show -p SubState chronyd to ensure the Substate is running and run cvpi
 check all to ensure all scopes are OK.
If these are both OK, please run systemctl enable zookeeper && systemctl restart zookeeper to
 restart zookeeper.

If zookeeper continues to fail, please contact Arista support.
```

When you see this error message, run the following commands and check the status:

```
[root@uno-sensor~]# sudo systemctl show -p SubState chronyd
SubState=start-pre
```

If the status shows `start-pre`, it indicates that the `chronyd` service is not running, preventing the `zookeeper` service from starting.

To resolve this issue, first verify if the `chronyd` service is running. If it is not running, the configured NTP servers may be down, or their configuration might have changed. If NTP server authorization is required, the authorization settings might have been updated.

```
[root@uno-sensor ~]# systemctl status chronyd
```

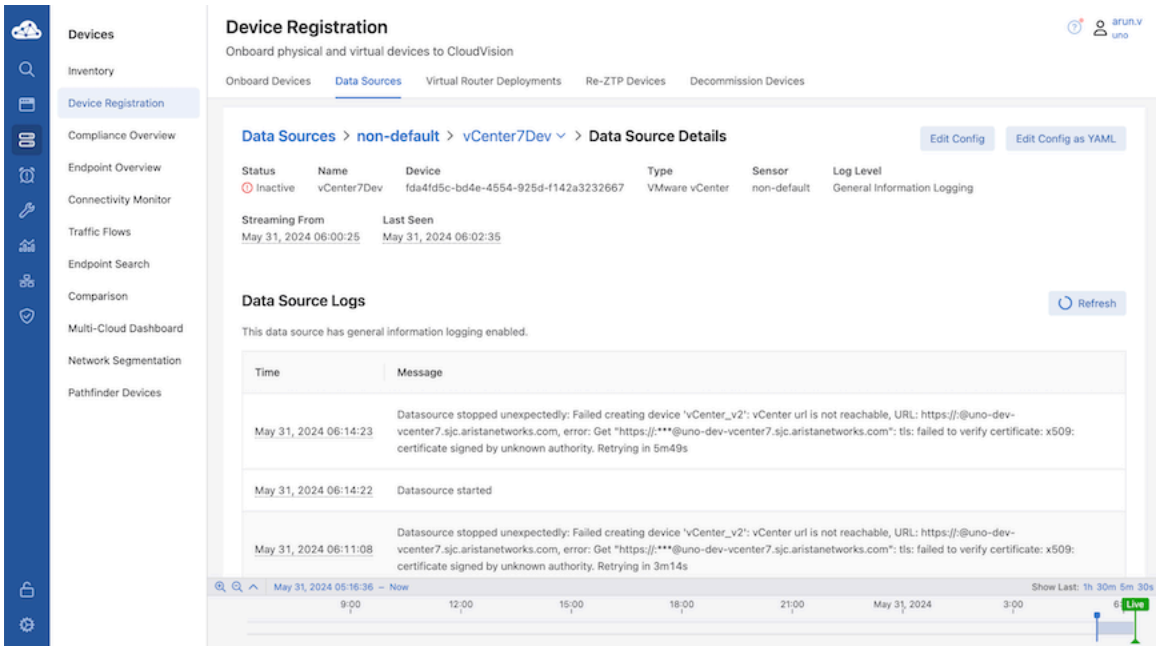
Check if the NTP server is UP:

```
[root@dc1-uno-sensor ~]# ping ntp-server.com
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
From 1.2.3.4 icmp_seq=1 Destination Host Unreachable
From 1.2.3.4 icmp_seq=2 Destination Host Unreachable
```

In this scenario, the NTP server is unreachable. Update the NTP server settings through the sensor configuration menu. Applying the new configuration initiates the installation process with the updated NTP server and starts all necessary services.

6.8 Failed to verify certificate for a vCenter data source.

If the sensor application is unable to verify the server certificates of a vCenter data source, an error message resembling “`tls: failed to verify certificate`” is displayed on the data source details page (see image).



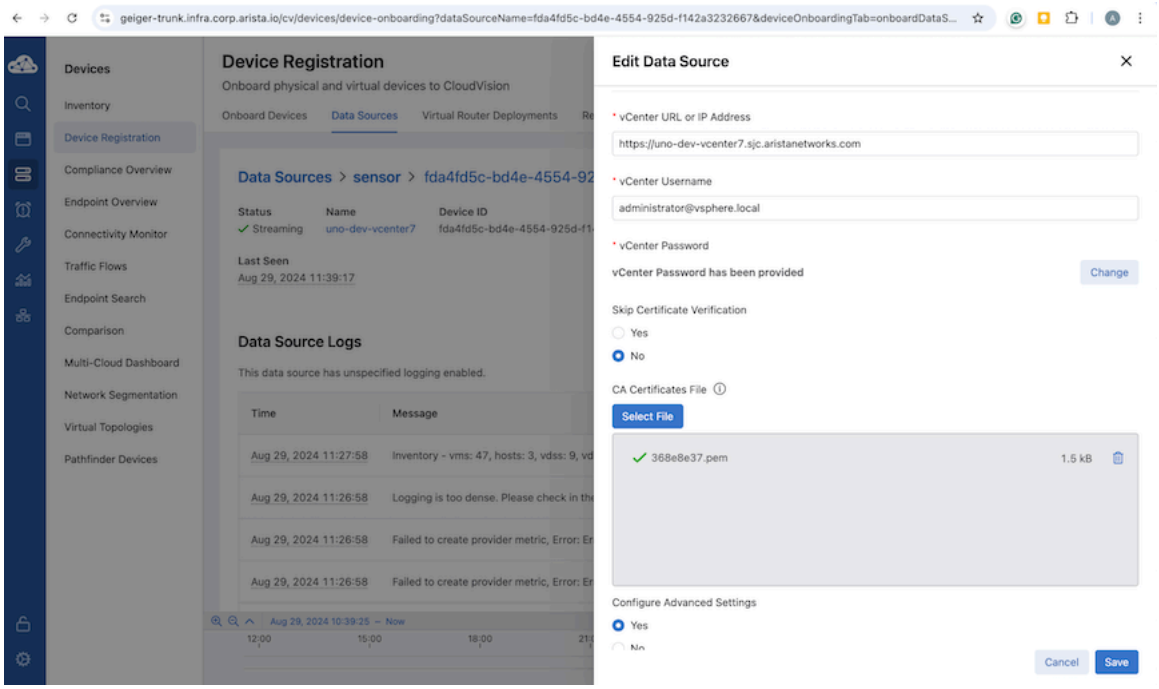
This issue occurs when the vCenter server uses a certificate issued by a private/internal CA, the data source configuration does not contain the necessary CA certificates, or the **Skip certificate** verification option is false.

To mitigate this issue, for a vCenter data source:

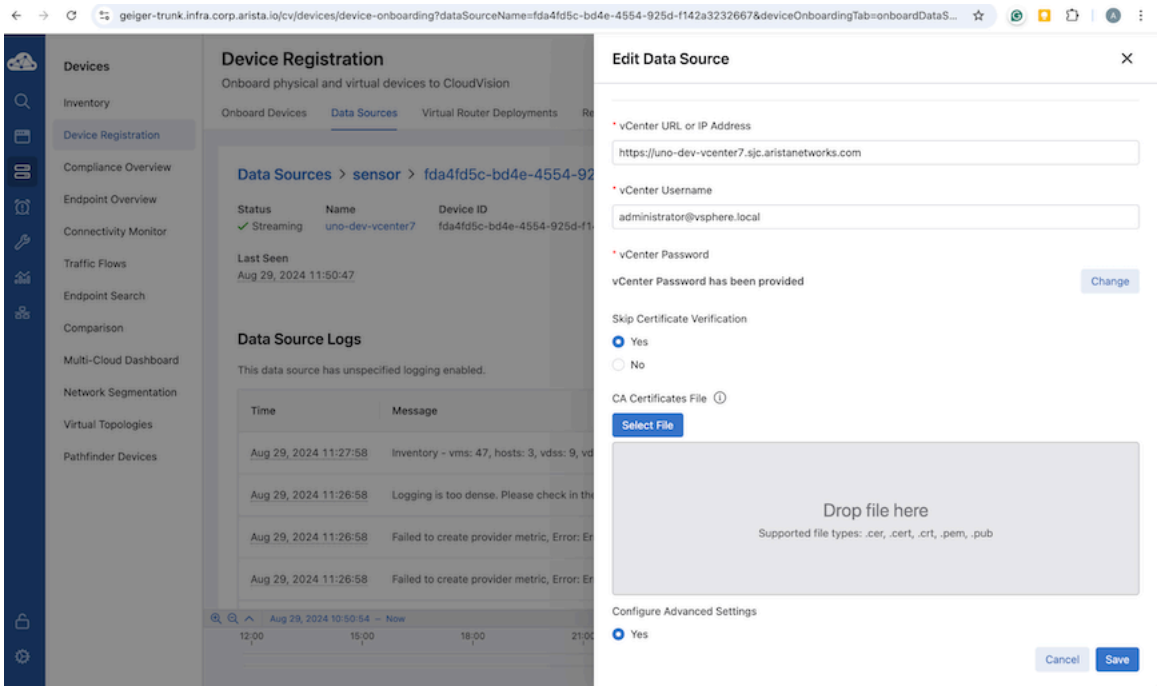
1. Edit Config in the Data Source Details page.
2. Do one of the following:
   a. Upload the correct CA certificate file.

      (i). The CA certificates file is a single file containing the necessary certificates in PEM format. It may contain multiple certificates, i.e., all the intermediate CA certificates and the root CA certificate.

      (ii). Obtain the CA certificate file from the vCenter admin or browse the vCenter server base URL.



b. Or, select **Yes** for the **Skip Certificate Verification** option.



3. Save and choose to onboard the device.