



Configure Collection

This section contains the following topics:

- [Collection Service Overview, on page 1](#)
- [Collection Considerations, on page 1](#)
- [Prerequisites for Device Model Driven Telemetry, on page 2](#)
- [About Collection Jobs, on page 4](#)
- [Collection Jobs, on page 5](#)
- [Monitoring Collection Jobs, on page 13](#)

Collection Service Overview

Networks maintain a large amount of data that spans thousands of devices. The Cisco Crosswork Optimization Engine collects and manages that flow of data via Cisco Crosswork Data Gateway in a multi-vendor environment to provide a real-time publish/subscribe model infrastructure to Cisco Crosswork Network Automation applications. The Collection Service is highly scalable in order to meet the performance demands of the Cisco Crosswork Optimization Engine applications.

Multiple applications requesting same data overload network devices causing outages. Cisco Crosswork Data Gateway's **Collection Optimization** feature tackles this problem by optimizing collection requests. Thereby, reducing redundant data collections.

Users with administrative privileges can monitor Collection Service status and performance, and start/stop/restart it or its underlying services, using the Cisco Crosswork Optimization Engine user interface. You can also collect logs and performance metrics for this service. For help with these tasks, see [Manage Cisco Crosswork Network Automation](#).

Collection Considerations

MDT Collection

When Cisco NSO is used in conjunction with Cisco Crosswork Optimization Engine, the telemetry configurations are pushed to the devices by Cisco NSO.

If you do not plan to use to use Cisco NSO, you must apply the telemetry configuration on your devices. See [Prerequisites for Device Model Driven Telemetry, on page 2](#).



Note The default MDT collector port is 9010.

Device Limits

Cisco Data Gateway collection supports 1000 devices. If your network requires collection of more than 1000 devices, multiple Cisco Data Gateways must be deployed.

Prerequisites for Device Model Driven Telemetry

The Cisco Crosswork Optimization Engine Collection Service configures telemetry as needed on the devices enrolled within the service. Telemetry configuration must be done on PCCs or provider edge routers.



Note Cisco Crosswork Optimization Engine uses the Cisco-IOS-XR-infra-tc-oper YANG module for MDT collection.



Note If an operator configures telemetry directly on the same devices either manually or through some mechanism outside of the Collection Service, the commands must not contain the keyword `cw`. The keyword `cw` is reserved for use by the Collection Service. In particular, the following commands must not contain the keyword `cw` when configured outside of the Collection Service:

```
destination-group
sensor-group
subscription
    sensor-group-id
    destination-id
```

For example (invalid telemetry configuration):

```
telemetry model-driven
  destination-group CW_1b4ac245d863cf3e787d42bae97f1d18dd300d5e
```

For more information, see the telemetry configuration documentation for your particular device (for example: [Telemetry Configuration Guide for Cisco ASR 9000](#))

The default MDT collector port is 9010. Cisco Data Gateway collection supports 1000 devices. If your network requires collection of more than 1000 devices, multiple Cisco Data Gateways must be deployed. See [Collection Considerations, on page 1](#).

Valid Telemetry Configuration

The following sample output shows a *valid* telemetry configuration on a device when configured outside of the Collection Service. If using a single interface network, then the IP address is the management IP address. In a dual interface network, then the IP address should be the data IP address.

```
telemetry model-driven
destination-group OE_43dc8a5ea99529715899b4f5218408a785e40fce
vrf default
```

```

address-family ipv4 192.168.0.3 port 9010
encoding self-describing-gpb
protocol tcp
!
!
destination-group OE_4b3c69a200668b0a8dc155caff295645c684a8f8
vrf default
  address-family ipv4 192.168.0.3 port 9010
  encoding self-describing-gpb
  protocol tcp
!
!
sensor-group OE_43dc8a5ea99529715899b4f5218408a785e40fce
  sensor-path Cisco-IOS-XR-infra-tc-oper:traffic-collector/afs/af/counters/tunnels/tunnel
!
sensor-group OE_4b3c69a200668b0a8dc155caff295645c684a8f8
  sensor-path
Cisco-IOS-XR-infra-tc-oper:traffic-collector/vrf-table/default-vrf/afs/af/counters/prefixes/prefix
!
subscription OE_43dc8a5ea99529715899b4f5218408a785e40fce
  sensor-group-id OE_43dc8a5ea99529715899b4f5218408a785e40fce sample-interval 60000
  destination-id OE_43dc8a5ea99529715899b4f5218408a785e40fce
!
subscription OE_4b3c69a200668b0a8dc155caff295645c684a8f8
  sensor-group-id OE_4b3c69a200668b0a8dc155caff295645c684a8f8 sample-interval 60000
  destination-id OE_4b3c69a200668b0a8dc155caff295645c684a8f8
!

```



Note The **sample-interval** can be changed depending on the size of your network. It is defined in milliseconds and determines how fast you want the data to be pushed out.

Confirm that all PCCs or provider edge routers have telemetry configured and report data to Crosswork Optimization Engine. For example, routers should report prefix and tunnel counters:

```

RP/0/RP0/CPU0:PE1#show traffic-collector ipv4 counters prefix
Thu Jul 11 08:32:32.993 UTC
Prefix Label Base rate TM rate State
(Bytes/sec) (Bytes/sec)
-----
192.168.0.1/32 16001 1 0 Active
192.168.0.2/32 16002 1 0 Active
192.168.0.3/32 16003 1 0 Active
192.168.0.4/32 16004 2 0 Active
192.168.0.6/32 16006 501023 501021 Active
192.168.0.7/32 16007 17320774 17320772 Active
192.168.0.8/32 16008 3737825 3737823 Active
192.168.0.9/32 16097 3 0 Active
192.168.0.10/32 16096 2 0 Active

```

```

RP/0/RP0/CPU0:PE1#show traffic-collector ipv4 counters tunnel
Thu Jul 11 08:32:20.746 UTC
Interface Base rate Base rate State
(Packet/sec) (Bytes/sec)
-----
srte_c_102_ep_192.168.0.7 0 0 Active

```

Cisco IOS XR devices that are onboarded through telemetry must have the following configuration settings on the device to ensure that NETCONF and SSH work correctly:

```
ssh server v2
ssh server vrf default
ssh server netconf vrf default
ssh server rate-limit 600
ssh server session-limit 1024
netconf-yang agent ssh
```

Cisco IOS XR devices that are onboarded through SNMP must have SNMP enabled on the device. The following is an example of an SNMP configuration on a Cisco IOS XR device:

```
snmp-server community public RO
```

Please note that, currently, Cisco Crosswork Optimization Engine does not itself support execution of EXEC privilege commands, such as **enable**, on devices. These types of commands must be executed using the device console or other means.

About Collection Jobs



Note

This section describes Crosswork Data Gateway features that are supported by various products in the Crosswork Network Automation suite. Please note that Cisco Crosswork Optimization Engine applications are, by default, automatically subscribed for required SNMP and MDT data collection. Cisco Crosswork Optimization Engine receives the data to internal Kafka and creates new collection jobs. Keep in mind that Crosswork Data Gateway extended collection features, MIBs, and other data described in this section may not apply to Cisco Crosswork Optimization Engine.

As mentioned earlier, Crosswork Data Gateway pulls functional images from the Crosswork. Each functional image represents a collection type. You can create multiple jobs for a given collection type. A collection job describes what task a Crosswork Data Gateway is expected to perform. Crosswork receives the data collection requests via these collection jobs and assigns to a Crosswork Data Gateway instance to serve the request.

You can collect more than one type of data at a time by using separate collection jobs.

For each collection job you create, Crosswork Data Gateway executes the collection request and deposits the collected data in the preferred data destination(s).

Crosswork Data Gateway lets you create three types of collection jobs:

CLI Collection Job

Enables CLI-based data collection (such as device configuration) from the network devices. The CLI collector uses XDE/PAL to collect device data for a given CLI. Only **show** commands are supported for this type of collection job.

SNMP Collection Job

Enables SNMP-based data collection based on the OIDs supported on the devices.

Supported SNMP versions include SNMPv1, SNMPv2c, and SNMPv3 for data polling and traps.

MDT Collection Job

Collects model driven telemetry data streamed from the device to the Crosswork Data Gateway.



-
- Note**
1. Crosswork Data Gateway drops incoming southbound traffic if there is no corresponding (listening) collection job request for the same. It also drops data/SNMP traps received from an unsolicited device (i.e., not attached to Crosswork Data Gateway). Crosswork Data Gateway records this in log and notifies Crosswork.
 2. Polled data cannot be requested from the device until Crosswork Data Gateway is ready to process and transmit the data. If it cannot keep up with the amount of data, it sends an error to northbound interface indicating when the throttling began and condition cleared.
 3. The status of a collection job shows as **Unknown** if the job has not received any data. For example, collection jobs for traps will show **Unknown** when traps are not configured on the device or when there are no changes to trigger a traps notification for the device.
-

Collection Jobs

This section contains sample collection job payloads for the following collection profiles:

- [CLI Collection Job, on page 5](#)
- [SNMP Collection Jobs, on page 6](#)
- [MDT Collection Job, on page 13](#)

CLI Collection Job

Crosswork Data Gateway supports CLI-based data collection from the network devices. It uses XDE/PAL to collect device data for a given CLI. Only show commands are supported for this type of collection job.



-
- Note**
- The initial status for all the collection jobs in the UI is Unknown. Upon receiving a CLI collection job, Cisco Crosswork Data Gateway performs basic validations on it. If the collection job is valid, its status changes to Successful, else it changes to Failed.
 - Device should not have any banner configuration for CLI collection to work properly. Please refer to device documentation on how to turn this off.
 - The value of **Cadence** is in seconds. It should be set either to 0 to indicate the sensor configured to be collected only once.
OR
It should be ≥ 60 (i.e. at least 1 minute) up to 2764800 seconds (i.e. at most 32 days) max, indicating how frequently configured sensor data should be collected.
 - When collection from a device is skipped due to previous execution still in progress, Cisco Crosswork Data Gateway raises a warning log. No alert is generated for this scenario.
-

Following is a CLI collection job sample:

```
{
  "collection_job": {
    "application_context": {
      "context_id": "collection-job1",
      "application_id": "APP1"
    },
    "collection_mode": {
      "lifetime_type": "APPLICATION_MANAGED",
      "collector_type": "CLI_COLLECTOR"
    },
    "job_device_set": {
      "device_set": {
        "devices": {
          "device_ids": [
            "658adb03-cc61-448d-972f-4fcec32cbfe8"
          ]
        }
      }
    },
    "sensor_input_configs": [
      {
        "sensor_data": {
          "cli_sensor": {
            "command": "show platform"
          }
        },
        "cadence_in_millisec": "tel:60000"
      }
    ],
    "sensor_output_configs": [
      {
        "sensor_data": {
          "cli_sensor": {
            "command": "show platform"
          }
        },
        "destination": {
          "destination_id": "1e71f2fb-ea65-4242-8efa-e33cec71b369",
          "context_id": "topic1"
        }
      }
    ]
  }
}
```

SNMP Collection Jobs

Crosswork Data Gateway supports SNMP-based data collection based on the OIDs supported on the devices.

The SNMP collector makes a poll request to Crosswork to get its configuration profile (a list of MIB objects to collect and a list of devices to fetch from). It determines the corresponding OIDs by looking up the pre-packaged list of MIB modules or the custom list of MIB modules.



Note MIBs are required only if the collection request references MIB TABLE names or SCALAR names. However, if the requests are OID-based, then MIBs are not required.

Once the OIDs are resolved, they are provided as input to the SNMP collectors.

The device packages can be imported into the Crosswork Data Gateway VM as described in Section [Add a Custom Software Package](#).

The following SNMP versions are supported:

- SNMPv1
- SNMPv2c
- SNMPv3

The table below lists supported privacy protocols and the value that needs to be given in the collection payload for SNMP and SNMP Trap collection jobs:

Protocol	SNMP Collection Payload	SNMP Trap Collection Payload
aes	AES	N/A
des56	DES	DES
3des	3DES	3DES
aes 128	AES128	AES128
aes 192	AES192 or CiscoAES192(Cisco specific)	AES192 or CiscoAES192(Cisco specific)
aes 256	AES256 or CiscoAES256(Cisco specific)	AES256 or CiscoAES256(Cisco specific)



Note

- The initial status for all the collection jobs in the UI is Unknown. Upon receiving a SNMP collection job, Cisco Crosswork Data Gateway performs basic validations on it. If the collection job is valid, its status changes to Successful, else it changes to Failed.
- The value of **Cadence** is in seconds. It should be set either to 0 to indicate the sensor configured to be collected only once.
OR
It should be ≥ 60 (i.e. at least 1 minute) up to 2764800 seconds (i.e. at most 32 days) max, indicating how frequently configured sensor data should be collected.
- When collection from a device is skipped due to previous execution still in progress, Crosswork Data Gateway raises a warning log. No alert is generated for this scenario.
- For SNMP v1/v2c, if the device details (such as host or community string) are incorrect in the payload, Crosswork Data Gateway ignores the traps received from the device and logs the a WARN message.

In case of SNMP v3, if the device details (such as auth, priv, and security name details) are incorrect in the payload, Crosswork Data Gateway filters it out and hence, does not receive the trap. Thus, no WARN message is logged.

Sample Configurations on Device:

Version	Configuration
V1	<pre>snmp-server group group1 v1 snmp-server user user1 group1 v1 snmp-server host <host_ip> traps <community_string> udp-port 1062</pre> <p>For example,</p> <pre>snmp-server host 172.29.194.78 traps test udp-port 1062</pre> <p>Note Version 1 is the default version used by the device.</p>
V2c	<pre>snmp-server group group1 v2c snmp-server user user1 group1 v2c snmp-server host 172.29.194.142 traps version 2c v2test udp-port 1062</pre>
V3	<pre>snmp-server group group1 v3 auth notify user1 read user1 write user1 snmp-server view user1 1.3 included snmp-server user user1 group1 v3 auth md5 <password> priv aes 128 <password> snmp-server host 172.23.92.193 traps version 3 priv user1 udp-port 1062</pre>

The SNMP Collector supports the following operations:

- SCALAR
- TABLE



Note For TABLE operation, you can either provide a Table OID or a Column OID.

- MIB_WALK
- TRAP
- DEVICE_PACKAGE

These operations are defined in the sensor config (see payload sample below).



Note There is an optional **deviceParams** attribute **snmpRequestTimeoutMillis** (not shown in the sample payloads) that should be used if the device response time is very high. It's not recommended to use **snmpRequestTimeoutMillis** unless you are absolutely certain that your device response time is very high.

The value for **snmpRequestTimeoutMillis** should be specified in milliseconds:

Default value is 1500 milliseconds

Minimum value is 1500 milliseconds

However, there is no limitation on the maximum value of this attribute.

Following is an SNMP collection job sample:

```
{
  "collection_job": {
    "application_context": {
      "context_id": "collection-job1",
      "application_id": "APPL"
    },
    "collection_mode": {
      "lifetime_type": "APPLICATION_MANAGED",
      "collector_type": "SNMP_COLLECTOR"
    },
    "job_device_set": {
      "device_set": {
        "devices": {
          "device_ids": [
            "c70fc034-0cbd-443f-ad3d-a30d4319f937",
            "8627c130-9127-4ed7-ace5-93d3b4321d5e",
            "c0067069-c8f6-4183-9e67-1f2e9bf56f58"
          ]
        }
      }
    },
    "sensor_input_configs": [
      {
        "sensor_data": {
          "snmp_sensor": {
            "snmp_mib": {
              "oid": "1.3.6.1.2.1.1.3.0",
              "snmp_operation": "SCALAR"
            }
          }
        },
        "cadence_in_millisecc": "60000"
      },
      {
        "sensor_data": {
          "snmp_sensor": {
            "snmp_mib": {
              "oid": "1.3.6.1.2.1.31.1.1",
              "snmp_operation": "TABLE"
            }
          }
        },
        "cadence_in_millisecc": "60000"
      }
    ],
    "sensor_output_configs": [
      {
```

```

    "sensor_data": {
      "snmp_sensor": {
        "snmp_mib": {
          "oid": "1.3.6.1.2.1.1.3.0",
          "snmp_operation": "SCALAR"
        }
      }
    },
    "destination": {
      "destination_id": "4c2ab662-2670-4b3c-b7d3-b94acba98c56",
      "context_id": "topic1_461cb8aa-a16a-44b8-b79f-c3daf3ea925f"
    }
  },
  {
    "sensor_data": {
      "snmp_sensor": {
        "snmp_mib": {
          "oid": "1.3.6.1.2.1.31.1.1",
          "snmp_operation": "TABLE"
        }
      }
    },
    "destination": {
      "destination_id": "4c2ab662-2670-4b3c-b7d3-b94acba98c56",
      "context_id": "topic2_e7ed6300-fc8c-47ee-8445-70e543057f8a"
    }
  }
]
}

```

SNMP Traps Collection Job

SNMP traps are handled in a similar manner. Trap listeners listen on a port and then dispatch data to recipients (based on their topic of interest).



Note

- Device should have been pre-configured by the traps.
- Crosswork Data Gateway listens on UDP port 1062 for Traps.
- If the collection job is invalid, there is missing configuration on the device, or no trap is received, the status of the job remains "Unknown".

On receiving a trap, Crosswork Data Gateway does the following validations:

1. Check if any collection job is created for the device.
2. Checks the trap version and community string.
3. For SNMP v3, validates for user auth and priv protocol and credentials.

Following is an SNMP-Trap collection job sample:

```

{
  "collection_job": {
    "application_context": {
      "context_id": "collection-job1",
      "application_id": "APP1"
    }
  },

```

```

"collection_mode": {
  "lifetime_type": "APPLICATION_MANAGED",
  "collector_type": "TRAP_COLLECTOR"
},
"job_device_set": {
  "device_set": {
    "devices": {
      "device_ids": [
        "a9b8f43d-130b-4866-a26a-4d0f9e07562a",
        "8c4431a0-f21d-452d-95a8-84323a19e0d6",
        "eaab2647-2351-40ae-bf94-6e4a3d79af3a"
      ]
    }
  }
},
"sensor_input_configs": [
  {
    "sensor_data": {
      "trap_sensor": {
        "path": "1.3.6.1.6.3.1.1.4"
      }
    },
    "cadence_in_millisecc": "60000"
  }
],
"sensor_output_configs": [
  {
    "sensor_data": {
      "trap_sensor": {
        "path": "1.3.6.1.6.3.1.1.4"
      }
    },
    "destination": {
      "destination_id": "4c2ab662-2670-4b3c-b7d3-b94acba98c56",
      "context_id": "topic1_696600ae-80ee-4a02-96cb-3a01a2415324"
    }
  }
]
}

```

Enabling Traps forwarding to external applications

As per the current implementation, in case of an SNMP Trap collection job, all traps are sent to the specified data destination even if the SNMP Trap OID is not provided in the sensor path.

Therefore, it is recommended to have a single SNMP Trap collection job per device (with any OID as sensor path) as it would be enough to get all traps from that device.



Note It is also recommended to selectively enable on the device only those traps that are needed by Crosswork.

To identify the type of trap from the data received on the destination, look for *oid* (OBJECT_IDENTIFIER, for example, 1.3.6.1.6.3.1.1.4.1.0) and *strValue* associated to the *oid* in the OidRecords (application can match the OID of interest to determine the kind of trap).

Below are some sample values and a sample payload:

- Link up

```
1.3.6.1.6.3.1.1.4.1.0 = 1.3.6.1.6.3.1.1.5.4
```

- Link Down

```
1.3.6.1.6.3.1.1.4.1.0 = 1.3.6.1.6.3.1.1.5.3
```

- Syslog

```
1.3.6.1.6.3.1.1.4.1.0 = 1.3.6.1.4.1.9.9.41.2.0.1
```

- Cold Start

```
1.3.6.1.6.3.1.1.4.1.0 = 1.3.6.1.6.3.1.1.5.1
```

```
{
  "nodeIdStr": "BF5-XRV9K1.tr3.es",
  "nodeIdUuid": "C9tZ5lJoSJKf5OZ67+U5JQ==",
  "collectionId": "133",
  "collectionStartTime": "1580931985267",
  "msgTimestamp": "1580931985267",
  "dataGpbkv": [
    {
      "timestamp": "1580931985267",
      "name": "trapsensor.path",
      "snmpTrap": {
        "version": "V2c",
        "pduType": "TRAP",
        "v2v3Data": {
          "agentAddress": "172.70.39.227",
          "oidRecords": [
            {
              "oid": "1.3.6.1.2.1.1.3.0",
              "strValue": "7 days, 2:15:17.02"
            },
            {
              "oid": "1.3.6.1.6.3.1.1.4.1.0", // This oid is the Object Identifier.
              "strValue": "1.3.6.1.6.3.1.1.5.3" // This is the value that determines the
kind of trap.
            },
            {
              "oid": "1.3.6.1.2.1.2.2.1.1.8",
              "strValue": "8"
            },
            {
              "oid": "1.3.6.1.2.1.2.2.1.2.8",
              "strValue": "GigabitEthernet0/0/0/2"
            },
            {
              "oid": "1.3.6.1.2.1.2.2.1.3.8",
              "strValue": "6"
            },
            {
              "oid": "1.3.6.1.4.1.9.9.276.1.1.2.1.3.8",
              "strValue": "down"
            }
          ]
        }
      }
    }
  ],
  "collectionEndTime": "1580931985267",
  "collectorUuid": "YmNjZjEzMTktZjFLOS00NTE5LWI4OTgtY2Y1ZmQxZDFjNWExOlRSQVBFQ09MTEVDVE9S",
  "status": {
    "status": "SUCCESS"
  },
  "modelData": {}
}
```

```
"sensorData": {
  "trapSensor": {
    "path": "1.3.6.1.6.3.1.1.5.4"
  }
},
"applicationContexts": [
  {
    "applicationId": "APP1",
    "contextId": "collection-job-snmp-traps"
  }
]
}
```

MDT Collection Job

Crosswork Data Gateway supports data collection from network devices using Model-driven Telemetry (MDT) to consume telemetry streams directly from devices (for IOS-XR based platforms only).



Note

- MDT collector retains the collection ID that comes as part of the telemetry proto for the device. This behavior is different from CLI and SNMP collectors which compute the collection ID based on the sequence number of the collection.
- MDT collection jobs require some configuration to be done on the device. This configuration is automatically taken care of by NSO.
- If there is some change (delete/update) in existing MDT jobs between backup and restore operations, Crosswork does not replay the jobs for config update on the devices as it involves Provider(NSO). You have to restore configs on provider/devices. Crosswork will just restore the jobs in database.

Monitoring Collection Jobs

Once a device is mapped to a Cisco Crosswork Data Gateway instance, the status of all the associated collection jobs is set to 'Unknown'. A job could have status as 'Unknown' for either of the following reasons:

- Cisco Crosswork Data Gateway has not yet reported its status.
- Loss of connection between Cisco Crosswork Data Gateway and Crosswork.
- Cisco Crosswork Data Gateway received the collection job, but actual collection is still pending.

After the collection job is processed, the status changes to 'Successful' if the processing was successful or else it changes to 'Failed'.

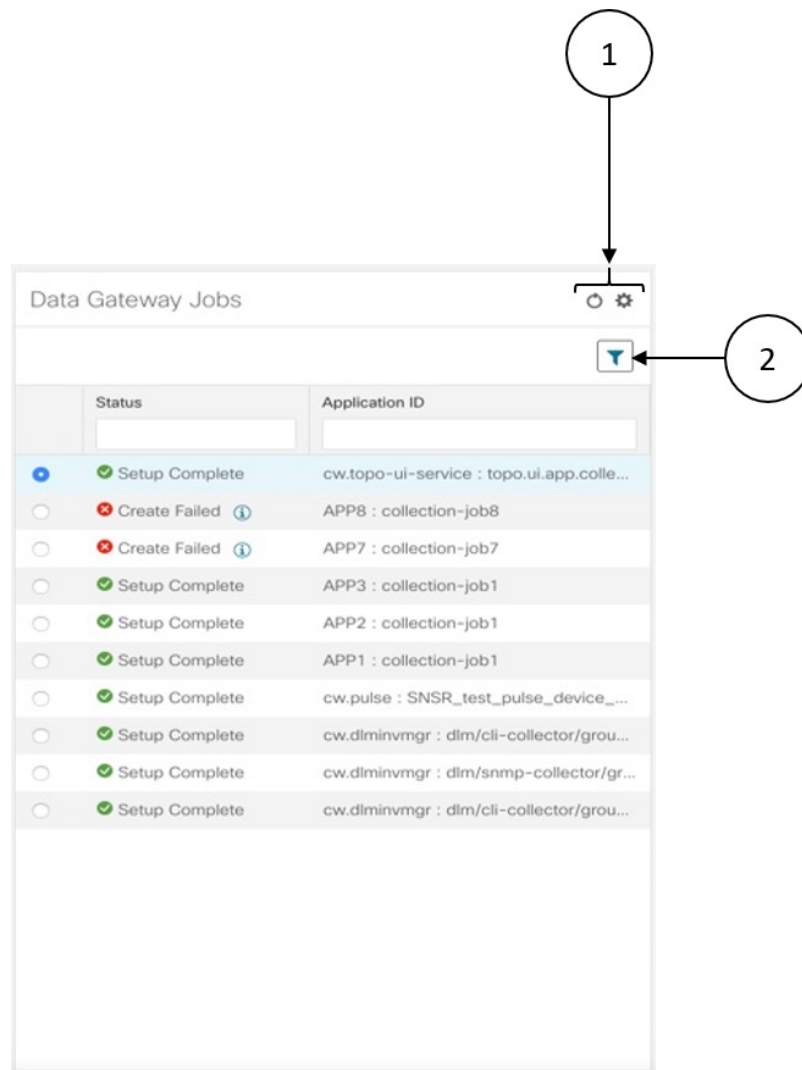
From the **Collection Jobs** view, you can monitor the status of the collection jobs currently active on all the Cisco Crosswork Data Gateway instances enrolled with Cisco Crosswork Optimization Engine, such as system jobs and API-defined collection jobs.




From the navigation bar, choose **Admin > Collection Jobs**.

Figure 1: Add Optima image here 442649.jpg

Item	Description
Data Gateway Jobs Pane	Shows the list of all active collection jobs along with their status and application ID.
Job Details Pane	Shows the details of a particular job selected in the Data Gateway Jobs pane.

To view details of a collection job, select the collection job from the **Data Gateway Jobs** pane. The details of the selected job are displayed in the **Job Details** pane right next to the **Data Gateway Jobs** pane.



Item	Description
1	Click  to refresh the Data Gateway Jobs window.
	Click  to choose the columns to make visible in the Data Gateway Jobs window (see Set, Sort and Filter Table Data).
2	Click  to show/hide the quick filters.
	Click the Clear All Filters link to clear any filter criteria you may have set.

Data Gateway Jobs pane displays only the status and application ID.




Note

- `Create Failed` error means out of N devices , some devices setup failed. However, the collection would happen on the other devices. You can identify the device(s) causing this error by using `Control Status API`.
- If job creation failed on a particular device because of NSO errors, after fixing NSO errors , you have to manually change the administration state of the device first to "Down" and then "Up". However, doing so resets the collection on the device.

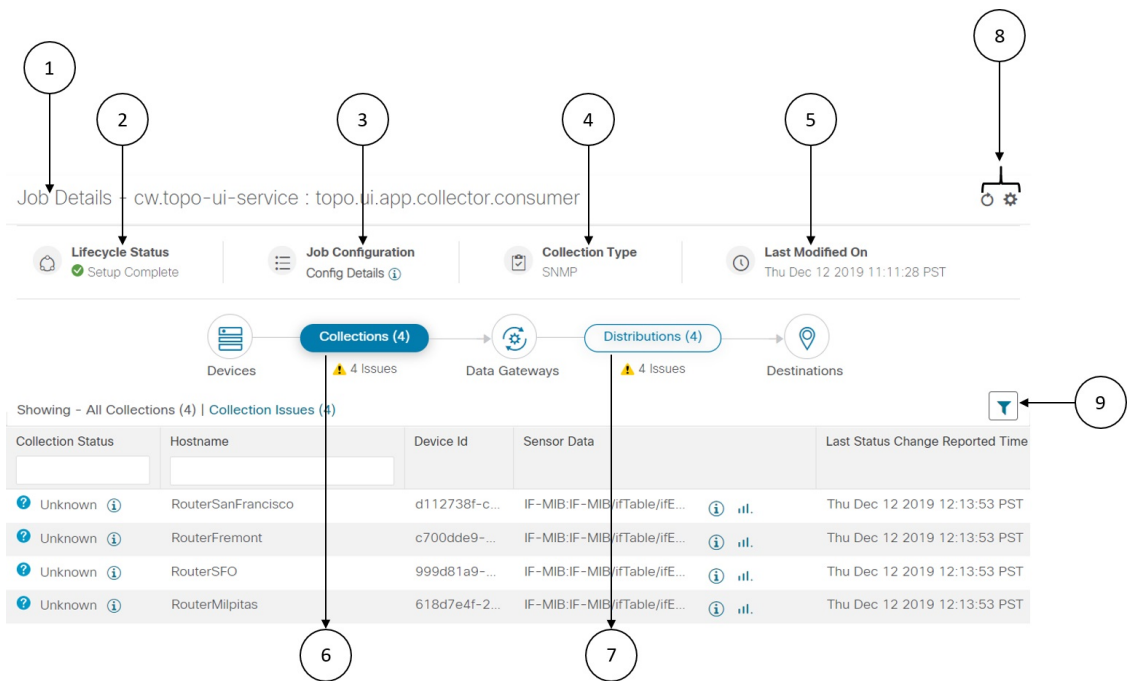





Note


Create/Delete failed errors are shown in a different screen pop up. Click  next to the job status to see details of the error.

- You may also try recreating the job using PUT collection job API with the same payload.




However, when you select a job, more details are displayed in the **Job Details** pane:



Item	Description
1	Application name and context associated with the collection job.
2	Lifecycle status of the collection job.
3	Job payload of the collection job that you pass in the REST API request. Click  icon next to Config Details to view the job configuration. Crosswork Data Gateway lets you view configuration in two modes: <ul style="list-style-type: none"> • View Mode • Text Mode
4	Collection Type
5	Time and date of last modification of the collection job.
6	Collections (x): x refers to requested input collections that span device by sensor paths. The corresponding (y) Issues is the count of input collections that are in UNKNOWN or FAILED state.
7	Distributions (x): x refers to requested output collections that span device by sensor paths. The corresponding (y) Issues is the count of output collections that are in UNKNOWN or FAILED state.
8	Click  to refresh the Job Details window. Click  to choose the columns to make visible in the Job Details window (see Set, Sort and Filter Table Data).

Item	Description
9	Click  to show/hide the quick filters.
	Click the Clear All Filters link to clear any filter criteria you may have set.

Job Details pane displays the following details about a collection job:

Field	Description
Collection/Distribution Status	Status of the collection/distribution. It is reported on a on change basis from Cisco Crosswork Data Gateway. Click  next to the collection/distribution status for details.
Hostname	Device hostname with which the collection job is associated.
Device Id	Unique identifier of the device from which data is being collected.
Sensor Data	<p>Sensor path</p> <p>Click  to see collection/distribution summary. From the sensor data summary pop up you can copy the sensor data by clicking Copy to Clipboard.</p> <p>and</p> <p>Click  to see collection/distribution metrics summary. The metrics are reported on cadence-basis i.e., once every 10 minutes by default. It shows the following metrics for a collection:</p> <ul style="list-style-type: none"> • last_collection_time_msec • total_collection_message_count • last_device_latency_msec • last_collection_cadence_msec <p>It shows the following metrics for a collection:</p> <ul style="list-style-type: none"> • total_output_message_count • last_destination_latency_msec • last_output_cadence_msec • last_output_time_msec • total_output_bytes_count

Field	Description
Last Status Change Reported Time	Time and date on which last status change was reported for that device sensor pair from Cisco Crosswork Data Gateway.