



## **Cisco Crosswork Workflow Manager 1.2 Get Started Guide**

**First Published:** 2024-09-27

### **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





## CHAPTER 1

# What is CWM

---

- [What is Crosswork Workflow Manager?, on page 1](#)

## What is Crosswork Workflow Manager?

Cisco Crosswork Workflow Manager is a tool that simplifies and automates complex network operations and other business processes. It provides a centralized platform for creating, managing, and executing workflows, allowing for manual operator intervention during workflow execution while ensuring repeatability and fault-tolerance. Workflows are defined using a standardized Domain Specific Language based on the Serverless Workflow specification, enabling workflow designers to express complex business processes, dependencies, and decision logic in a unified and readable format.





## CHAPTER 2

# Core concepts

---

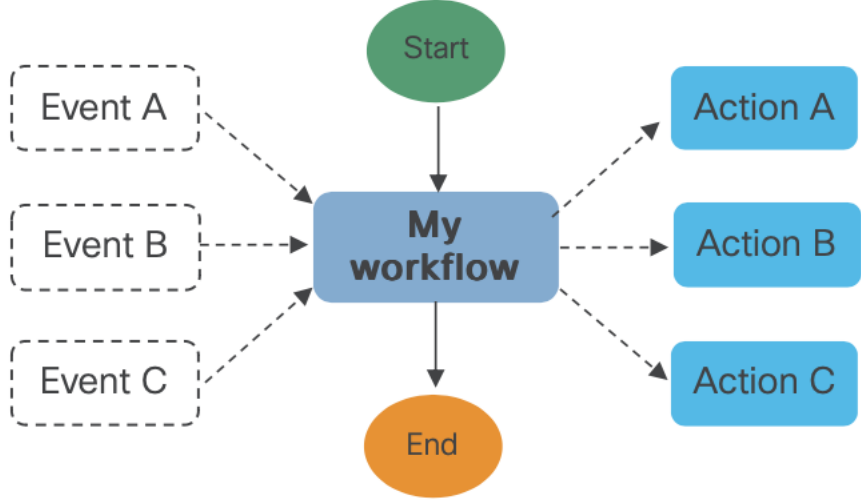
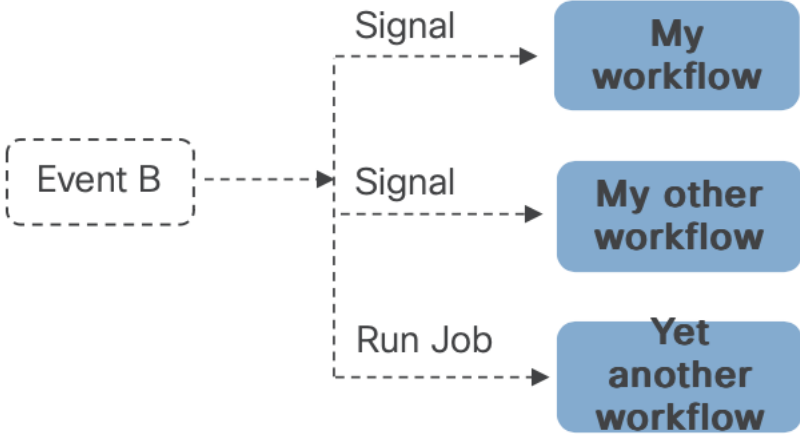
The section contains the following topics:

- [Core concepts, on page 3](#)

## Core concepts

This section defines the main concepts and components used in the Crosswork Workflow Manager (CWM) that help understand how the platform works and its features. You can familiarize yourself with them to ease the very first steps with CWM.

Concept	Description
Activity	An activity is a function that executes a single, well-defined action against the target, outside system (whether it's an app or solution). Activities are defined in adapters and allow communication with an outside system.
Adapter	Adapters are responsible for communication with external services, like other applications, systems or environments. The adapters define and expose activities that are consumed by workflow definitions. Every adapter can be associated with the worker that will execute the adapter activities.
Adapter SDK	The adapter SDK automatically generates the structure for the required adapter's components. Developers can further define activities that are needed and then extend the integrations with the client environment.
CWM UI	The CWM UI is a graphical user interface of CWM that allows users to interact with the system and gives access to its core functionalities.

Concept	Description
Event	<p data-bbox="457 283 1516 409">Events are signals coming from external sources that workflows run by CWM can interact with. For the CWM 1.1 version, support for interaction with external Kafka, AMQP and HTTP brokers is added. This means that events can be either consumed or produced by an instantiated workflow (a job). A workflow can listen on one or multiple events and consume them to trigger an action/actions:</p>  <pre data-bbox="548 485 1403 982"> graph TD     Start((Start)) --&gt; MW[My workflow]     EA[Event A] -.-&gt; MW     EB[Event B] -.-&gt; MW     EC[Event C] -.-&gt; MW     MW --&gt; End((End))     MW -.-&gt; AA[Action A]     MW -.-&gt; AB[Action B]     MW -.-&gt; AC[Action C] </pre> <p data-bbox="457 1066 1101 1098">can also be initiated by an event that comes in to the system:</p>  <pre data-bbox="526 1157 1321 1591"> graph LR     EB[Event B] -.-&gt; S1[Signal]     EB -.-&gt; S2[Signal]     EB -.-&gt; RJ[Run Job]     S1 -.-&gt; MW[My workflow]     S2 -.-&gt; MOW[My other workflow]     RJ -.-&gt; YAW[Yet another workflow] </pre>
Execution engine	<p data-bbox="457 1669 1471 1726">CWM has an internal worker called the execution engine. It enables the execution of workflow definition. This worker is not visible in the CWM UI.</p>

Concept	Description
Job	A job represents the single execution of a particular workflow definition. To be able to run a job in Crosswork Workflow Manager, you need first to add your workflow definition to CWM. Running a new job instantiates a workflow definition stored in CWM. Before starting a job run, you enter the initial start data (Input variables). It means that your workflow executions are isolated and may use different data than other executions of the same workflow definition.
Job event	Events are created during workflow execution based on the occurrences defined in the workflow definition. All events that happened during workflow execution are recorded in the Job Event Log table in CWM UI.
Schedule	Scheduling a job allows you to define when a workflow execution should start at the predefined date and time in the future, once or on a recurring basis. You can create a scheduled job via CWM UI or API, but currently some of the schedule functionalities, like editing or pausing/unpausing the schedule are available only via API. Each scheduled job is a separate entity and has a unique Run ID, although all scheduled runs in a given schedule share the same Schedule ID.
Workflow	Workflows help you capture, organize and automate processes with repeatable actions performed in a specified order. In the context of CWM, documentation differentiates between: workflow definition: piece of code written in JSON or YAML, based on the Serverless Workflow Specification and vendor-neutral, domain-specific language. workflow execution (job): single execution of a workflow definition.
Workflow engine	The workflow engine manages the way how your workflow definitions are interpreted and conducted. It receives events, schedules tasks, and manages the execution of workflows.
Worker	Workers carry out the workforce and are responsible for executing the workflow definition code, relevant adapter code and activities defined in the workflow definition. Depending on your needs and scale, you can have multiple workers for every workflow definition. Your worker can be associated with one adapter and its activities or with multiple ones.







## CHAPTER 3

# Examples

---

This section contains the following topics:

- [Run example workflow using Cisco NSO adapter, on page 7](#)
- [Combine NetBox, NSO and Webex in a workflow, on page 15](#)

## Run example workflow using Cisco NSO adapter

This quick start uses a locally installed [Cisco Crosswork Network Service Orchestrator](#) application and the CWM with the Cisco NSO adapter to show you a basic use case scenario for creating and running a successful workflow. It will guide you through how to install an adapter, create a worker for the workflow execution and run the created workflow to quickly get tangible results in Cisco NSO.

### Workflow overview

The purpose of the example workflow is to automatically create a VPN service for two NSO devices.

First, we point to the devices in the data input and then try to perform the NSO `check-sync` operation on them. Then, depending on the result:

- if not in sync, we push a device to perform a `sync-from`, and only then try to create a VPN for it;
- if in sync, we don't perform `sync-from` but directly create a VPN for the device.

If all the steps are executed successfully, the execution engine reports workflow execution completion and displays the final data input. The results are visible in NSO too. If the engine encounters errors while performing a step, it uses the specified `retry` policy. In case errors persist beyond the retry limits, the engine ends the execution with a **Failed** status.

Go through the sections below to learn the details of how data input, functions, states, actions, and data filters are defined. If you want to know how the sausage is made, you can read the Create workflow chapter in the Create Workflows guide.

### Prerequisites

- Cisco NSO 6.0 local install. If you don't have it, follow the [installation instructions](#).
- CWM installed using OVA. Go to CWM Administrator guide for instructions.

## Step 1: Install NSO adapter

To interact with Cisco NSO, CWM needs a dedicated Cisco NSO adapter. Here's how you install it using the CWM API:

### Upload NSO adapter file

#### Procedure

---

- Step 1** Get the latest NSO adapter installation file from the CWM Software Package.
- Step 2** Go to the CWM User Interface in a browser, and log in using credentials generated upon the installation of CWM.
- Step 3** Navigate to the **Admin** -> **Adapters** tab.
- Step 4** Click **Add Adapter**.
- Step 5** In the **Install a new adapter** modal, click on the file uploader to select an `tar.gz` installable archive from your local machine and click **Upload**.
- Step 6** After the adapter file is uploaded to the CWM database, tick **Automatically create worker for this adapter** checkbox if you want to create one, then click **Install Adapter** to finish the installation process.

#### Note

If you want to create a worker manually, follow the instructions in the Operator Guide. Remember that in this case, you will need to update the workflow definition with your created worker name.

- Step 7** In the adapter list, click on the name of your adapter to enter its details. Tick the **Use as default version for associated activities** checkbox.
- 

## Step 2: Create secret and resource

To define the resources and secrets to be passed in securely to the Cisco NSO adapter, you need to create a secret and resource in CWM. Here's how to do it:

### Create secret

#### Procedure

---

- Step 1** In CWM, navigate to the **Admin** -> **Secrets** tab.
- Step 2** Click **Add Secret**.
- Step 3** In the **New secret** view, specify the following:
  - a) Secret ID: `NSOSecret`
  - b) Secret type: `basicAuth`
- Step 4** After selecting the secret type, a set of additional fields is displayed under the Secret type details section. Fill in the fields with the following:
  - a) password: `admin`(or your custom password)

- b) username: `admin`(or your custom username)

**Step 5** Click **Create Secret**.

---

## Create resource

### Procedure

---

**Step 1** In CWM, navigate to the **Admin** -> **Resources** tab.

**Step 2** Click **Add Resource**.

**Step 3** In the **New resource** window, specify the following:

- a) Resource name: `NSOLocal`
- b) Resource type: `cisco.nso.resource.v1.0.0`
- c) Secret ID: `NSOSecret`
- d) Connection:
  - Host: `127.0.0.1` (or, replace with the address where you host the NSO instance)
  - Port: `8080` (or, replace with the port where the NSO web UI is available)
  - Scheme: `http`
  - Timeout: `60`
  - Allow Insecure: `true`

**Step 4** Click **Create resource**.

---

## Step 3: Set up NSO example service

The NSO example that we use for the purposes of our workflow is setting up a Layer3 VPN in a service provider MPLS network for two NSO-simulated devices. Here's how you set up the example:

### Procedure

---

**Step 1** In a terminal, open your main NSO directory and go to `mpls-vpn-new-template`:

```
cd examples.ncs/service-provider/mpls-vpn-new-template
```

**Step 2** Execute the Makefile by running:

```
make stop clean all start
```

This command will start your local NSO instance and the sample netsim devices.

**Step 3** For the example workflow to execute successfully, execute a **Sync from** on all the netsim devices beforehand:

a) Log in to the CLI as admin:

```
ncs_cli -C -u admin
```

b) Run `sync-from`:

```
devices sync-from
```

---

## Step 4: Run the workflow

Now that we have the NSO adapter, the worker, and the NSO example all up and running, we can create a workflow in the CWM UI and run the job.

### Add new workflow

#### Procedure

---

**Step 1** In the CWM UI, select the **Workflows** tile from the navigation menu on the left.

**Step 2** In the **Workflows** panel, click **Create new workflow**.

**Step 3** In the **Create new workflow** modal, provide the required input:

a) **Workflow definition name** - provide the name for the example workflow definition: `CreateL3VPN`.

b) **Version** - provide workflow definition version: `1.0`.

**Step 4** Click **Create workflow**.

Figure 1: Add workflow

---

## Create new workflow

Workflow definition name\*

Version\*

**Create Workflow** **Cancel**

---

## Run job

### Procedure

---

- Step 1** In the **Workflows** panel, enter the newly created workflow definition by clicking its name.
- Step 2** Click the **Code** tab and delete the sample content from the **Code** field.
- Step 3** Download the workflow definition from the link below, copy it, and paste inside the **Code** field, then click **Save changes**.  
[https://www.cisco.com/c/dam/en/us/td/docs/net\\_mgmt/cisco\\_workflow/workflow\\_v1-1.zip](https://www.cisco.com/c/dam/en/us/td/docs/net_mgmt/cisco_workflow/workflow_v1-1.zip)
- Step 4** Click **Run**.
- Step 5** In the **Run job** modal, provide a name for the job and in the **Input variables** field, paste the data input from the section below inside the brackets:
- ```
"device0Name": "ce0",  
"device1Name": "ce1",  
"nsoResource": "NSOLocal"
```
- Step 6** Click **Run job**.

*Figure 2: Run job*

---

## Step 5: Check results

### In CWM UI

#### Procedure

---

- Step 1** In the CWM UI, select the **Job Manager** tile from the navigation menu on the left.
- Step 2** In the **All jobs** tab, find your job and check the status of the workflow execution in the **Status** table column.
- If the workflow is executed correctly, a green tick with **Completed** status will be visible.
  - If the workflow execution is still in progress or the engine is retrying an action, a blue label with the **Running** status will be displayed.
- Step 3** Click the job name to enter its details.
- Step 4** In the **Job Event Log** table, expand the bottommost **WorkflowExecution** entry by clicking its name.
- Step 5** In the JSON payload displayed, find the *data* key. It presents the final data output updated by the successful execution of the workflow actions for which `toStateData` inside the `actionDataFilter` was defined:

Figure 3: Job event log

## Job Event Log

Full event history in JSON format

| Job Event Name       | Job Event Type       | Status                                                                         | Attempts             |
|----------------------|----------------------|--------------------------------------------------------------------------------|----------------------|
| <input type="text"/> | <input type="text"/> | <input type="button" value="All se"/> <span style="font-size: 1.2em;">▼</span> | <input type="text"/> |

```

    "result": {
      "payloads": [
        {
          "metadata": {
            "encoding": "anNvbi9wbGFpbg=="
          },
          "data": {
            "checkSyncResult0": "in-sync",
            "checkSyncResult1": "in-sync",
            "createServiceResult": 201,
            "device0Name": "ce0",
            "device1Name": "ce1"
          }
        }
      ],
      "workflowTaskCompletedEventId": "22"
    }
  ]
  
```

## In NSO

### Procedure

- Step 1** Log in to your NSO account and in the **Application hub** view, click the **Service manager** tile.



**Step 2** From the **Select service points** drop-down, select `/l3vpn:vpn/l3vpn`.

**Step 3** In the table, find `testnetwork` and click the **devices** arrow to see that your netsim devices `ce0` and `ce1` now belong to the `testnetwork` together with a `pe0` device.

Figure 4: NSO VPN test network

The screenshot shows the Cisco Service Manager interface. At the top left is the Cisco logo and the text "Service manager NSO VERSION:6.0". Below this is a breadcrumb navigation bar showing the path "/l3vpn:vpn/l3vpn" with expand/collapse and list view icons, and a "0 / 2" indicator. The main content area is titled "services in /l3vpn:vpn/l3vpn" and contains a table with the following data:

| <input type="checkbox"/> | name        | devices | check |
|--------------------------|-------------|---------|-------|
| <input type="checkbox"/> | testnetwork | 3 ▲     | che   |

## Combine NetBox, NSO and Webex in a workflow

With this workflow example, we'll be exploring the possibilities of using multiple external services in a workflow and calling them by means of separate subworkflows (called *child workflows*) for each service. The aim of the workflow is to automatically allocate subnet prefixes in NetBox, spin up a VPN service instance in Cisco NSO with network endpoint configurations, and send a confirmation message via Cisco Webex that the workflow has completed.



**Note** The present example consists of one main workflow (parent) and three subworkflows (children). Thanks to this modularity, you can try out one child workflow separately without the need to set up and run all the others.

The aim of using child workflows is that each service (NetBox, NSO, and Webex) can be encapsulated in its own workflow logic, making debugging and updates easier. The parent workflow in turn, controls the overall process and ensures the child workflows execute in sequence. It also handles inter-workflow data passing and monitors the overall status.

## Prerequisites

- Cisco NSO 6.0 or higher with an example service set up (`mpls-vpn-new-template` or other).
- NetBox v4.1.
- Webex account with a personal room.

## Main workflow

Download the [complete workflow JSON file, all subworkflow files and input data here](#).

The main workflow not only coordinates the subworkflows to perform actions inside external services, but also fills in the sequence with auxiliary actions: after the first subworkflow allocates prefixes in NetBox, it performs a `sync from` on given NSO devices and creates an L3VPN service for them.

For the NSO part of the workflow, we'll need to install the Cisco NSO adapter and create a secret and a resource in CWM. To do it, follow the instructions below.

## Install NSO adapter

To install the Cisco NSO adapter, locate the latest adapter `tar.gz` file on your machine and follow [the steps in the Operator guide](#).

## Create secret and resource for NSO

To define the resources and secrets to be passed in securely to the Cisco NSO adapter, you need to create a secret and resource in the CWM API. Here's how to do it:

### Create NSO secret

#### Procedure

- 
- Step 1** In CWM, navigate to the **Administration** -> **Secrets** tab.
- Step 2** Click **Add Secret**.
- Step 3** In the **New secret** view, specify the following:
- Secret ID: `NSOSecret`
  - Secret type: `basicAuth`

- Step 4** After selecting the secret type, a set of additional fields is displayed under the Secret type details section. Fill in the fields with the following:
- username: `admin` (or replace with your custom username).
  - password: `admin` (or replace with your custom password).
- Step 5** Click **Create Secret**.
- 

## Create NSO resource

### Procedure

---

- Step 1** In CWM, navigate to the **Administration** -> **Resources** tab.
- Step 2** Click **Add Resource**.
- Step 3** In the **New resource** window, specify the following:
- Resource name: `NSOResource`
  - Resource type: `cisco.nso.resource.v1.0.0`
  - Secret ID: `NSOSecret`
  - Connection:
    - Host: provide the address where your NSO instance is hosted.
    - Port: provide the port on which the NSO web UI is available.
    - Scheme: `http`
    - Timeout: `60`
    - Allow Insecure: `true`
- Step 4** Click **Create resource**.
- 

## NetBox subworkflow #1

This subworkflow involves allocating a subnet in Netbox, which will subsequently be used in the configuration of an L3VPN. The communication with Netbox will use the Generic REST adapter, meaning the exact resource path and payload must be clearly defined. Specifically, this example requires a POST request to the `/api/ipam/prefixes/` endpoint in NetBox, with the prefix and description provided in the request body.

## Install Generic REST adapter for NetBox

To install the Generic REST adapter, locate the latest adapter `tar.gz` file on your machine and [follow these steps](#).

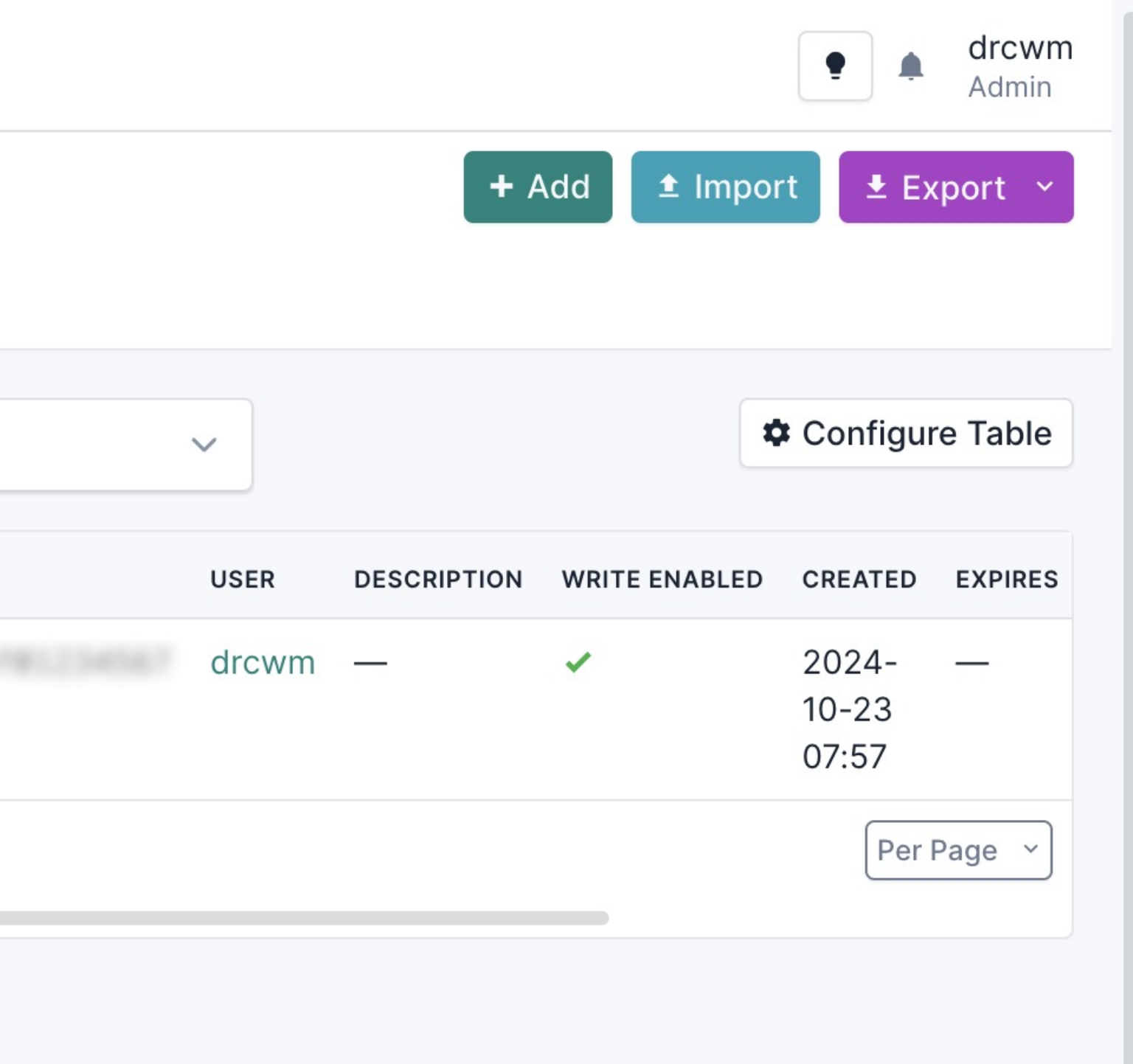
## Create NetBox secret

To authorize NetBox in CWM, you first need to retrieve the token for your NetBox installation and then add it in the secret.

## Procedure

---

- Step 1** Log in to NetBox.
- Step 2** In the left menu, click the **Admin** tab to expand it and select **API Tokens**.
- Step 3** Add a new token and copy it or, if a token exists, copy it to the clipboard.
- Step 4** Log in to CWM and navigate to the **Administration - Secrets** tab.

**Step 5** Click **Add**

Lightbulb icon, Bell icon, drcwm Admin

+ Add   ↑ Import   ↓ Export

Configure Table

| USER  | DESCRIPTION | WRITE ENABLED | CREATED          | EXPIRES |
|-------|-------------|---------------|------------------|---------|
| drcwm | —           | ✓             | 2024-10-23 07:57 | —       |

Per Page

**Step 6** In the **New secret** view, specify the following:

- Secret ID: `NetBoxSecret`
- Secret type: `token`

**Step 7** In the **token** field, provide your NetBox token that you've copied.

**Step 8** Click **Create Secret**.

---

## Create NetBox resource

### Procedure

---

**Step 1** In CWM, navigate to the **Administration - Resources** tab.

**Step 2** Click **Add Resource**.

**Step 3** In the **New resource** window, specify the following:

- a) **Resource name:** `NetBoxResource`
- b) **Resource type:** `generic.rest.resource.v1.0.0`
- c) **Secret ID:** `NetBoxSecret`
- d) **Connection:**
  - **Host:** provide the address where your NetBox instance is hosted.
  - **Port:** provide the port on which the NetBox web UI is available.
  - **Scheme:** `http`
  - **Timeout:** `60`
  - **Allow Insecure:** `true`

**Step 4** Click **Create resource**.

---

## NSO subworkflow #2

This subworkflow uses the function `NSO.RestconfPost` to interact with the **NSO RESTCONF API**. It iterates over a collection of network endpoints to configure each one. For each endpoint, it sends a POST request with the endpoint-specific data (such as device ID, interface, IP network, and bandwidth) to the `l3vpn:vpn` resource for the VPN service that has been created by the main workflow (if you run the full example). The results of each configuration action are filtered and stored in an output collection called `endpointsConfigureResponses`.

!!! note If you only run this subworkflow separately, you need to add an L3VPN service to your NSO instance. You can do it manually in NSO CLI by running the following commands:

```
ncs_cli -C -u admin
vpn l3vpn network1
route-distinguisher 999
```

To enable communication between CWM and NSO, you need the NSO adapter, secret, and resource. If already set up during the main workflow configuration, they can be reused in the subworkflow without changes.

Follow the [steps given above](#) for a detailed instruction if you haven't configured them yet.

## Webex subworkflow #3

The purpose of this child workflow is to notify a Webex user about workflow completion and its end status to a Webex room. It consists of two actions, both using the REST.Post function to send messages to the Webex API.

First Action: Posts a message to the specified Webex room (roomId) with the text "Workflow completed."

Second Action: Posts a status message to the same room, with the content depending on the terminate flag. If terminate is true, the message is "Status: Failed"; otherwise, it's "Status: Success."

Both actions send requests to the `v1/messages/` endpoint of the Webex API and use `webex_room` as the resource configuration. The response data for each message is stored in `.webexResponse`.



---

**Note** If you run the subworkflow independently from the main workflow, you'll need a reduced version of the input data. Remember to replace the "roomId" value with your personal room ID. Follow a subsection below to learn how to retrieve it.

---

## Install Generic REST adapter for Webex

As in the case of NetBox, to communicate with Webex you need to add the Generic REST adapter and create a secret and a resource in CWM for Webex. To install the adapter, locate the latest Generic REST adapter `tar.gz` file on your machine and [follow these steps](#).

## Create Webex secret

To authorize Webex in CWM, you first need to retrieve the bearer from the Webex API and the room ID of your personal room.

### Procedure

---

- Step 1** Go to [developer.webex.com/docs/getting-started](https://developer.webex.com/docs/getting-started) and log in.
  - Step 2** From the **Your Personal Access Token** field, copy the bearer.
  - Step 3** Log in to CWM and navigate to the **Administration - Secrets** tab.
  - Step 4** Click **Add Secret**.
  - Step 5** In the **New secret** view, specify the following:
    - a) Secret ID: `webex_secret`
    - b) Secret type: `bearer`
  - Step 6** In the **bearer** field, provide your Webex bearer that you've copied.
  - Step 7** Click **Create Secret**.
-

## Create Webex resource

### Procedure

---

**Step 1** In CWM, navigate to the **Administration - Resources** tab.

**Step 2** Click **Add Resource**.

**Step 3** In the **New resource** window, specify the following:

- a) Resource name: `WebexResource`
  - b) Resource type: `generic.rest.resource.v1.0.0`
  - c) Secret ID: `webex_secret`
  - d) Connection:
    - Host: `webexapis.com`.
    - Port: `443`.
    - Scheme: `https`
    - Timeout: `60`
    - Allow Insecure: `true`
- 

## Run the main workflow

Now that we have the NSO, NetBox, and Webex all up and running, we can create a workflow in the CWM UI and run the job with the appropriate workflow input data.

## Add workflows

### Procedure

---

**Step 1** In the CWM UI, select the **Workflows** tile from the navigation menu on the left.

**Step 2** In the **Workflows** panel, click **Create new workflow**.

**Step 3** In the **Create new workflow** modal, provide the required input:

- a) **Workflow definition name** - provide the name for the example workflow definition: `NetBox_NSO_Webex_example`.
- b) **Version** - provide workflow definition version: `1.0`.



**Step 4** Click **Create**

## Create new workflow

Workflow definition name\*

Version\*

Create WorkflowCancel**Step 5** Enter the newly created workflow definition by clicking its name.**Step 6** Click the **Code** tab and delete the sample content from the **Code** field.**Step 7** Copy the downloaded workflow definition and paste it inside the **Code** field, then click **Save changes**.**Note**

Repeat this process for the remaining three subworkflows to add them.

## Run job

Before you run the job, you need to correctly fill in the value of the `roomId` key, To do that, retrieve the room ID of your personal room in Webex.

### Retrieve Webex room ID

#### Procedure

**Step 1** Go to <https://developer.webex.com/docs/getting-started> and log in.**Step 2** From the left menu, click **Full API reference** and **Rooms**.**Step 3** Select the **List rooms** endpoint and click **Run**.

**Step 4** In the **Response** field, find `My personal room` and copy its

**Step 5** Paste the Room ID in the input data as the value of the `roomId` key.

---

### Run the main workflow

You can now run the main workflow:

#### Procedure

---

**Step 1** In the **Workflows** panel, enter the newly created workflow definition by clicking its name.

**Step 2** Click **Run**.

**Step 3** In the **Run job** modal, provide a name for the job and in the **Input variables** field, paste the data input that you have updated with your Webex room ID.

**Step 4** Click **Run job**.

---

## Check results in CWM UI

#### Procedure

---

**Step 1** In the CWM UI, select the **Job Manager** tile from the navigation menu on the left.

**Step 2** In the **All jobs** tab, find your job and check the status of the workflow execution in the **Status** table column.

- a) If the workflow is executed correctly, a green tick with **Completed** status will be visible.
- b) If the workflow execution is still in progress or the engine is retrying an action, a blue label with the **Running** status will be displayed.

**Step 3** Click the job name to enter its details.

**Step 4** In the **Job Event Log** table, expand the bottommost **WorkflowExecution** entry by clicking its name.

---

