



## **Cisco Crosswork Workflow Manager 1.1 Get Started guide**

**First Published:** 2024-03-25

### **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> <div data-bbox="548 478 1404 982" data-label="Diagram"> <pre> graph TD     Start((Start)) --&gt; MyWorkflow[My workflow]     EventA[Event A] -.-&gt; MyWorkflow     EventB[Event B] -.-&gt; MyWorkflow     EventC[Event C] -.-&gt; MyWorkflow     MyWorkflow --&gt; ActionA[Action A]     MyWorkflow --&gt; ActionB[Action B]     MyWorkflow --&gt; ActionC[Action C]     MyWorkflow --&gt; End((End)) </pre> </div> <p data-bbox="457 1066 1101 1098">can also be initiated by an event that comes in to the system:</p> <div data-bbox="522 1150 1323 1591" data-label="Diagram"> <pre> graph LR     EventB[Event B] -.-&gt; Signal1[Signal]     Signal1 -.-&gt; MyWorkflow[My workflow]     Signal1 -.-&gt; MyOtherWorkflow[My other workflow]     Signal1 -.-&gt; RunJob[Run Job]     RunJob -.-&gt; YetAnotherWorkflow[Yet another workflow] </pre> </div>
Execution engine	<p data-bbox="457 1669 1469 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

# Run example workflow using Cisco NSO adapter

This section contains the following topics:

- [Run example workflow using Cisco NSO adapter, on page 7](#)

## 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

---

- 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

---

- 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

---

- 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:
- Resource name: `NSOLocal`
  - Resource type: `cisco.nso.resource.v1.0.0`
  - Secret ID: `NSOSecret`
  - 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:

---

- 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:

  - Log in to the CLI as admin:

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

- 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:
- Workflow definition name** - provide the name for the example workflow definition: `CreateL3VPN`.
  - Version** - provide workflow definition version: `1.0`.
- Step 4** Click **Create workflow**.

Figure 1: Add workflow

## Create new workflow

Workflow definition name\*

Version\*

### Run job

- 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


---

- 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"/> | All se  | <input type="text"/> |

```

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

```

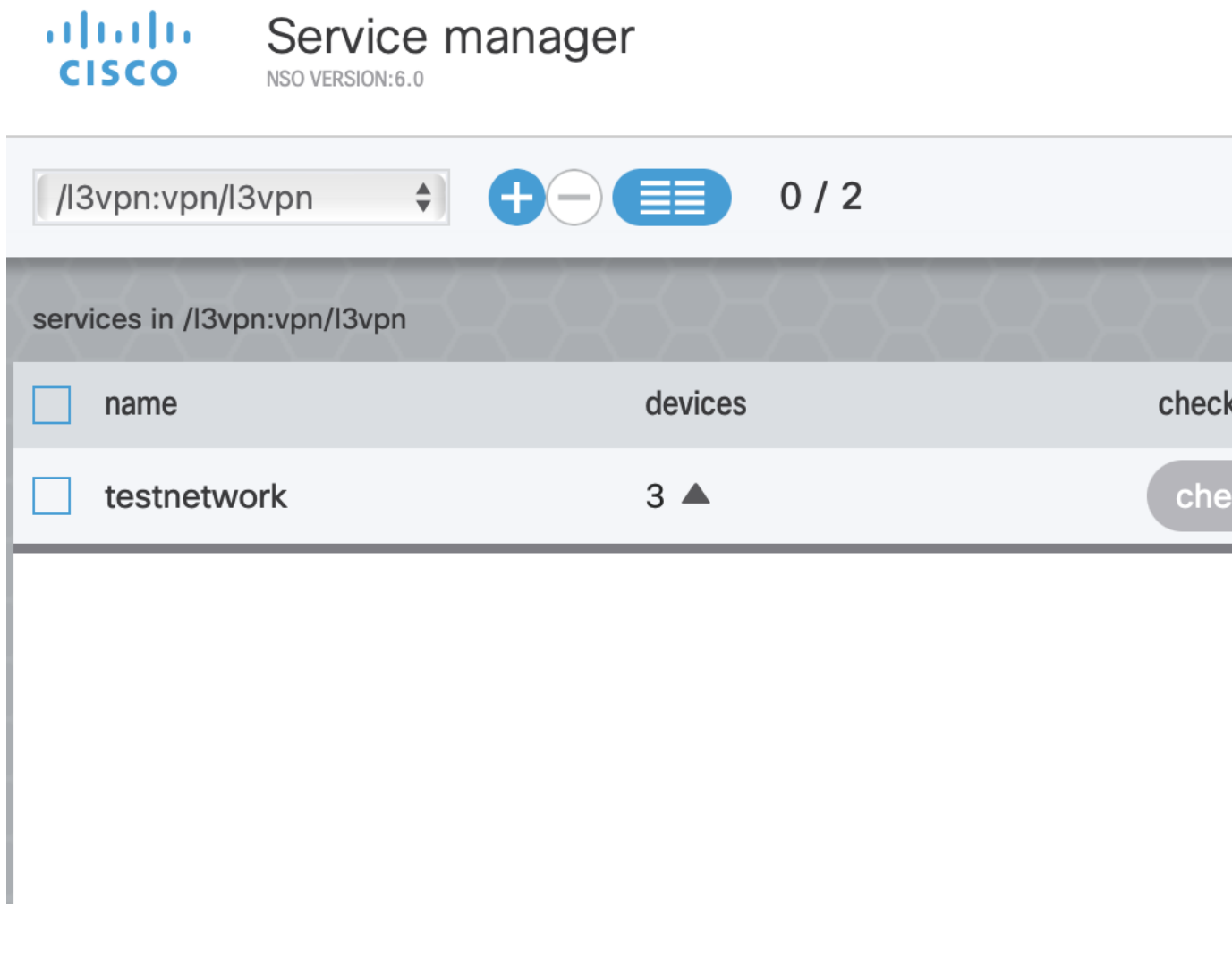
## In NSO

- 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   |

