Orchestration and Automation

- Orchestration and Automation Overview, page 1
- Cisco UCS Director Orchestrator, page 2
- Tasks, page 2
- Workflows, page 2
- Service Requests, page 3
- Approvals, page 3
- Rollback, page 3

Orchestration and Automation Overview

Cisco UCS Director enables you to automate a wide array of tasks and use cases across a wide variety of supported Cisco and non-Cisco hardware and software data center components, including physical infrastructure automation at the compute, network, and storage layers. A few examples of the use cases that you can automate include, but are not limited to:

- VM provisioning and lifecycle management
- Network resource configuration and lifecycle management
- Storage resource configuration and lifecycle management
- Tenant onboarding and infrastructure configuration
- Application infrastructure provisioning
- Self-service catalogs and VM provisioning
- Bare metal server provisioning, including installation of an operating system

For each of the processes that you decide to automate with orchestration workflows, you can choose to implement the processes in any of the following ways:

- Use the out-of-the-box workflows provided with Cisco UCS Director
- Modify the out-of-the-box workflows with one or more of the tasks provided with Cisco UCS Director

- Create your own custom tasks and use them to customize the out-of-the-box workflows
- Create your own custom workflows with custom tasks and the out-of-the-box tasks

For more information about automation and orchestration in Cisco UCS Director, see the Cisco UCS Director Orchestration Guide.

Cisco UCS Director Orchestrator

The Cisco UCS Director Orchestrator (also called the Cisco UCS Director Orchestration Engine, or just Orchestration) enables IT administrators to automate cloud deployment and provisioning and to standardize IT services.

At the scale of cloud computing, manually performing actions such as creating VMs and provisioning networks is prohibitively time-consuming. In Cisco UCS Director, these actions, or *tasks*, are executable elements that can be run from the GUI. Cisco UCS Director Orchestrator further automates these complex tasks by organizing them into *workflows*.

Orchestrator works by executing a series of scripted actions called *tasks*. Each task performs one action. By connecting tasks so that the input of one task is the output of a previous task, you build up a *workflow* to automate administrative processes such as creating VMs, provisioning bare metal servers, setting up storage, compute, and network resources, and many others.

Tasks

A task is the atomic unit of work in Cisco UCS Director Orchestrator. A task is a single action or operation with inputs and outputs (except in rare cases where the task operation does not require inputs or outputs). A task cannot be decomposed into smaller operations (exception: a compound task is made up of atomic tasks. But for now, think of a task as an indivisible unit of orchestration work).

Cisco UCS Director has a task library containing hundreds of predefined tasks that cover most of the actions an administrator must perform using Orchestration. In cases where there is no suitable predefined task, you can create custom tasks; see the Cisco UCS Director Custom Task Getting Started Guide.

Some examples of predefined tasks are:

- SSH Command task - Execute a command in a secure shell (SSH) session.
- Collect Inventory Task - Gathers information about available devices.
- Create New User - Add a user to the system.
- New VM Provision - Create a new VM (hypervisor-specific).

Workflows

A workflow is a series of tasks arranged to automate a complex operation. The simplest possible workflow contains a single task, but workflows can contain any number of tasks.

Workflows are the heart of the Cisco UCS Director Orchestrator; they enable you to automate processes of any level of complexity on your physical and virtual infrastructure.

You build workflows using the **Workflow Designer**, a drag-and-drop interface. In the **Workflow Designer**, you arrange tasks in sequence and define inputs and outputs to those tasks. Outputs from earlier tasks are available to use as inputs to any subsequent task.

Looping and conditional branching can be implemented using special flow-of-control tasks.

Service Requests

Service Requests are closely related to workflows. You create service requests by running workflows; a service request is generated every time you execute a workflow in Cisco UCS Director. A service request is a process under the control of Cisco UCS Director.

You can schedule a workflow for later execution, and Cisco UCS Director stores details of completed service requests. Thus a service request can have one of several states depending on its execution status: it can be scheduled, running, blocked (for example, awaiting an approval), completed, or failed (a service request can fail when one of its component tasks fails to execute properly).

Approvals

An approval is a "gate" task that requires the intervention of a Cisco UCS Director user to allow a workflow to run to completion. This user is typically an administrator who has go-or-no-go authority over the workflow process.

You can create *custom approvals* that allow users to enter input values when they approve a workflow. For example, you can create a custom approval to enable an IT administrator to approve provisioning of a VM, then specify the memory size of the VM before the workflow creates the VM.

Rollback

Workflows can be "rolled back" to a state identical or similar to the state before the workflow was executed. You can do this, for example, to remove virtual components that were created in error.

The term "rollback" often implies that a process is transactional, especially in systems using relational technology. However, it is important know that workflows are *not* transactional in a relational database sense. Instead, rollbacks work like this:

- Each task consists of two scripts. One script performs the work that the task was designed for. The other is a rollback script designed to "undo" the task. For example, if a task creates a VM, the rollback script deletes the VM.
- When you run a workflow, the scripts of the tasks in that workflow are executed in the order indicated by the workflow.
- When you roll back a workflow, the task rollback scripts are run in the reverse of the order they are run during normal workflow execution.
- A request to roll back a workflow creates a new service request, unconnected to the original service request.
- State information can be saved and used to roll back a task. For example, if you run a task to resize a VM, the pretask size can be saved to enable a rollback. This state persistence is a feature of the API that is used to write tasks; see the *Cisco UCS Director Custom Task* documentation for details.

- Tasks are atomic; workflows are not. You can roll back a workflow starting with any task in the workflow, but you cannot partially roll back a task.
- It is possible for a workflow to partially succeed; that is, for some but not all of its tasks to execute. Similarly, it is possible for a rollback to completely or partially fail; that is, that some or all of the rollback scripts could fail to execute.
- It is possible for a task to be written with a defective rollback script, or without a rollback script altogether. (Omitting the rollback script is not recommended.)